Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-16886-6

CNN and transfer learning methods with augmentation for citrus leaf diseases detection using PaaS cloud on mobile

Madhusudan G. Lanjewar[1] · **Jivan S. Parab**[1]

Received: 25 October 2022 / Revised: 8 June 2023 / Accepted: 4 September 2023

Abstract
Leaf and fruit infections are the primary cause of the maximum harm to the crop, which decreases the quality and amount of the goods. To improve the productivity of plants, the timely identification of the infection is vital, which is a highly challenging task. Deep learning (DL) with image processing allows farmers to distinguish between healthy and infected crops. This work intends to identify healthy and diseased citrus leaf images using a convolutional neural network (CNN) on the Platform as a Service (PaaS) cloud. The dataset of five types of healthy and unhealthy citrus images was used, namely, black spot, melanose, canker, greening, and healthy. Furthermore, the four-transfer learning (TL) pre-trained deep CNN (DCNN) models, namely, ResNet152V2, InceptionResNetV2, DenseNet121, and DenseNet201, were used to classify the leaf type. The performance of the CNN and four DCNNs were assessed using the confusion matrix (accuracy, precision, recall, and F1-score) and receiver operating characteristic-area under the curve (ROC-AUC) curve. An augmentation technique was utilised to enhance the dataset images, which helped to improve the model's performance and achieved an accuracy of 98% precision and recall and an F1 score of 99% and an ROC-AUC score of 0.99. Moreover, the suggested CNN has only 15 layers, 427317 parameters, and 1.68MB size, while DCNN models have more layers, parameters, and large size. The small-size CNN was deployed to the Platform as a Service (PaaS) cloud. The deployed model link is available on a smartphone to upload a citrus leaf image to the cloud, and the result is instantly available on a mobile screen.

Keywords CNN · Citrus leaf diseases · DenseNet201 · InceptionResNetV2 · Platform as a service

✉ Madhusudan G. Lanjewar
madhusudan@unigoa.ac.in

Jivan S. Parab
jsparab@unigoa.ac.in

[1] School of Physical and Applied Sciences, Goa University, Taleigao Plateau, Goa, India

Published online: 19 September 2023

1 Introduction

In India, the citrus industry is the country's third-largest after the mango and banana fruit industry. India ranks ninth among the world's top orange producers, contributing 3% to the total orange production worldwide and just 1.72% exported. Mandarin oranges from Nagpur, Maharashtra, are some of the best oranges in the world. Plant diseases are the main reasons for degrading the crop's quantity and quality, leading to economic losses [1–4]. To maximize crop yield, precise and timely detection of plant infection is essential [5]. The critical challenge is diagnosing diseases early to undertake efficient and targeted plant preservation interventions in crop production [6]. The standards of fruits, herbs, leaves, stems, and their products can be affected by illnesses, such as viruses, fungi, phytoplasmas, viroids, bacteria, and other pathogens. As the population rises, food production should increase for a steady supply [7]. Diseases and disorders affect plants and their products [8]. In the disease category, fungi, bacteria, or algae are the main reason for infection or illness. In the disorder category, rainfall, temperature, nutrient deficiency, moisture, etc., are the main reasons for the disorder [9, 10]. Environmental conditions are favourable for agriculture in developing countries like India, but incomes are low due to minimal investment in crop management infrastructure [10]. The pesticide is used for disease control after diagnosis. If the diagnosis is incorrect, the cost and the commodity will be affected by using the wrong pesticide [9]. The disease control of plants using pesticides is one of agriculture's most important research fields [11].

Fruits and vegetables benefit health and help reduce disease risks [12]. Citrus fruit is a natural source of carbohydrates, potassium, vitamin C, and glucose. For heart and sugar patients, the use of citrus fruit is helpful. High-water content of more than 85% helps to avoid dehydration by supplying fewer calories in energy to minimize weight [13]. Citrus is the world's second most significant fruit in cultivation and production. A group of vitamins, fibre, and minerals such as *carotenoids*, *limonoids*, and *flavonoids* that show healthy organic activity, which is present in citrus fruit [2]. Citrus fruit has antioxidants, which benefit human health [14]. The critical task of improving plant quality for economic growth is identifying and recognizing plant lesions [15]. More commercial and spontaneous hybrids of citrus varieties are available such as grapefruits, lemons, limes, oranges, and citric fruit [2]. Citrus plants can be infected by lesions, such as black spots and cankers [16]. The signs of these diseases are also first observed in leaves as spots. The leaves constitute the fundamental unit of plants and differ in form, scale, colour, structure, and texture [17, 18]. Numerous researchers have appealed to the classification of citrus leaves in South Asia [13, 18].

The deep learning (DL) method of artificial intelligence (AI) plays an essential role in the identification of infections by utilizing leaf images [19]. Various techniques are already available to detect diseases of citrus leaves, such as edges, borders, clustering, sedimentation, active contour, thresholding, etc. However, most of them require more memory, or the model size is also significant. CNN has been invented to fit almost all areas using an automated image extraction method [20], but many applications' performance could be better. The results of the CNN were improved by utilizing the rectified linear unit (ReLU) and the dropout concepts. The main advantage of CNN and pre-trained transfer learning (TL) DCNN models is automatically extracting features from the input image. Therefore, no feature extraction techniques are required. It improves the network's reliability and the number of parameters utilized in future CNN computations. This is one of the reasons why CNN is considered superior in the classification applications.

The use of CNN models in agricultural applications increased rapidly due to their various advantages. CNN gives cutting-edge output for many image recognition applications, such as object identification and recognition, image subtitling, and tracking [21]. Different CNN architectures, such as Inception and residual networks, were proposed for the classification and object recognition applications. Smart agriculture is suitable for developing technological devices with highly accurate algorithmic adaption [23–25]. Various environmental factors affect agriculture productivity, including weather, soil reserves, and quality. Many startups in agriculture are involved for better outcome, including the Agri Information Management System, Livestock Monitoring, Climate Recipes, the Intelligent Spot Spraying System (ISSS), and Image-Based Anomaly Detection. CNN can detect crop disease by scanning the leaf images [26–29].

The primary motivation for this paper is to build CNN and four DCNN models in Python. Test with different kernel sizes, convolution layers, max-pooling/average polling, dense layers, and dropout layers with their hyperparameters fine-tuning to extract and recognise characteristics that distinguish the unique type of citrus disease that affects leaves. CNN and DCNN models were executed using the Python programming language. This paper aims to organise data to aid in the economy of thought, save this data from reaffirming the categorised things of their characteristics, and discover new relationships and concepts in those identified characteristics. This effort also assists in predicting their behaviour, compiling a list of their optimal utilisation, assessing their output, and researching results that may be applied to other industries. Moreover, the key motivation for this effort is that the framework should function in the real world; authors should install it on real-time systems, cloud systems, or embedded systems. If the system for forecasting can be accessed on mobile, then other hardware is not required; all that is required for prediction is a mobile device with an internet connection. The authors attempted to create an efficient model that could be accessed by mobile, PC, or laptop.

The manuscript is organised in the following sections. Section 2 discusses the literature reviews, the objective of the work, and the research gap with the main contributions. Section 3 discusses the materials and methods, including the proposed framework, dataset information, CNN, and DCNN model details with pseudocode. Section 4 presents the obtained experimental results. Section 5 discusses the comparison of obtained results with literature-reported work. Section 6 concludes the proposed work.

2 Related work

This section will discuss the literature-reported plant disease detection methods, including citrus diseases. Around the world, various citrus varieties are grown. However, a hand-held, efficient disease detection system on the mobile or embedded system can be developed to detect diseases at early stage. Different methodology for segmentation and extraction, including threshold, edge, regional, Gabor, wavelet transform, and principal component analysis (PCA), was discussed by Prajapati et al. [30]. Most plant diseases were recognized using leaf symptoms [31, 32].

Iqbal et al. [2] present a review article on automatic image processing techniques to identify citrus plant illness. Xiang et al. [5] implemented a lightweight CNN-based system with channel shuffle operation and multiple-size module (L-CSMS) to detect plant diseases. They achieved 97.9% and 90.6% accuracy on the PlantVillage and plant disease severity datasets, respectively. Hamuda et al. [32] presented a survey on image processing for plant segmentation and removal of the background techniques. They used colour segmentation with green pixel masking to eliminate the background and employed the infected image with the Otsu threshold technique. Revathi et al. [33] introduced the Homogeneous Pixel Counting technique for Cotton Diseases Detection (HPCCDD) to identify and categorize cotton diseases from mobile captured images. The proposed method employed various strategies, including RGB feature ranging techniques, colour image segmentation, and the utilization of Sobel and Canny filters to identify disease spots and edges for disease identification. Their approach achieved an impressive accuracy of 98.1%. Patil et al. [34] employed an SVM classifier to differentiate the illness after identifying features such as contrast, texture, cluster shade, homogeneity, and cluster prominence. The authors found fungal diseases in sugar cane by simple and triangular threshold values for the segmentations of leaves and lesions and achieved 98.60% accuracy. Szczypiski et al. [35] proposed a method based on isolated kernels' image-derived form, texture parameters, and colour to classify barley varieties. Attributes dimension reduction, linear classifiers, and artificial neural networks (ANN) were used, achieving accuracy between 67% and 86%. Pydipati et al. [36] introduced a colour co-occurrence method (CCM) for identifying normal and diseased citrus leaves with greasy spots, melanose, and scabs. The method combined texture-based hue, saturation, and intensity (HSI) colour attributes with statistical categorization technique and achieved accuracies of over 95%.

Rauf et al. [37] generated a citrus fruits and leaves dataset of infected and uninfected plants such as canker, black spot, scab, melanose, and greening to detect and classify illness utilizing the ML approach. Qadri et al. [38] proposed a machine vision (MV) framework for classifying the eight types of lemon leaves. The collected digital images were converted into a multi-feature dataset comprised of histogram, binary, spectral, texture, scalability, rotational, and translational invariant characteristics. The optimized multi-features dataset was fed into various MV classifiers: Random Forest (RF), Multilayer Perceptron (MLP), Nave Bayes (NB), and J48. They also applied a 10-fold cross-validation technique. MLP obtained an overall accuracy of 98.14%. Parraga-Alava et al. [39] suggested a VGG16-based method to detect aphid lemon leaf images and reach an average of 81% and 97% rates on a real lemon leaf dataset. Yang et al. [40] created a leaf segmentation and identification framework from 2500 complex background images using Mask Region-based CNN (Mask R-CNN) and VGG16 methodologies. They trained a Mask R-CNN model for leaf segmentation and fed over 1500 training photos from 15 species to a VGG16 for leaf categorization. The average mis-classification error of 80 test images utilizing Mask R-CNN was 1.15%, and the average accuracy for categorizing leaves in 150 test photos using VGG16 was 91.5%. Guo et al. [41] used the VGG16 and region proposal network (RPN) for plant illness detection (bacterial plaque, black rot, and rust). The RPN was employed to identify and localize the leaves and image segmentation through the Chan–Vese algorithm. They achieved an accuracy of 83.57%.

Tsolakidis et al. [42] developed a framework for plant leaf detection using the Histogram of Oriented Gradients and Zernike Moments approaches. The venation identification approach was used by Kolivand et al. [43] to categorize leaf forms and identify species of plants. The suggested method includes canny edge detection, curve extraction, leaf boundary removal, hue normalization picture, and image fusion. Furthermore, the lines retrieved from pre-processing were further split into smaller pieces to localize the edge direction effectively. Flavia and Acer databases assess and evaluate 32 leaf pictures of Malaysian plants and achieved an average accuracy of 98.6% and 89.83%, respectively. Puri et al. [44] proposed a framework using mask analysis and an SVM to recognize the different varieties of medical plant leaves and achieved the highest accuracy of 90.27%. Vilasini et al. [45]

employed cell phones to recognize Indian leaf varieties utilizing KNN, SVM, and CNN algorithms. They use a cluster of edge detectors, a traditional Sobel edge detector, and a Laplacian edge detector to segment leaf edges and veins. The results of the Sobel and Laplacian operators were averaged with Prewitt edge detection, and the leaf skeleton was created for further categorization. They obtained 100%, 92%, and 80% accuracy for the 2, 5, and 10 classes, respectively.

Deepalakshmi et al. [46] suggested a CNN-based approach for distinguishing between diseased and healthy leaves of different plants. They employ CNN as a feature extractor and the extracted features aid in identifying the most suitable class. The suggested system identifies the picture class with greater than 94.5% accuracy in an average time frame of 3.8 seconds. To categorise four forms of citrus leaf illnesses (canker, greening, sooty mould, and leaf-miner), Dang-Ngoc et al. [47] developed a hierarchical SVM method. Various image pre-processing methods were used, and leaf characteristics were retrieved in multiple colour spaces, with the most significant ones selected based on feature distribution assessment. These specified characteristics were supplied to SVM to identify and categorise illnesses. They obtained a recognition rate of 92.5% and a high accuracy rate of 91.76% for diseased leaves.

2.1 Research gap and objectives

According to the literature mentioned earlier, most scientists created ML models such as SVM, RF, and others, as well as DL models such as CNN, VGG16, and others, to diagnose illnesses and get satisfactory outcomes. The references [5, 33–36] used CNN, HPCCDD, SVM, CCM and other methods. They achieved accuracies between 90.6% to 98.6% but they have not reported the model size. The Parraga-Alava et al. [39], Yang et al. [40], and Guo et al. [41] used VGG16 and achieved lower accuracies compared to our work. All the literature reported work also achieved lower performance than our work. Most of the researchers repeated the accuracy metric, but in this work, the imbalanced dataset was used, and the accuracy metric is not a good performance measure which can evaluate the models, but F1-score and ROC-AUC curve could be better choices with accuracy, precision, and recall metrics.

The authors discovered an opportunity to increase the categorization system's performance. By exploring the issue with conventional models, the study aims to produce a new awareness of it. In this study, the authors used the technology-assisted research technique. This research aims to improve existing systems by altering CNN frameworks to successfully diagnose citrus plant diseases while lowering model size and parameters. The authors used the small citrus leaf dataset with a basic random sampling approach. Moreover, most researchers implemented the models, but the model size still needs to be reported, which is essential when the models are deployed to real-time systems, cloud or embedded systems. Developing an efficient design with a small model can fill this research gap.

The main objective of this study is to develop a small-size, reliable and efficient method for recognizing citrus leaf diseases with healthy leaves that will benefit farmers and other users, especially those doing smart farming. The authors feel that more research may be required to establish techniques for detecting citrus leaf diseases. To fill these research gaps, an efficient method was proposed by integrating camera captures colour imaging with a customized CNN and four DCNN (ResNet152V2, InceptionResNetV2, DenseNet121, and DenseNet201) models for identifying citrus leaf diseases. Image processing with ML or DL is used in a variety of new ways to build dependable approaches with speedy results. Furthermore, typical image processing algorithms are susceptible to background noise, which reduces the model's performance. The suggested system must meet these requirements while considering computing expenses, noise, and accuracy. Moreover, if the complete system is available on a smartphone, it will be easy for users to check the citrus leaf disease on the phone instantly. These requirements were considered; the authors developed a small-size CNN and deployed it to the PaaS cloud so the user could access the system-deployed link on his smartphone. The user will access the system link with just one click with an internet connection. The user captures the photo using a smartphone camera and uploads it to the system-deployed link, and the result will display instantly on the smartphone screen.

Previously, experts or producers utilized manual approaches to disease identification, but they are time-consuming and costly. Agro-technology improvement can help farmers recognize the disease in the early stage. It will aid in increasing agricultural output and farmer income. If the farm is small, the classification challenge is straightforward and can be performed with the naked eye without much experience. Because smart farming demands an early illness-predicting model, this research problem must be considered. Farmers may profit from the proposed plan if the sickness prediction model is accessible through a mobile phone or other embedded device.

CNNs have demonstrated beneficial results; however, notwithstanding their impressive results, CNNs face unique problems, such as a trade-off between usefulness and CPU utilization. CNN, in contrast, demands a significant size of RAM and incurs computing resource costs in classification and verification, rendering its implementation on IoT networks, smartphones, and single-board processors impractical in most scenarios. As a result, analyzing images on a local computer is often tricky, even without graphics processing units (GPUs) or cloud application services. The main contribution of this work includes:

- Develop an effective system with customized CNN using various layer combinations and fine-tuning their hyperparameters, which helps to extract the best features for citrus leaf disease classification. The methodology aimed to solve traditional issues, such as high data noise in photos and the inefficiency of DL/ML approaches. The recommended solution incorporates pre-processing procedures such as imaging re-sizing and scaling to address the limitations of recognizing citrus leaf illnesses from camera-obtained pictures, which influence accuracy;
- The proposed technique is primarily based on a CNN and DCNN, which extract beneficial characteristics and help in performance improvement. To help in successful feature extraction, the Adam and RMSprop optimizers were fine-tuned with varied hyperparameters, including learning rate;
- The authors experimented with different kernel sizes of convolutional layers, which helped to reduce processing duration and computational burden while enhancing model efficacy;
- The augmentation method was applied to enhance the dataset, which helped improve the models' performance;
- A confusion matrix and AUC-ROC were utilized for checking model performance; and
- The CNN achieved an F1 score of 99%, and the model size is also small. It should also be highlighted that none of the investigators has revealed the model's size, which is critical for deploying it on the embedded platform and cloud. As a result, the proposed

CNN model is smaller than others. This makes deploying the model on the PaaS cloud less costly and will display results quickly.

3 Materials and methods

The proposed CNN-based citrus leaf disease prediction system is shown in Fig. 1. The citrus leaf images dataset of healthy and unhealthy images was used [37]. A dataset consists of 609 RGB images of citrus leaves such as black spots, canker, greening, melanose, and healthy. The dataset images were resized to 224×224×3 and 299×299×3, and the feature scaling method was applied. Feature scaling is a technique used to standardize or normalize the characteristics in dataset photographs, which multiplied every element by 255. The label encoding approach was utilized to turn labels into numbers that ML/DL algorithms may process. Every group has its numerical value, allowing systems to work efficiently with the coded label. A label encoder was used to label each image, allocating a label (0, 1, 2, 3, and 4). The training dataset's augmentation process was utilized to enhance the pictures artificially. The CNN and DCNN were developed using Python. The healthy or diseased citrus leaf was predicted using CNN and DCNN. The performance measures, such as accuracy, precision, recall, F1 score, and the ROC-AUC curve, were employed to evaluate the model performance. After the model performance assessment, the CNN was deployed to the cloud.

3.1 Citrus leaves images dataset

The citrus leaves image dataset includes five categories of healthy and unhealthy citrus photos (black spot, melanose, canker, greening, and healthy), with 609 pictures [37]. The

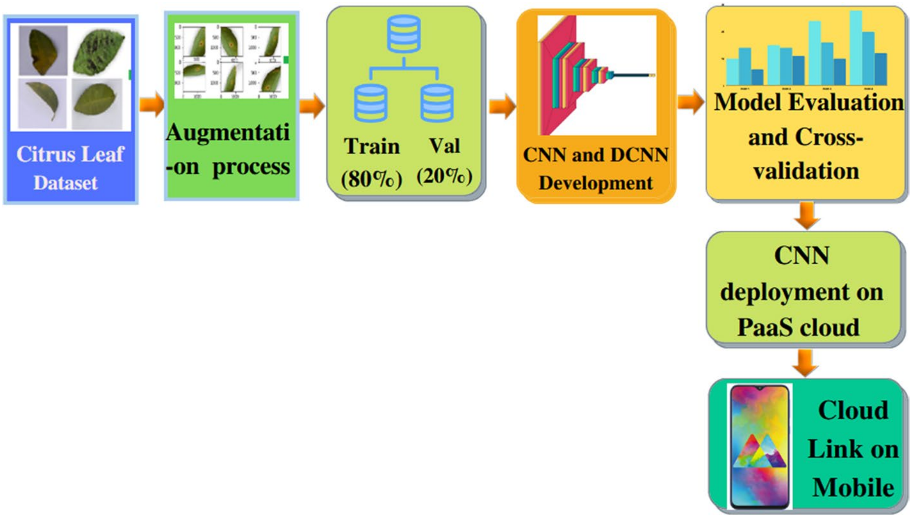


Fig. 1 Entire citrus leaf disease detection system framework

Table 1 Dataset details

Leaf Types	Training Dataset	Validation Dataset	Total Images
Black Spot	141	30	171
Melanose	10	03	13
Canker	133	30	163
Greening	174	30	204
healthy	48	10	58
Total	506	103	609

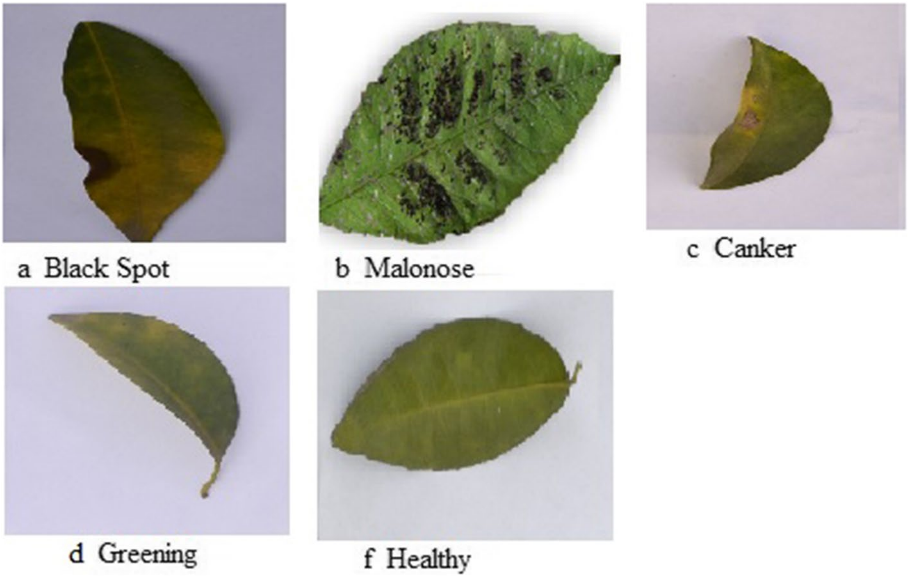


Fig. 2 Citrus leaf dataset images **a**) Black spot, **b**) Melanose, **c**) Canker, **d**) Greening, and **e**) Healthy

details of the citrus leave photos dataset are shown in Table 1. The citrus leaf image collection was separated into a training dataset (506 photos), and a validation dataset (103 images) at around 80:20. Figure 2 depicts various dataset citrus leaves.

3.2 Data augmentation

The data augmentation process is the artificial generation of new images from the original dataset while maintaining the label of the newly generated images [48]. It is a convenient and commonly used tool for producing more data for training to enhance the model performance. The data augmentation method is often used for image analysis in ML [49]. The stochastic and pipeline-based approach is used for image augmentation. This approach enables the user to increase the training dataset using chain operations such as shears, vertical and horizontal rotations, zoom, crops, etc. The augmentation parameters used in this work are tabulated in Table 2. The dataset is small, but due to the augmentation, the dataset images were increased from 503 (training dataset images) to approximately 7084 images. Figure 3 shows the augmented image of the black spot diseased leaf.

Table 2 List of augmented parameters

Augmentation parameters	Value	Training dataset	Images after augmentation process	Total images
width_shift_range	0.20	506	2x506	1012
rotation_range	45	506	2x506	1012
shear_range	0.15	506	2x506	1012
height_shift_range	0.20	506	2x506	1012
vertical_flip	True	506	2x506	1012
horizontal_flip	True	506	2x506	1012
zooming range	0.40	506	2x506	1012
			Total	7084

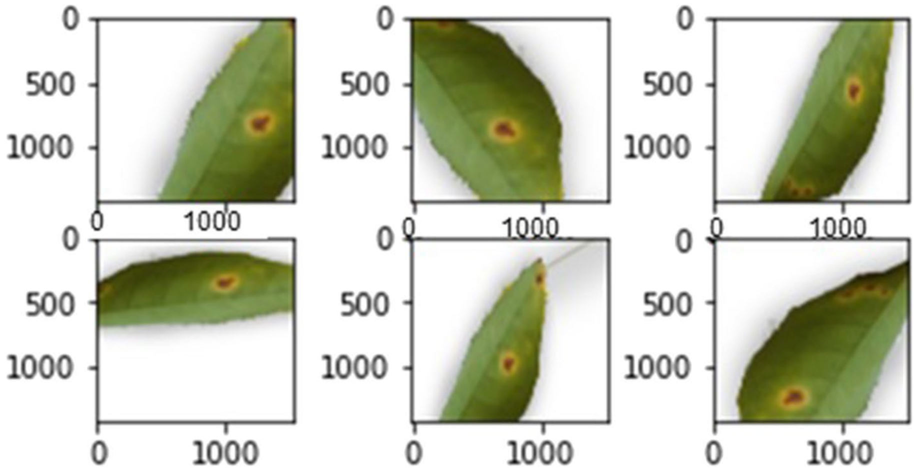


Fig. 3 Augmented image of black spot disease

3.3 Convolutional neural network (CNN)

The standard architecture of any CNN model includes convolution layers (CL), pooling layers (PL), dropout layers, flatten layers (FL), dense layers, etc [22, 50]. CL consists of filters or kernels. Input features and weights pass through the CL, which gives corresponding output features. The suggested CNN framework uses the ReLU activation function to predict citrus leaf disease. The activation function is essential in CNN to learn and carry out more complex tasks. Various low-level characteristics, such as edges, corners, and lines, are extracted in the first CL [51]. The CLs' assembly guides the CNN network to acquire more global features. With a rise in the number of CLs, network parameters exponentially grow. Pooling operations are then carried out throughout the area to decrease the number of network parameters.

In this work, average polling reduces the representation's spatial size. A 3x3 filter with a stride of 2 was a standard size for the pooling layer. Flatten or fully connected (FC) layer was among the final layers of the CNN architecture. Figure 4 shows the proposed CNN architectures, and Table 3 depicts the output shape of the layers and the number of

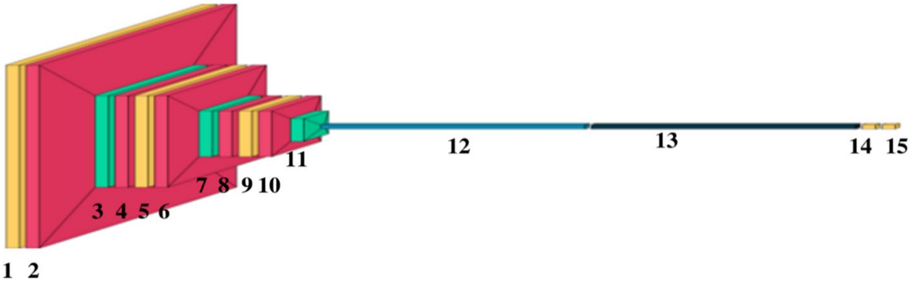


Fig. 4 CNN architecture with layer numbers

parameters. The proposed CNN had three CL layers, which have kernel sizes of 94, 32, and 16 first, second, and third CL, respectively. The filter size of 3x3 and ReLU was utilized for all CLs, with padding set to 'same'. Following CL, batch normalization (BN), average pooling layer (APL) with a pool size of 2 and stride of 2, then again BN was applied. The first dense layer contains 32, followed by the ReLU. Finally, an FC layer was made up of 5 Softmax neurons, as depicted in Table 3.

3.4 DCNN models

The DCNNs models, like Resnet152V2, InceptionResNetV2, DenseNet121, and DenseNet201, were chosen because of their remarkable achievements in various applications. The Resnet152V2 model has 152 layers, InceptionResNetV2 has 572, DenseNet121 has 121, and DenseNet201 has 201 layers. Each of the four models has a unique structure as well as differences in layers and sizes. These models are described briefly below:

Table 3 The CNN model details

Layer numbers	Layers (type)	Output Shape	Param #
1	conv2d_32 (Conv2D)	(None, 224, 224, 64)	1792
2	batch_normalization_40 (BN)	(None, 224, 224, 64)	256
3	average_pooling2d_24 (APL)	(None, 112, 112, 64)	0
4	batch_normalization_41 (BN)	(None, 112, 112, 64)	256
5	conv2d_33 (Conv2D)	(None, 112, 112, 32)	18464
6	batch_normalization_42 (BN)	(None, 112, 112, 32)	128
7	average_pooling2d_25 (APL)	(None, 56, 56, 32)	0
8	batch_normalization_43 (BN)	(None, 56, 56, 32)	128
9	conv2d_34 (Conv2D)	(None, 56, 56, 16)	4624
10	batch_normalization_44 (BN)	(None, 56, 56, 16)	64
11	average_pooling2d_26 (APL)	(None, 28, 28, 16)	0
12	flatten_8 (Flatten)	(None, 12544)	0
13	dropout_8 (Dropout)	(None, 12544)	0
14	dense_20 (Dense)	(None, 32)	401440
15	dense_21 (Dense)	(None, 5)	165
	Total params: 427,317		

Resnet152V2: In 2015, Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun proposed the Residual Network (ResNet) DCNN model [52]. ResNet was a family of several deep neural networking frameworks with different depths [29]. ResNet introduces ReLU, which reduces deep neural network degradation [7]. Among members of the ResNet family, the Resnet152 DCNN model has the best accuracy [29]. This network can take the input image height and width in multiples of 32x3 because of the width of the channel, and 224x224x3 is the default image size.

Inception-ResNet-V2: Inception-ResNet-V2 combines an Inception framework and a residual connection [52]. Multi-size convolutional filters are combined with residual connections on the Inception-Resnet block [52]. Using residual links prevents the issue of deterioration caused by profound structures and reduces the training time [7].

DenseNet121and DenseNet201: Because redundant maps are not learned, Dense Convolutional Network (DenseNet) requires fewer parameters than regular CNN [9, 52]. This work employs DenseNet121 and DenseNet201 models. In DenseNet, each layer takes and sends new inputs from all previous levels. The loss function allows each layer to access the original input image and gradients. As a result, the computation efficiency improves significantly, giving DenseNet a better option for image classification.

3.5 Cloud computing

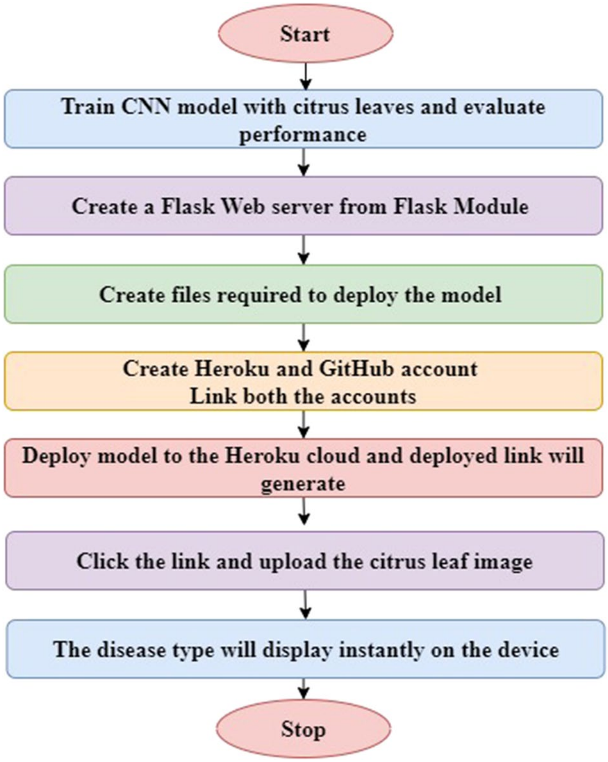
Cloud computing is a breakthrough in information technology (IT). The cloud computing system can also provide Infrastructure for the Internet of Things (IoT), mobile technology, big data, and artificial intelligence. Cloud computing speeds up industry dynamics, disrupts old models, and encourages digital change [50]. Cloud computing is classified according to its deployment and service types. Model deployment services are classified into four types: Function-as-a-Service (FaaS), Infrastructure as a Service (IaaS), PaaS, and Software as a Service (SaaS). PaaS has grown in popularity among cloud computing service models due to its ability to improve overall business performance agility [53]. PaaS provides cloud clients with middleware resources [53]. It also provides customers with development and testing environments. Depending on deployment, clouds are categorized as private, public, or hybrid. The authors utilized Heroku Cloud, a simple cloud service platform for the application.

The Hypertext Markup Language (HTML) was employed for page layout, while Flask was used for the front-end look. Flask is a Python-based web platform that offers valuable features and resources to build web applications. The model may be deployed to clouds in many methods [50, 53], including GitHub, Docker-based deployments, etc. The flowchart of the model deployment process is shown in Fig. 5.

4 Results

The citrus leaves dataset contained 609 images, split into training (506) and validation (103) datasets. The citrus plant leaf image was resized to 224x224, and the training dataset employed the augmentation process. These augmented images were used to train the models. All models used the same training and validation dataset. All experiments were done online on the Google Colab platform. Python was used to implement all four state-of art transfer

Fig. 5 Model deployment process on to the Heroku cloud



learning (TL) pre-trained DCNN and CNN models. The RMSprop and Adam optimizers were employed to attain global minima, and the batch size was 32. The Adam optimizer hyperparameters were initialized with lr of '0.0003', beta-1 of '0.9', beta-2 of '0.999', epsilon of 'None', decay of '1e-8', and amsgrad of 'False'. The loss was initialized with 'categorical_crossentropy'. Similarly, The RMSprop optimizer hyperparameters were initialized with a learning_rate of '0.0003', rho of '0.99', epsilon of 1e-08, and decay of '0.0'. The loss was initialized with 'categorical_crossentropy'.

4.1 Performance measures

The confusion matrix method was employed in this work to check the model's performance. It is used to track the classification model's helpfulness and types of errors. The confusion matrix metrics can be measured using the following equations [22, 54, 55].

$$\text{Accuracy}\ \left(\text{Acc}_{\text{c}}\right) = \frac{\text{TPc} + \text{TNc}}{\text{TPc} + \text{FPc} + \text{TNc} + \text{FNc}} \tag{1}$$

$$\text{Precision}\ \left(\text{Pre}_{\text{c}}\right) = \frac{\text{TPc}}{\text{TPc} + \text{FPc}} \tag{2}$$

$$\text{Sensitivity or Recall or TPR}\left(\mathrm{Re}\,c_c\right) := \frac{\mathrm{TPc}}{\mathrm{TPc} + \mathrm{FNc}} \tag{3}$$

$$\mathrm{F1} - \mathrm{score}\left(\mathrm{F1}_c\right) = \frac{2}{\left(\frac{1}{\mathrm{Recall}}\right) + \left(\frac{1}{\mathrm{Precision}}\right)} \tag{4}$$

Where TNc (True Negative): correctly recognizes negative cases; TPc (True Positive): correctly recognize positive cases; FPc (False Positive): actual negative cases categorised as positive; and FN (False Negative): actual positive instances categorised as negative.

4.2 Psedocodes For CNN

```
1. print("Citrus Leaf disease detection system...")
2. input Dataset Labels
3. Image resize to 224x224
4. Feature scaling= Image/255.0
5. Apply Augmentation method
6. Split Dataset
7. input batch_size=32 epochs=200 Max_acc
8. model=Sequential CNN model Adam RMSprop
9. For i= 1 to epochs Do
        Start Training process
        print(" Training accuracy Val accuracy with losses")
        Callback If Val_acc=Max_acc
            break
            Endif
      endSubroutine
       i=i+1
       Endfor
10.     Accuracy=(TPc+TNc)/(TPc+FPc+TNc+FNc)
11.     precision =(TPc)/(TPc+FPc)
12.     Recall=(TPc)/(TPc+FNc)
13.     F1score= (2/ ((1/Recall)+ (1/Precision))
14.     End
```

4.3 Feature map of CNN layers

The first CL extracts low-level characteristics such as borders, curves, and lines, as shown in Fig. 6. The following CLs acquire the global features. The CNN model learns these features through a back-propagation training technique. The four separate sections of the back-propagation algorithm are the forward pass, the loss function, the backward pass, and the weight update.

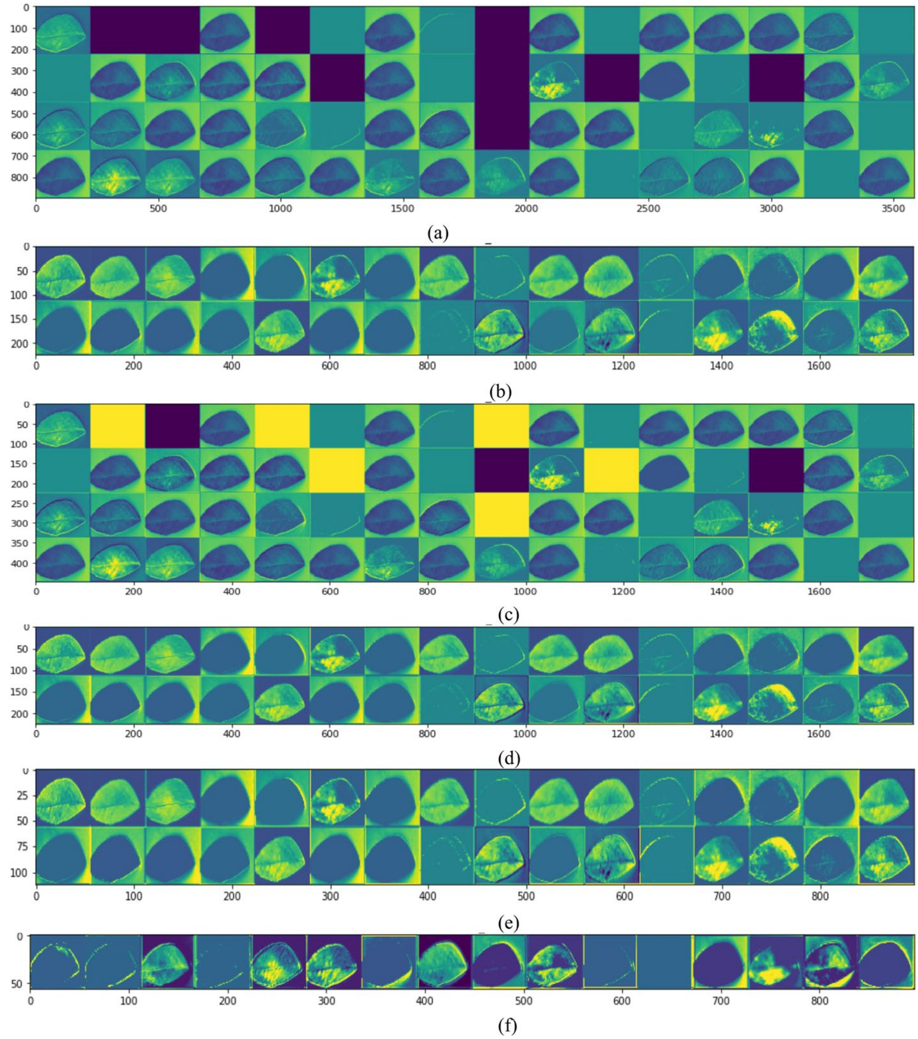


Fig. 6 Output of **a**) first CL, **b**) second CL, **c**) third CL, **d**) flatten layer, **e**) first dense layer, **f**) second dense layer

4.4 CNN performance

In CNN, the kernel size plays an important role, as it aids in acquiring the provided features and processes the data at the convolution layer. The kernel's size is less than the input size [56]. The authors tried kernel sizes from 2x2 to 5x5 and found that 3x3 was better than others. Furthermore, the authors vary the learning rate (LR) to fine-tune the models. The LR is a predefined hyper-parameter for neural network development that considers how quickly a machine understands its surroundings, and LR values are between 0.0 and 1.0 [56]. The authors varied the values of LR from 0.0001 to 0.0005 and found that 0.0003 was the best LR.

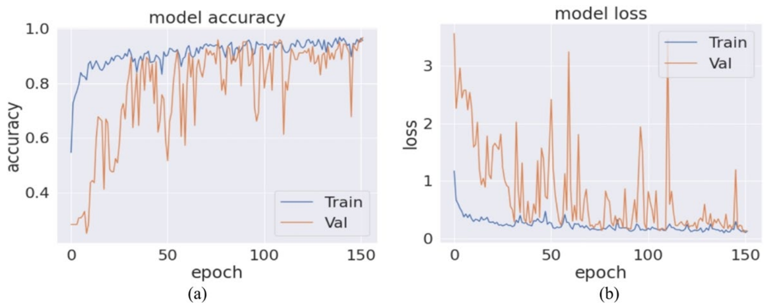


Fig. 7 CNN with Adam optimizer training and validation performance

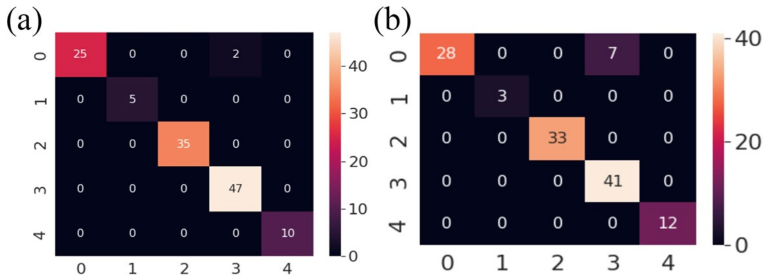
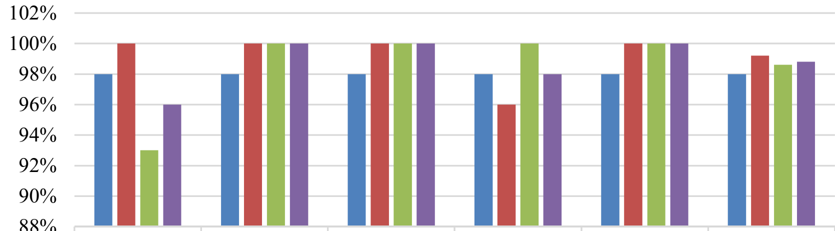


Fig. 8 Confusion matrix of **a**) CNN-Adam-Aug and **b**) CNN-RMS-Aug. (1 indicates Black spot, 2-Melanose, 3-canker, 4-greening, and 5-healthy)

Initially, the authors assigned 200 epochs with a learning rate of 0.0003 and then checked the accuracies, and we found that above 150 epochs, the accuracies were saturated. After that, the authors again run code with a callback function to stop that desired validation accuracy. The 152 epoch was required for the desired accuracy. Figure 7a and b depict the CNN-Adam-Aug training and validation process and achieved 97% training accuracy with a loss of 0.0994 and 97% validation accuracy with a val_loss of 0.1255.

This work used two optimizers with augmentation methods: CNN with Adam optimizer with augmentation (CNN-Adam-Aug) and CNN with RMSprop optimizer and augmentation (CNN-RMS-Aug). Figure 8 shows the confusion matrix of CNN-Adam-Aug (Figure 8a) and CNN-RMS-Aug (Figure 8b). Figure 8a shows that 101 images out of 103 were predicted correctly by CNN-Adam-Aug, while 96 images correctly predicted CNN-RMS-Aug. There are five classes in the dataset, and only two images were mispredicted by CNN-Adam-Aug. This indicates that the CNN-Adam-Aug perform better than the CNN-RMS-Aug model.

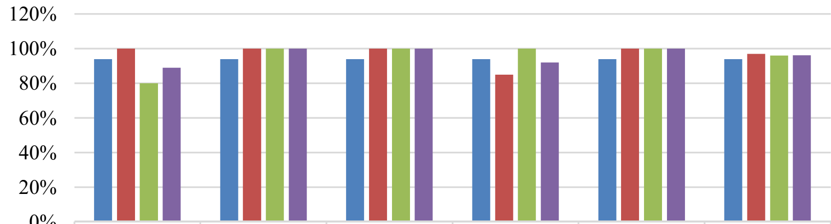
Figure 9a shows the Accuracy (Acc_c), Precision (Pre_c), Recall (Rec_c), and F1-score (F1_c) of individual class performance of the CNN-Adam-Aug, while Fig. 9b for CNN-RMS-Aug. The citrus leaf dataset is unbalanced, so the F1_c is the best metric for the overall performance of the models. Figure 9a shows that the Acc_c for all classes was 98%. The Pre_c, Rec_c, and F1_c of Melanose, cancer, and healthy were 100%. The CNN-Adam-Aug model predicts Black spot images with 100% Pre_c, 93% Rec_c, and 96% F1_c, while greening images predict 96%, 100%, and 98%, respectively. The average Acc_c, Pre_c, Rec_c, and F1_c performance of all classes was 98%,



	Black spot	Melanose	canker	greening	healthy	Average
Acc_c	98%	98%	98%	98%	98%	98%
Pre-c	100%	100%	100%	96%	100%	99%
Rec_c	93%	100%	100%	100%	100%	99%
F1-c	96%	100%	100%	98%	100%	99%

Acc_c Pre-c Rec_c F1-c

(a)



	Black spot	Melanose	canker	greening	healthy	Average
Acc_c	94%	94%	94%	94%	94%	94%
Pre-c	100%	100%	100%	85%	100%	97%
Rec_c	80%	100%	100%	100%	100%	96%
F1-c	89%	100%	100%	92%	100%	96%

Acc_c Pre-c Rec_c F1-c

(b)

Fig. 9 Accuracy (Acc_c) , Precision (Pre_c), Recall (Rec_c), and F1-score (F1_c) for **a**) CNN-Adam-Aug and **b**) CNN-RMS-Aug model

99%, 99%, and 99%, respectively. Figure 9b shows that the Acc_c for all classes was 94%. The Pre_c, Rec_c, and F1_c of Melanose, cancer, and healthy were 100% similar to the CNN-Adam-Aug. The CNN-RMS-Aug model predicts Black spot images with 100% Pre_c, 80% Rec_c, and 89% F1_c, while greening images predict 85%, 100%, and 92%, respectively. The average Acc_c, Pre_c, Rec_c, and F1_c performance of all classes was 94%, 97%, 96%, and 96%, respectively. This performance indicates that the CNN-Adam-Aug performs better than the CNN-RMS-Aug model.

4.4.1 AUROC (Area Under the Receiver Operating Characteristics)/ROC-AUC performance

Figure 10a and b shows the AUROC performance of the CNN-Adam-Aug and CNN-RMS-Aug models, respectively. The AUROC curve is utilized to assess or display the effectiveness of the multi-class classification task. This is among the most effective assessment methods for evaluating the efficacy of any classification algorithm. AUC indicates the level or measurement of separability, while ROC is a probability graph showing how well the model can differentiate among categories [54]. The greater the AUC, the more accurately the model predicts 0 classes as 0, 1 class as 1, etc. Similarly, the greater the AUC, the more accurately the model distinguishes between diseased and healthy citrus images. The ROC curve is displayed with True Positive Rate (y-axis) versus False Positive Rate (x-axis).

Figure 10a shows AUROC of 0.96, 1.0, 1.0, 0.99, and 1.0 for Black spot, Melanose, cancer, greening, and healthy class, respectively, which indicates the CNN-Adam-Aug performs better for predicting all categories of images. Only the Black spot class has 0.96; otherwise, all other classes have above 0.99 AUC. The average AUROC score was 0.99, which shows remarkable performance.

On the other hand, Figure 10b shows AUROC of 0.90, 1.0, 1.0, 0.96, and 1.0 for Black spot, Melanose, cancer, greening, and healthy class, respectively, which indicates the CNN-RMS-Aug performance is less compared to CNN-Adam-Aug. The Blackspot and greening types have 0.90 and 0.96 AUC; otherwise, all other categories have 1.0 AUC. The average AUROC score was 0.97, which shows acceptable performance.

4.5 Comparison of CNN with DCNN models

Figure 11 shows the performance of all models in terms of Acc_c, Pre_c, Rec_c, and F1_c. All DCNN models were trained with similar sets with the Adam optimizer, and all other parameters were the same for CNN and DCNN models, including augmentation parameters. The ResNet152V2 with Adam and augmentation (ResNet152V2-Aug) achieved Acc_c, Pre_c, Rec_c, and F1_c of 88%, 92%, 83%, and 87%, respectively, while InceptionresNetV2-Aug achieved 85%, 92%, 80%, and 84%, respectively. On the other hand, DenseNet121-Aug and DenseNet201-Aug achieved Acc_c, Pre_c, Rec_c, and F1_c of 88% and 92%, 93% and 96%, 89% and 93%, and 90% and 94%, respectively. The Acc_c, Pre_c, Rec_c, and F1_c performance of CNN-Adam-Aug was 98%,99%, 99%, and 99% respectively. The performance of CNN-Adam-Aug is better compared to other models. CNN with

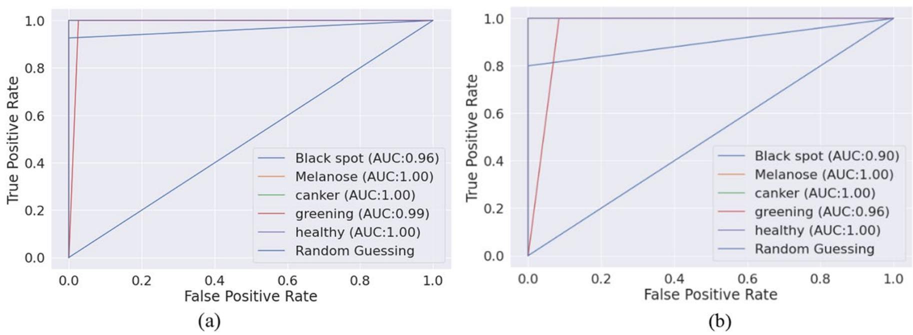


Fig. 10 AUROC performance of the **a**) CNN-Adam-Aug and **b**) CNN-RMS-Aug

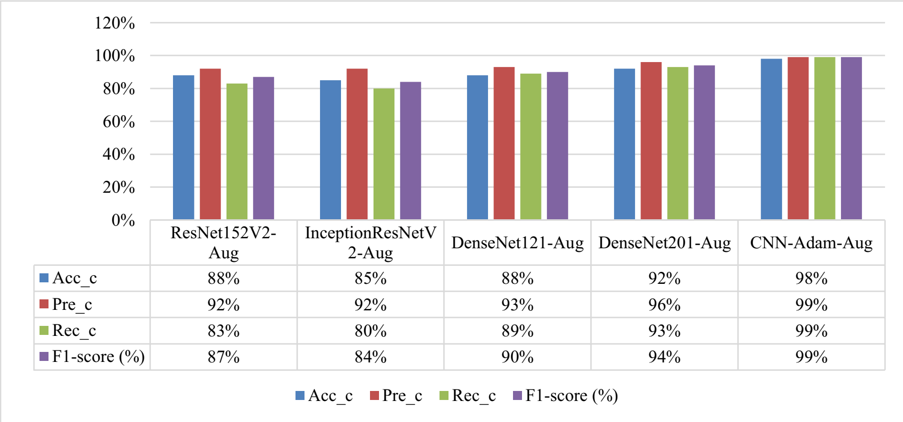


Fig. 11 comparisons of CNN-Adm-Aug with DCNN models

Adam optimizer outperformed all other models and was recommended for citrus leaf disease prediction.

4.6 Cloud computing

The empirical analysis of the CNN-Adam-Aug model was excellent. The CNN-Adam-Aug result was studied and compared with DCNN models in layers, memory size, parameters, and F1-score. This type of comparison is essential in the real world because, in the end, we have to deploy these models to the single board computer, embedded devices or cloud. In this work, the authors deployed models to the PaaS type cloud to access the prediction system on mobile. In such cases, the accuracy or F1-score performance is not the sole criterion, but we have to check the number of parameters used in models, the size of the model, and the depth of the models. Deploying models to the cloud includes cost and prediction time or latency. Table 4 tabulates the comparisons of the CNN and DCNN models.

The CNN-Adam-Aug has only 15 layers less compared ResNet152V2-Aug (152), InceptionResNetV2-Aug (572), DenseNet121-Aug (121), and DenseNet201-Aug (201), while the number of trainable parameters was used in CNN-Adam-Aug was 427,317, which are less than other models. The CNN-Adam-Aug size is 1.68MB, the smallest compared

Table 4 Comparison of CNN with DCNN models

Models	Layers	Number of Parameters	Size (MB)	F1-score
ResNet152V2-Aug	152	60,380,648	229.74	87%
InceptionResNetV2-Aug	572	55,873,736	221.48	84%
DenseNet121-Aug	121	8,062,504	30.85	90%
DenseNet201-Aug	201	20,242,984	77.14	94%
CNN-Adam-Aug	**15**	**427,317**	**1.68**	**99%**

ResNet152V2-Aug (229.74MB), InceptionResNetV2-Aug (221.48), DenseNet121-Aug (30.85MB), and DenseNet201-Aug (77.14MB) whereas F1-score of the CNN-Adam-Aug was 99%, which is highest among the models.

The suggested CNN-Adam-Aug model is appropriate for the citrus leaves illness forecast because of its compact size, fewer parameters, and higher F1 score. The CNN-Adam-Aug efficacy assessments have proven entirely accurate in forecasting citrus leaf disease. This paper deploys the CNN model with Adam optimizer to the Heroku cloud. Based on the successful deployment of the model, the Heroku cloud link was generated. The website opens with a single tap on the link for mobile phones to submit a citrus leaf image. After uploading the image to the cloud, the algorithm appropriately forecasts citrus leaf illness.

5 Discussions

Accurate identification of the disease in the early stage plays a critical role in the agricultural domain. The conventional method, like the continuous eye tracking experience, can help distinguish conditions, but it is time-consuming and costly. Several computer-based techniques have been established for recognizing horticultural and agricultural illnesses to solve human errors in conventional methods. The proposed CNN-Adam-Aug (CNN) performed well compared to the literature reported work in Table 5.

From Table 5, it can be seen that most researchers created ML-based methods and used SVM, RF, CNN, VGG16, and others, to diagnose illnesses and get satisfactory outcomes. Revathi et al. [33], Szczypiski et al. [35], Pydipati et al. [36], and Qadri et al. [38] used HPCCDD, ANN, CCM, and MLP methods and achieved accuracies between 90.6% to 98.6%, but we achieved a 99% F1 score and also reported model size. The Parraga-Alava et al. [39], Yang et al. [40], and Guo et al. [41] used VGG16 and achieved lower accuracies compared to our work and models size was not reported.

Sladojevic et al. [57] suggested a CNN framework to classify healthy and diseased leaves of various plant species and achieved 96.3%, but CNN size was not reported. Wu et al. [58] developed a Probabilistic Neural Network (PNN) recognition algorithm for plant recognition and achieved an accuracy of 90%. Both methods achieved less accuracy than our proposed work. AlexNet and ResNet models with and without data augmentation were proposed by Luaibi et al. [59]. Data augmentation was applied to increase the number of training images. A collection of 200 photos of diseased and healthy citrus leaves was made. This method achieved 95.83% and 97.20% accuracy using ResNet and AlexNet, respectively. They predicted healthy and diseased leaf means two-class problems still needed to achieve more accuracy than our work. Furthermore, ResNet and AlexNet are more extensive than our proposed CNN model.

Khattak et al. [60] suggested a CNN to identify healthy citrus fruits and leaves such as black spots, cankers, scabs, greening, and Melanose. By combining different layers, the suggested CNN obtains complementing discriminative characteristics. The CNN model outperformed numerous DCNNs, with a precision of 94.55%. They achieved less accuracy than our work. Elaraby et al. [61] presented the AlexNet and VGG19 techniques for detecting citrus illness pictures from gallery databases, infected scale citrus image data sources, and plant villages. Citrus illnesses, including anthracnose, black spot, canker, scab, greening, and melanose, were identified and classified and achieved up to 94% accuracy. This work also has less accuracy than ours, and AlexNet and VGG19 sizes are enormous compared to our CNN model. Dananjayan et al. [62] access the ability of cutting-edge DCNN

Table 5 list of literature-reported works

References	Dataset/disease	Methods	Performances	Remarks
Xiang et al. [5]	PlantVillage and plant disease severity datasets	CNN	Accuarcy=97.9% and 90.6%	A similar method was used, but the accuracy could be better than our work.
Revathi et al. [33]	Cotton diseases	HPCCDD	Accuracy=98.1%	Accuracy could be better than our work.
Patil et al. [34]	Fungal diseases in sugar cane	SVM	Accuracy=98.6%	Performance is good
Szczypiski et al. [35]	Barley varieties	ANN	Accuracy=86%	Performance is poor
Pydipati et al. [36]	Normal and diseased citrus leaves with greasy spot, melanose, and scab	CCM	Accuracy=95%	Performance is good
Qadri et al. [38]	Eight types of lemon leaves	MLP	Overall accuracy = 98.1%	Performance is good
Parraga-Alava et al. [39]	Lemons leaf image dataset	VGG16	Average rates of 81% and 97%	A similar method was used, but performance is poor than our work.
Yang et al. [40]	2500 complex background images	Mask R-CNN and VGG16	Average accuracy =91.5%	A similar method was used, but performance is poor than our work.
Guo et al. [41]	Plant illness dataset	VGG16 and RPN	Accuracy = 83.57%	A similar method was used, but performance is poor than our work.
Kolivand et al. [43]	Flavia and Acer databases	Venation identification approach	Average accuracy= 98.6% (Flavia) and 89.8% (Acer)	Performance is good
Puri et al. [44]	Leafsnap Dataset	Mask analysis and an SVM	Highest accuracy of 90.27%	Performance is poor
Vilasini et al. [45]	Indian leaf varieties	KNN, SVM, and CNN	100%, 92%, and 80% accuracy for the 2, 5, and 10 classes	Used a similar method, but performance is poor for 5 classes compared to our work
Deepalakshmi et al. [46]	Diseased and healthy leaves of different plants	CNN	Accuracy = 94.5%	A similar method was used, but performance is poor than our work.
Dang-Ngoc et al. [47]	Citrus leaf illnesses (canker, greening, sooty mould, and leaf-miner)	SVM	Recognition rate of 92.5% and a high accuracy rate of 91.76%	Performance is poor

Table 5 (continued)

References	Dataset/disease	Methods	Performances	Remarks
Sladojevic et al [57]	Healthy and diseased leaves of various plant species	CNN (CaffeNet)	Accuracy=96.3%	Performance is good
Wu et al. [58]	Plant Classification	PNN	Accuracy=90%	Performance is poor
Luaibi et al. [59]	Citrus leaf dataset of 200 images	AlexNet and ResNet	95.83%	Two class problems that still achieved less accuracy than our work. Furthermore, ResNet and AlexNet are larger than our proposed CNN model.
Khattak et al. [60]	healthy citrus fruits and leaves	CNN	94.55%	A similar method was used, but performance is poor than our work.
Elaraby et al. [61]	Citrus illness picture gallery databases	AlexNet and VGG19	94%	A similar method was used, but performance is poor than our work. AlexNet and VGG19 sizes are enormous compared to our CNN model
Dananjayan et al. [62]	CCL'20 dataset	YOLOv4, Foveabox, CenterNet, DetectoRS, Faster-RCNN,Cascade -RCNN, and Deformable	78.5% to 98%	Used a similar method, but models sizes not reported
Janarthan et al. [63]	citrus plant illnesses	deep Siamese network	94.04%	Used a similar method, but performance is poor than our work
Pan et al. [64]	Anthracnose, Black spot, canker, scab, sand rust, and greening from 2,097 pictures	DenseNet	88%	Used a similar method, but performance is poor than our work
Zhang et al. [65]	Canker illness	GANs and AlexNet	90.9%	Performance is poor
Present Work	**Citrus leaf dataset** [37]	**CNN,** ResNet152V2, Inception-ResNetV2, DenseNet121, and DenseNet201	**F1-score=99%** **AUC=0.99**	CNN with 15 layers, 427317 parameters, and 1.68MB size. CNN deployed on PaaS cloud successfully

(YOLOv4, Foveabox, CenterNet, DetectoRS, Faster-RCNN, Cascade-RCNN, and Deformable Detr) to identify citrus illness utilizing labelled leaf images optically on the CCL'20 dataset. According to the results, the Scaled YOLOv4 P7 accomplishes rapid and early illness detection and CenterNet2 with Res2Net 101 DCN-BiFPN detects the beginning stages of citrus leaf illnesses with the highest average precision and an average recall of 78.5% and 98%, but the model size was enormous. Janarthan et al. [63] developed a tool incorporating embedding modules, deep neural network and patch subsystem to classify four unique citrus plant illnesses using 609 pictures. The background removal and data augmentation techniques were employed to remove the noise and enhance dataset images artificially. The deep Siamese network was used for training and obtained accuracy of 94.04%. The training of the network requires modifying approximately 2.3 million parameters. This work has lesser accuracy than our work and also uses more number trainable parameters. Pan et al. [64] proposed a DCNN-based method for identifying illnesses such as Anthracnose, Black spot, canker, scab, sand rust, and greening from 2,097 pictures. Data augmentation techniques were employed to increase the number of images. The DenseNet model was used to extract and classify features, and it attained an accuracy of 88%. This work has lesser accuracy than ours and the vast DenseNet model size. Zhang et al. [65] developed GANs (Generative adversarial networks) and an AlexNet-based approach for diagnosing canker illness, achieving 90.9% accuracy and 86.5% recall. This work has lesser accuracy than ours and the vast AlexNet model size.

From the above discussion, it is observed that most researchers used ML classifiers, CNN and DCNN methods to detect the healthy and diseased citrus leaf images and achieved the highest 98% accuracy. Most researchers used DCNN models with more layers, more trainable parameters and huge model sizes. Some of the researchers reported results in precision and recall metrics. This work used the imbalanced citrus leaf dataset with five classes, and the F1-score is the best metric to evaluate the model performance in addition to the accuracy, precision, recall, and AUROC. The F1-score of the proposed CNN was 99%. The model size should be small to deploy to the cloud because of the cost involved. In this work, the authors developed the CNN model so that the trainable parameters, size of the models and depth are kept less without affecting the performance.

6 Conclusions

Designing a highly accurate disease prediction system is challenging work. Therefore, ML/DL approaches are applied in agriculture to help detect plant diseases. Research on developing the precise technique has been attempted and has found promising results. This research provided a complete description of citrus disorders of commercial consequences, paving the path for future investigation in modern agriculture and contributing to the knowledge base in the study.

The authors developed five methods: CNN, ResNet152V2, InceptionResNetV2, DenseNet121, and DenseNet201. The augmentation method was applied to enhance the dataset images because the dataset has only 609 images. Two optimizers (Adam and RMSprop) were used, and it was found that CNN with Adam optimizer and augmentation achieved precision, recall, and an F1 score of 99%, which is the highest in the literature-reported work. On the other hand, ResNet152V2, InceptionResNetV2, DenseNet121, and DenseNet201 achieved F1-score of 87%, 84%, 90%, and 94%, respectively. The

DenseNet201 with Adam optimizer and augmentation achieved acceptable performance, but fine-tuned CNN outperformed other models.

In this work, the CNN model was deployed to the PaaS cloud. Due to this, the authors have also checked the model's size, the number of layers, and the number of trainable parameters. CNN has a 1.68MB size, 427317 trainable parameters, and 15 layers, which is also lower in the literature-reported work. The AUC-ROC performance also checked and found that 0.96, 1, 1, 0.99, and 1 for Black spot, Melanose, cancer, greening, and healthy class, which indicates the CNN-Adam-Aug performs better for predicting all categories images. The average AUROC score was 0.99, which was quite good. Moreover, the authors have successfully deployed the CNN model to the PaaS cloud and tested the same on all types of mobiles. No authors tried such a novel approach that directly helps the farmers.

However, the suggested CNN model was successfully adopted for citrus illness identification. The present study has limitations, like a small dataset (609 photos). Generally trained models on small datasets are subject to overfitting, resulting in erroneous assessment. As a result, the data augmentation method was used to enhance the photos artificially. However, augmentation introduces a new issue: if the source data consists of biases, augmented images will have the same issue. As a result, choosing the appropriate augmentation technique is a difficult task.

In the future, the possibility to create larger datasets or examine new picture augmentation methods to increase the number of photos in the dataset allows for constructing a more suitable model under challenging circumstances. The authors will also look into different forms of disorders and different combinations of the TL models. The authors will expand this research to identify more citrus types with fruit diseases in the future.

Funding The authors did not receive support from any organization for the submitted work.

Data availability Data is available on Mandely website (link: https://data.mendeley.com/datasets/3f83gxmv57/2)

Declarations

Conflicts of interest The authors declare that they have no conflict of interest. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Zhang S, You Z, Wu X (2019) Plant disease leaf image segmentation based on superpixel clustering and EM algorithm. Neural Comput & Applic 31:1225–1232. https://doi.org/10.1007/s00521-017-3067-8
2. Iqbal Z, Khan MA, Sharif M et al (2018) An automated detection and classification of citrus plant diseases using image processing techniques: a review. Comput Electron Agric 153:12–32. https://doi.org/10.1016/j.compag.2018.07.032
3. Vishnoi VK, Kumar K, Kumar B (2021) Plant disease detection using computational intelligence and image processing. J Plant Dis Prot 128:19–53. https://doi.org/10.1007/s41348-020-00368-0
4. Ahila Priyadharshini R, Arivazhagan S, Arun M, Mirnalini A (2019) Maize leaf disease classification using deep convolutional neural networks. Neural Comput & Applic 31:8887–8895. https://doi.org/10.1007/s00521-019-04228-3
5. Xiang S, Liang Q, Sun W et al (2021) L-CSMS: novel lightweight network for plant disease severity recognition. J Plant Dis Prot 128:557–569. https://doi.org/10.1007/s41348-020-00423-w

6. Thomas S, Kuska MT, Bohnenkamp D et al (2018) Benefits of hyperspectral imaging for plant disease detection and plant protection: a technical perspective. J Plant Dis Prot 125:5–20. https://doi.org/10.1007/s41348-017-0124-6
7. Nguyen LD, Lin D, Lin Z, Cao J (2018) Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In: 2018 IEEE international symposium on circuits and systems (ISCAS). IEEE, Florence, pp 1–5
8. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2016) Inception-v4, inception-ResNet and the impact of residual connections on learning. https://doi.org/10.48550/ARXIV.1602.07261
9. Singh UP, Chouhan SS, Jain S, Jain S (2019) Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. IEEE Access 7:43721–43729. https://doi.org/10.1109/ACCESS.2019.2907383
10. Chouhan SS, Kaul A, Singh UP, Jain S (2018) Bacterial foraging optimization based radial basis function neural network (BRBFNN) for identification and classification of plant leaf diseases: an automatic approach towards plant pathology. IEEE Access 6:8852–8863. https://doi.org/10.1109/ACCESS.2018.2800685
11. Uğuz S, Uysal N (2021) Classification of olive leaf diseases using deep convolutional neural networks. Neural Comput Applic 33:4133–4149. https://doi.org/10.1007/s00521-020-05235-5
12. Boeing H, Bechthold A, Bub A et al (2012) Critical review: vegetables and fruit in the prevention of chronic diseases. Eur J Nutr 51:637–663. https://doi.org/10.1007/s00394-012-0380-y
13. Gorinstein S, Caspi A, Libman I et al (2006) Red grapefruit positively influences serum triglyceride level in patients suffering from coronary atherosclerosis: studies in vitro and in humans. J Agric Food Chem 54:1887–1892. https://doi.org/10.1021/jf058171g
14. Atlas of African agriculture research and development: Revealing agriculture's place in Africa. https://ebrary.ifpri.org/digital/collection/p15738coll2/id/128169. Accessed 24 Oct 2022
15. Radhika G, Sudha V, Mohan Sathya R et al (2008) Association of fruit and vegetable intake with cardiovascular risk factors in urban south Indians. Br J Nutr 99:398–405. https://doi.org/10.1017/S0007114507803965
16. Ellis B, Daly DC, Hickey LJ et al (2009) Manual of leaf architecture.
17. Miller SA, Beed FD, Harmon CL (2009) Plant disease diagnostic capabilities and networks. Annu Rev Phytopathol 47:15–38. https://doi.org/10.1146/annurev-phyto-080508-081743
18. Thangaraj R, Anandamurugan S, Kaliappan VK (2021) Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. J Plant Dis Prot 128:73–86. https://doi.org/10.1007/s41348-020-00403-0
19. Prilianti KR, Anam S, Brotosudarmo THP, Suryanto A (2020) Real-time assessment of plant photosynthetic pigment contents with an artificial intelligence approach in a mobile application. J Agricult Engineer 51:220–228. https://doi.org/10.4081/jae.2020.1082
20. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/ARXIV.1409.1556.
21. Alom MZ, Hasan M, Yakopcic C et al (2021) Inception recurrent convolutional neural network for object recognition. Mach Vis Appl 32:28. https://doi.org/10.1007/s00138-020-01157-3
22. Lanjewar MG, Parab JS, Shaikh AY (2022) Development of framework by combining CNN with KNN to detect Alzheimer's disease using MRI images. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-13935-4
23. Too EC, Yujian L, Njuki S, Yingchun L (2019) A comparative study of fine-tuning deep learning models for plant disease identification. Comput Electron Agric 161:272–279. https://doi.org/10.1016/j.compag.2018.03.032
24. Meshram V, Patil K, Meshram V et al (2021) Machine learning in agriculture domain: a state-of-art survey. Artificial Intell Life Sci 1:100010. https://doi.org/10.1016/j.ailsci.2021.100010
25. Santos L, Santos FN, Oliveira PM, Shinde P (2020) Deep learning applications in agriculture: a short review. In: Silva MF, Luís Lima J, Reis LP et al (eds) Robot 2019: fourth Iberian robotics conference. Springer International Publishing, Cham, pp 139–151
26. Lanjewar MG, Parab JS, Shaikh AY, Sequeira M (2022) CNN with machine learning approaches using ExtraTreesClassifier and MRMR feature selection techniques to detect liver diseases on cloud. Cluster Comput. https://doi.org/10.1007/s10586-022-03752-7
27. Al Bashish D, Braik M, Bani-Ahmad S (2011) Detection and classification of leaf diseases using K-means-based segmentation and neural-networks-based classification. Information Technology J 10:267–275. https://doi.org/10.3923/itj.2011.267.275
28. Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: a survey. Comput Electron Agric 147:70–90. https://doi.org/10.1016/j.compag.2018.02.016

29. Singh V, Misra AK (2017) Detection of plant leaf diseases using image segmentation and soft computing techniques. Inform Process Agric 4:41–49. https://doi.org/10.1016/j.inpa.2016.10.005
30. Prajapati BS, Dabhi VK, Prajapati HB (2016) A survey on detection and classification of cotton leaf diseases. In: 2016 international conference on electrical, electronics, and optimization techniques (ICEEOT). IEEE, Chennai, India, pp 2499–2506
31. Siddiqi MH, Sulaiman S, Faye I, Ahmad I (2009) A real time specific weed discrimination system using multi-level wavelet decomposition. Int J Agricult Biol (Pakistan)
32. Hamuda E, Glavin M, Jones E (2016) A survey of image processing techniques for plant extraction and segmentation in the field. Comput Electron Agric 125:184–199. https://doi.org/10.1016/j.compag.2016.04.024
33. Revathi P, Hemalatha M (2012) Classification of cotton leaf spot diseases using image processing edge detection techniques. In: 2012 international conference on emerging trends in science, engineering and technology (INCOSET). IEEE, Tiruchirappalli, Tamilnadu, India, pp 169–173
34. Patil S, Bodhe S (2011) Leaf disease severity measurement using image processing. Int J Eng Technol 3(5):297–301
35. Szczypiński PM, Klepaczko A, Zapotoczny P (2015) Identifying barley varieties by computer vision. Comput Electron Agric 110:1–8. https://doi.org/10.1016/j.compag.2014.09.016
36. Pydipati R, Burks TF, Lee WS (2006) Identification of citrus disease using color texture features and discriminant analysis. Comput Electron Agric 52:49–59. https://doi.org/10.1016/j.compag.2006.01.004
37. Rauf HT, Saleem BA, Lali MIU et al (2019) A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. Data Brief 26:104340. https://doi.org/10.1016/j.dib.2019.104340
38. Qadri S, Furqan Qadri S, Husnain M et al (2019) Machine vision approach for classification of citrus leaves using fused features. Int J Food Prop 22:2072–2089. https://doi.org/10.1080/10942912.2019.1703738
39. Parraga-Alava J, Alcivar-Cevallos R, Riascos JA, Becerra MA (2021) Aphids detection on lemons leaf image using convolutional neural networks. In: Botto-Tobar M, Zamora W, Larrea Plúa J et al (eds) Systems and information sciences. Springer International Publishing, Cham, pp 16–27
40. Yang K, Zhong W, Li F (2020) Leaf segmentation and classification with a complicated background using deep learning. Agronomy 10:1721. https://doi.org/10.3390/agronomy10111721
41. Guo Y, Zhang J, Yin C et al (2020) Plant disease identification based on deep learning algorithm in smart farming. Discret Dyn Nat Soc 2020:1–11. https://doi.org/10.1155/2020/2479172
42. Tsolakidis DG, Kosmopoulos DI, Papadourakis G (2014) Plant leaf recognition using Zernike moments and histogram of oriented gradients. In: Likas A, Blekas K, Kalles D (eds) Artificial intelligence: methods and applications. Springer International Publishing, Cham, pp 406–417
43. Kolivand H, Fern BM, Saba T et al (2019) A new leaf venation detection technique for plant species classification. Arab J Sci Eng 44:3315–3327. https://doi.org/10.1007/s13369-018-3504-8
44. Puri D, Kumar A, Virmani J, Kriti (2022) Classification of leaves of medicinal plants using laws' texture features. Int j inf tecnol 14:931–942. https://doi.org/10.1007/s41870-019-00353-3
45. Vilasini M, Ramamoorthy P (2020) CNN approaches for classification of Indian leaf species using smartphones. Comput, Mat Continua 62:1445–1472. https://doi.org/10.32604/cmc.2020.08857
46. Deepalakshmi P, Prudhvi KT, Siri CS et al (2021) Plant Leaf Dis Detection Using CNN Algorithm: Int J Inform Syst Model Design 12:1–21. https://doi.org/10.4018/IJISMD.2021010101
47. Dang-Ngoc H, Cao TNM, Dang-Nguyen C (2021) Citrus leaf disease detection and classification using hierarchical support vector machine. In: 2021 international symposium on electrical and electronics engineering (ISEE). IEEE, Ho Chi Minh, Vietnam, pp 69–74
48. Bloice DM, Stocker C, Holzinger A (2017) Augmentor: an image augmentation library for machine learning. JOSS 2:432. https://doi.org/10.21105/joss.00432
49. Hauberg S, Freifeld O, Larsen ABL et al (2015) Dreaming more data: class-dependent distributions over diffeomorphisms for learned data augmentation. https://doi.org/10.48550/ARXIV.1510.02795
50. Lanjewar MG, Morajkar PP, Parab J (2022) Detection of tartrazine colored rice flour adulteration in turmeric from multi-spectral images on smartphone using convolutional neural network deployed on PaaS cloud. Multimed Tools Appl 81:16537–16562. https://doi.org/10.1007/s11042-022-12392-3
51. Lanjewar MG, Shaikh AY, Parab J (2022) Cloud-based COVID-19 disease prediction system from X-ray images using convolutional neural network on smartphone. Multimed Tools Appl. https://doi.org/10.1007/s11042-022-14232-w
52. Lanjewar MG, Gurav OL (2022) Convolutional neural networks based classifications of soil images. Multimed Tools Appl 81:10313–10336. https://doi.org/10.1007/s11042-022-12200-y

53. Lanjewar MG, Panchbhai KG (2022) Convolutional neural network based tea leaf disease prediction system on smart phone using paas cloud. Neural Comput & Applic. https://doi.org/10.1007/s00521-022-07743-y
54. Lanjewar MG, Panchbhai KG, Charanarur P (2023) Lung cancer detection from CT scans using modified DenseNet with feature selection methods and ML classifiers. Expert Syst Appl 224:119961. https://doi.org/10.1016/j.eswa.2023.119961
55. Lanjewar MG, Parate RK, Wakodikar R, Parab JS (2023) Detection of starch in turmeric using machine learning methods. In: Kumar S, Sharma H, Balachandran K, Kim JH, Bansal JC (eds) Third congress on intelligent systems. CIS 2022, Lecture notes in networks and systems, vol 613. Springer, Singapore. https://doi.org/10.1007/978-981-19-9379-4_10
56. Srinivasu PN, Bhoi AK, Jhaveri RH et al (2021) Probabilistic deep Q network for real-time path planning in censorious robotic procedures using force sensors. J Real-Time Image Proc 18:1773–1785. https://doi.org/10.1007/s11554-021-01122-x
57. Sladojevic S, Arsenovic M, Anderla A et al (2016) Deep neural networks based recognition of plant diseases by leaf image classification. Comput Intell Neurosci 2016:1–11. https://doi.org/10.1155/2016/3289801
58. Wu SG, Bao FS, Xu EY et al (2007) A leaf recognition algorithm for plant classification using probabilistic neural network. In: 2007 IEEE international symposium on signal processing and information technology. Pp 11–16.
59. Luaibi AR, Salman TM, Miry AH (2021) Detection of citrus leaf diseases using a deep learning technique. IJECE 11:1719. https://doi.org/10.11591/ijece.v11i2.pp1719-1727
60. Khattak A, Asghar MU, Batool U et al (2021) Automatic detection of citrus fruit and leaves diseases using deep neural network model. IEEE Access 9:112942–112954. https://doi.org/10.1109/ACCESS.2021.3096895
61. Elaraby A, Hamdy W, Alanazi S (2022) Classification of citrus diseases using optimization deep learning approach. Comput Intell Neurosci 2022:1–10. https://doi.org/10.1155/2022/9153207
62. Dananjayan S, Tang Y, Zhuang J et al (2022) Assessment of state-of-the-art deep learning based citrus disease detection techniques using annotated optical leaf images. Comput Electron Agric 193:106658. https://doi.org/10.1016/j.compag.2021.106658
63. Janarthan S, Thuseethan S, Rajasegarar S et al (2020) Deep metric learning based citrus disease classification with sparse data. IEEE Access 8:162588–162600. https://doi.org/10.1109/ACCESS.2020.3021487
64. Pan W, Qin J, Xiang X et al (2019) A smart Mobile diagnosis system for citrus diseases based on densely connected convolutional networks. IEEE Access 7:87534–87542. https://doi.org/10.1109/ACCESS.2019.2924973
65. Zhang M, Liu S, Yang F, Liu J (2019) Classification of canker on small datasets using improved deep convolutional generative adversarial networks. IEEE Access 7:49680–49690. https://doi.org/10.1109/ACCESS.2019.2900327