

# Leveraging Convolutional Neural Network and Transfer Learning for Cotton Plant and Leaf Disease Recognition

**Md. Rayhan Ahmed**

Department of Computer Science and Engineering, Stamford University Bangladesh, Dhaka-1217, Bangladesh.  
E-mail: rayhansimanto@gmail.com ; rayhansimanto@stamforduniversity.edu.bd

Received: 06 February 2021; Revised: 01 March 2021; Accepted: 20 March 2021; Published: 08 August 2021

**Abstract:** Automatic Recognition of Diseased Cotton Plant and Leaves (ARDCPL) using Deep Learning (DL) carries a greater significance in agricultural research. The cotton plant and leaves are severely infected by a disease named Bacterial Blight-affected by bacterium, *Xanthomonas axonopodis* pv. *Malvacearum* and a new rolling leaf disease affected by an unorthodox leaf roll dwarf virus. Existing research in ARDCPL requires various complicated image preprocessing, feature extraction approaches and cannot ensure higher accuracy in their detection rates. This work suggests a Deep Convolutional Neural Network (CNN) based DCPLD-CNN model that achieves a higher accuracy by leveraging the DL models ability to extract features from images automatically. Due to the enormous success of numerous pre-trained architectures regarding several image classification task, this study also explores eight CNN based pre-trained architectures: DenseNet121, NasNetLarge, VGG16, VGG19, ResNet50, InceptionV3, InceptionResNetV2, and Xception models by Fine-Tuning them using Transfer Learning (TL) to recognize diseased cotton plant and leaves. This study utilizes those pre-trained architectures by adding extra dense layers in the last layers of those models. Several Image Data Augmentation (IDA) methods were used to expand the training data to increase the model's generalization capability and reduce overfitting. The proposed DCPLD-CNN model achieves an accuracy of 98.77% in recognizing disease in cotton plant and leaves. The customized DenseNet121 model achieved the highest accuracy of 98.60% amongst all the pre-trained architectures. The proposed method's feasibility and practicality were exhibited by several simulated experimental results for this classification task.

**Index Terms:** Cotton Plant Leaf Disease Recognition, Deep Learning, CNN, Transfer Learning, Image Data Augmentation.

## 1. Introduction

Cotton, often known as the “silver fiber,” is among the essential fiber crops used as raw resources in the clothing manufacturing industry. While the garment sector of Bangladesh is growing each day, it requires a tremendous amount of imported cotton since the amount of cotton produced internally is relatively low compared to the demand of the ever-growing textile industry in Bangladesh, costing a significant amount of foreign currency. However, the cotton leaf and plant face many challenges regarding numerous diseases such as Bacterial Blight, Root Knot Nematode, Fusarium Wilt, Root Rot, Verticillium Wilt, and recently a new disease called rolling-leaf-hampering its growth, and the conventional technique that is now embraced is that an expert crop pathologist can diagnose the disease by optical inspection of infected plants [1], incorporating a substantial amount of complexity regarding time, money and effective solution.

The emergence of Deep Learning (DL) and Computer Vision (CV) has unlocked the way to automatically detect crop plant disease through its exceptional feature extracting power and complex computational capacity. Embedding a DL model in a simplified graphical user interface-based mobile application enables farmers with even little educational knowledge to detect infected cotton plant and leaves without an expert crop pathologist's aid. Before the surge of DL approaches, researchers mostly used image processing techniques and conventional Machine Learning (ML) models to identify infected plants or leaves, see for reference, [2–7]. The biggest challenge for these models is manually extracting a substantial degree of required features for the classification task. DL methods solve this problem due to their automatic feature extraction aptitude from raw representations of input data during training the model using CNN.

Using Transfer Learning (TL), we can train our DL model to achieve greater accuracy. TL is an advancement of unearthing in a novel job by carrying intelligence from a similar task that was previously grasped [8]. It is reusing a pre-trained architecture on a novel problem on hand. Transfer learning is typically performed for tasks where the dataset does not have enough data to train a full-scale DL model from scratch. Fine-tuning the final neural network layers of a

pre-trained model to a relatively new set of data has been the popular choice of tools for scholars worldwide. Researchers who used TL and publicly available plant disease dataset like Plant Village [9] include [10–13].

However, the shortage of proper real-life plant image dataset of various kinds is an issue. One of the significant ways to increase the quantity of the dataset is to use Image Data Augmentation (IDA) [14]. The Supervised DL models' predictability mainly depends on the quantity and diversity of data available during training. Some of the typical IDA techniques are Flipping, Scaling, Rotation, Translation, Cropping, Padding, Brightness, Contrast, Saturation, Hue, Gaussian Noise, and Generative Adversarial Networks (GAN) [15]. This study adopts some of these IDA techniques to increase the no. of image quantity to train the models.

Manually inspecting plants for disease recognition and diagnosis is tedious, laborious, expensive, and to some extent, subjective. Swift developments of DL-based technology have facilitated significant feats in detecting plant disease in recent years. For this reason, researchers worldwide are adopting DL approaches as an alternative method for detecting disease in numerous plants. This study aims to solve the automatic recognition of newly discovered rolling-leaf disease as well as bacterial blight disease of cotton leaves and plants using a stack deep CNN model. A comparative analysis of eight successful pre-trained architectures through customization of final layers was also evaluated to test these models' suitability regarding diseased cotton plant and leaves recognition.

## 2. Related Work

A substantial percentage of works have been conducted on problems with plant disease recognition. The expansion of Image Processing (IP), Computer Vision (CV), ML, and DL opened the door for automated recognition of the diseased plant due to their extensive feature processing performance. There has not been extensive research on specific disease recognition in cotton plant and leaves using ML and DL methods. One reason is the lack of a cotton plant-specific dataset. For this study, the literature review is conducted in two ways: ML-based and DL-based methods used in detecting diseases in plant and leaves.

### 2.1 Machine Learning (ML) based Methods

Researchers worldwide have produced several accomplishments in the identification and prevention of plant diseases through ML technology. Some of the existing study and research mechanisms have been formulated as follows.

Cheng et al. [16] presented a feature-based method that differentiates weeds from rice plants. They have extracted the features like the leaf tips and rice ear using the Harris Corner Algorithm, then applied ML algorithms for the classification. Decision Tree performed best with a precision of 98.2% and recall of 97.7%.

Patil et al. [17] proposed a Support Vector Machine (SVM)-based cotton leaf spot detection model. Acquired images were segmented, and color and shape features were extracted from the segmented portion before feeding these to the SVM classifier.

Shah and Jain [18] proposed an Artificial Neural Network (ANN)-based approach to detect diseased cotton leaves. Image segmentation was performed using hue saturation value and RGB components of the diseased images.

Al-Tarawneh et al. [19] proposed olive disease detection and severity rating mechanism based on clustering algorithm of K-Means (KMC) and Fuzzy C-Means (FCM) algorithm. They have preprocessed the images using polygon cropping and applied grayscale representation to define the masked polygon region. The true acceptance rate obtained with these methods was 66% for KMC and 86% for FCM.

Adeel et al. [20] recommended a model for dissection and identification of grape leaf diseases by using the popular Plant Village dataset. They have recommended a low contrast haze reduction scheme reducing noise and enhancing the diseased region, following a segmentation method for detecting the disease. Their SVM based method achieved 94.1% accuracy after Neighborhood Component Analysis based feature reduction.

Zhang et al. [21] demonstrated a sparse depiction classification centered cucumber disease detection from its plant leaf. Their proposed method consists of three parts. At first, they have segmented the diseased leaf images by means of the k-means clustering system, secondly from the lesion information, they have extracted the color and shape features of those plants; finally, they have classified the diseased leaf from the fresh ones using sparse representation. Their sparse representation-based method achieved the best recognition rate of 91.25% for gray mold disease. For other types of diseases, the recognition rate varies from 82.36% to 88.43%.

Plant diseases can be identified and categorized by examining the color, shape, and texture from the images of diseased or infected leaves, see references [22–24]. Ashourloo et al. [25] in their investigation used spectral vegetation guides and numerous regression techniques to identify rust disease of wheat leaves. They have used the SVR, and PLSR, and Gaussian Process Regression-GPR algorithms for detecting rust in wheat leaf and compared performance among them.

Sahoo et al. [26] proposed an image-based study of twelve different leaves to identify existing dead spots. They used various image segmentation procedures for the task, including gradient, magnitude, hue saturation value, grey threshold, and salient feature-based analysis using K-means. The diseases were classified after the images were segmented based on a predefined color spectrum.

Using SVM along with spectral crop sequences related to physiological parameters, Rumpf et al. [27] proposed an early disease recognition mechanism of sugar beet leaf. A summary of the reviewed ML and Image Processing based diseased plant leaf recognition system is presented in Table 1.

Table 1. Studies using ML techniques for plant and leaf disease recognition.

Ref.	Machine Learning Algorithm	Feature Selection	Dataset	Evaluation Metric
[16]	Decision Tree, SVM, Naïve Bayes	Color, and Surface	Self-Collected	98.2%-Precision 97.7%-Recall
[17]	SVM	Color, Shape	Not mentioned	Not mentioned
[18]	ANN	Color		Relative Error = 0.051.
[19]	K-means, Fuzzy C-means	None		66% and 86%
[20]	SVM	Color, Surface, and Geometric features	Plant Village	94.1%-Accuracy
[21]	K-means	Color	Self-Collected	91.25%-Accuracy
[22]	Classification Tree	Color, Shape, and Surface		97%-Accuracy
[23]	K-means	Color, and Surface		94%-Precision
[24]	K-means	Color		84%-Accuracy
[25]	SVR, GPR, PLSR	Color, Surface, and Shape		Co-efficient of determination $R^2 = 0.98$
[26]	K-means	Color		Not mentioned
[27]	SVM	Spectral Vegetation guides		97%-Accuracy

## 2.2 Deep Learning (DL) based Methods

The automatic feature extraction techniques of DL methods have become further operative for classification, segmentation, and detection-related tasks regarding images. Out of many methods, CNN is the most preferred choice of researchers to carry out those tasks for plant disease recognition and categorization. Some of the existing research and research mechanisms in plant disease detection based on DL have been formulated as follows.

Combining dilated convolution and global pooling, Zhang et al. [28], suggested a cucumber leaf disease recognition model. Their study's significant influences are reducing the training parameter, replacing fully connected layers of conventional CNN model with global pooling mechanism to escalate the receptive area of convolution without dropping feature details. The accuracy obtained was 94.75%.

Uğuz et al. [29] presented a CNN based olive leaf disease recognition. They have also used pre-trained VGG16 and VGG19 transfer learning models for the study. A comparative summary was also presented on the model's accuracy with and without IDA, where IDA based model yields better results with a precision value of 95%.

Barbedo et al. [30] proposed a DL method that focuses on the spots and individual lesions for plant disease recognition instead of considering the whole plant leaf. They have used a diverse dataset for this task used 1567 images covering eighty-nine diseases affecting fourteen plant species. They have used the pre-trained GoogLeNet architecture with ten-fold cross-validation for this study and accomplished a state of the art accuracy ranging from 75% to 98%.

Many researchers [27–29] have used the Plant Village dataset for their work and applied CNN and TL methods like LeNet, AlexNet, and GoogLeNet architecture.

Wu et al. [34] explored three transfer learning methods, ResNet, AlexNet, and GoogLeNet architecture, to detect diseases in soybean leaves. They have performed a comparative study of those methods' performance using batch of different sizes and the various no. of epochs. Among them, ResNet achieved the maximum accuracy of 94.29%.

A CNN-based rice disease recognition system was suggested by Lu et al. [35]. They have learned the features using sparse auto-encoding, and by applying stochastic pooling and convolution, they have made the classification of ten rice disease types from a reduced dataset and got a detection rate of 95%.

Kawasaki et al. [36] presented a novel CNN-based cucumber leaf disease recognition method. They have used a 4-fold cross-validation approach and got an avg. accuracy of 92.5% in the free aspects ratio and 94.9% in the constant aspects ratio dataset.

Jenifa et al. [37] presented a CNN-based model to identify diseased cotton leaf. They have identified four types of cotton leaf disease using a small dataset of only 500 training images and 100 testing images. They have used a four-layer CNN model without mentioning any information regarding strides, padding, polling, or normalization. With such a low amount of images for training without IDA brings the question of overfitting of the model. A summary of the reviewed DL based plant disease recognition system is presented in Table 2.

Table 2. Studies using DL techniques for plant and leaf disease recognition.

Ref.	DL Framework	Transfer Learning Architecture	Dataset	IDA	Accuracy
[28]	CNN	AlexNet	Self-Collected	Yes	94.75%
[29]		VGG16, VGG19		Yes	95%
[30]		GoogLeNet		No	75%-98%
[31]		LeNet	Plant Village	No	92%-98%
[32]		AlexNet, GoogLeNet		No	99%
[33]		AlexNet, GoogLeNet		No	99%
[34]		ResNet, AlexNet, GoogLeNet	Self-Collected	Yes	94.29%
[35]		None		No	95%
[36]		None		No	94.9%
[37]		None		No	96%

In this paper, a Deep CNN based DCPLD-CNN model is projected to detect the diseased cotton plant and leaves from the scratch. The study also analyzes the performance of eight individual pre-trained models: VGG16, VGG19, Xception, ResNet50, NasNetLarge, InceptionV3, InceptionResNetV2, and DenseNet121 by customizing them in the final layer using TL. To the best of the author’s knowledge, all these models were not previously implemented together by customization, and performances were not evaluated to detect cotton plant and leaf disease in a single research paper.

### 3. Materials and Methods

#### 3.1 Study Outline

This study utilizes the CNN architecture for the task. The key benefit of CNN is that it recognizes critical features without human intervention. Since CNN has features such as sharing of parameters as well as dimensionality reduction, the key idea is that what it learns in one part of the image will be expedient in a different part of the image. The computing power needed in CNN is reduced due to the reduction in dimensionality. Those automatically extracted features using techniques such as pooling, strides and padding of CNN are then trained along with augmented image data to build the model for detecting diseased cotton plants and leaves. The methodology is presented visually in Fig. 1.

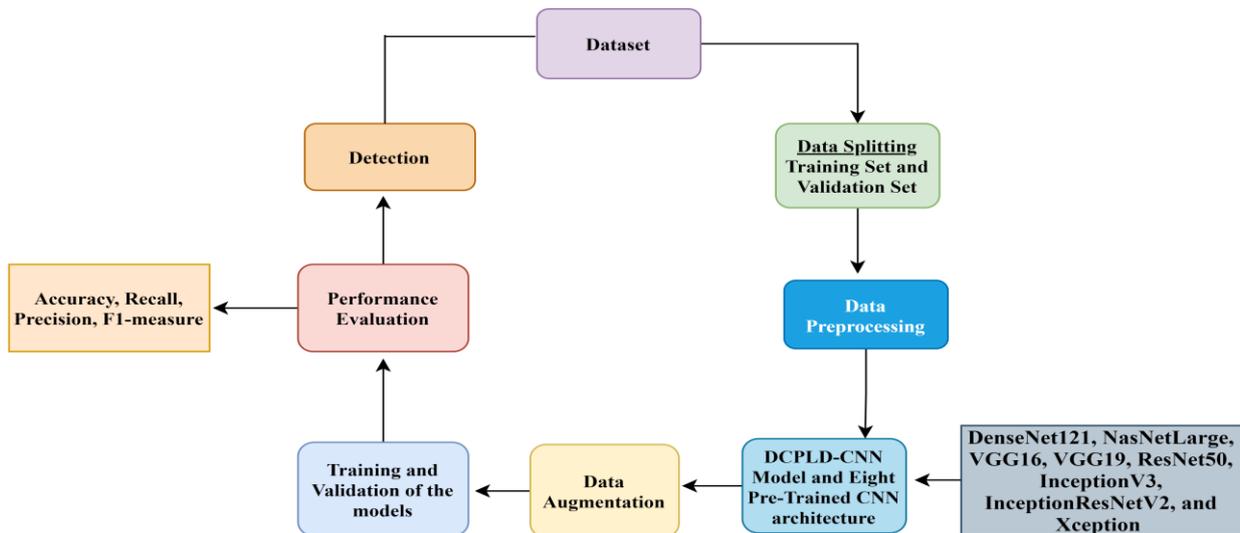


Fig. 1. Proposed methodology of the study.

#### 3.2 Dataset

The dataset used in this study has been collected from [38]. The dataset contains 1951 training images and 324 test images belonging to four classes: Diseased Cotton Leaves, Diseased Cotton Plants, Fresh Cotton Leaves, and Fresh Cotton Plants, see Fig. 2. The diseased plant and leaves are affected with Bacterial Blight-infected by bacterium, Xanthomonas axonopodis pv. Malvacearum and a new rolling leaf disease affected by an unorthodox leaf roll dwarf virus. Since the training images quantity is relatively low, IDA was performed to increase the quantity so that the

proposed Deep CNN model faces diversity during the classification task as well as expands its generalization capacity and reduce overfitting.

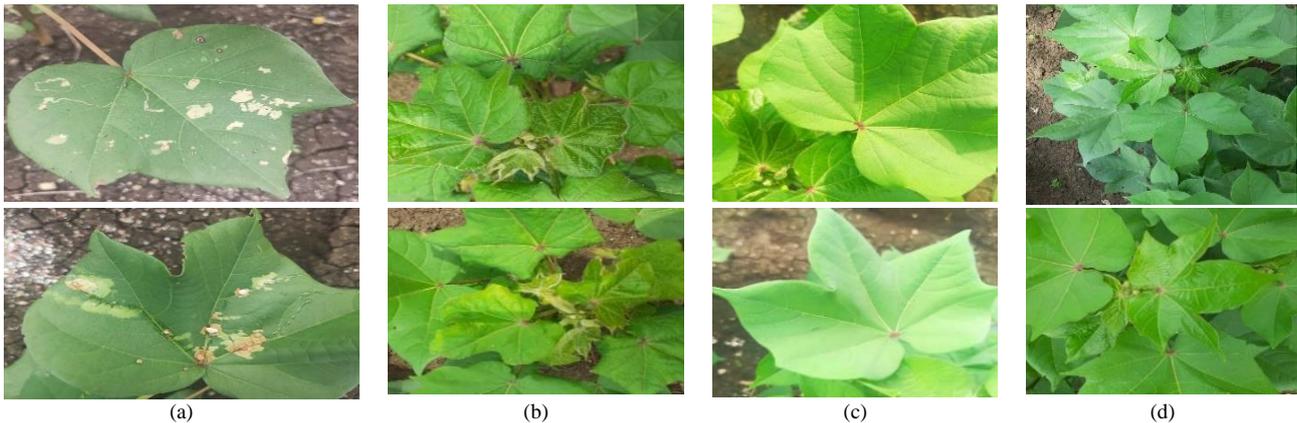


Fig. 2. (a) Diseased cotton leaf, (b) Diseased cotton plant (c) Fresh cotton leaf (d) Fresh cotton plant, from the used dataset.

### 3.3 Image Data Augmentation (IDA)

Using IDA, we create disparities and deviations in the existing training images to increase the images' quantity for the experiment. Two significant IDA variables that affect DL techniques' performance are the augmentation method and augmentation rate [39]. For a small dataset, IDA increases the no. of images necessary for the proper training of the CNN model. For this study, flipping, rotation, shifting, scaling, shearing, and zooming augmentation techniques were applied during training the models. Augmentation parameters along with values used in this experiment is provided in Table 3.

Table 3. IDA techniques used in this study.

Augmented parameter	Description	Range of Values
Scaling	The image is adjusted to the given dimension, e.g., an image width can be halved or doubled.	[-0.2 to 0.2]
Shearing	Sliding the image in vertical or horizontal direction with a given degree.	[-0.2 to 0.2]
Horizontal Flipping	Flipping of the image in left, right direction.	True
Shifting	Shifting allows the objects position in the image to be changed.	[-0.2 to 0.2]
Rotating	The image is rotated in a defined degree.	[-20 to 20]
Zooming	Arbitrarily zooms images.	[0.8 to 1.2]

### 3.4 Transfer Learning (TL) of Pre-Trained Architectures

For this study, eight pre-trained models were used using TL techniques and fine-tuned in the last layer for a comparative analysis of evaluated performance in cotton leaf plant disease recognition task. Existing literature in recognition of the disease in cotton leaves used conventional ML techniques. Since the dataset used in this study is not very large, training could be affected by overfitting. Those pre-trained architectures were used because of their learning through a larger dataset of images. As a result, more features are extracted through this process. TL is a strategy in which the model utilizes learned information during the training of a significantly larger dataset and uses it to train a model with a relatively small number of datasets since plant disease dataset collection in the agricultural field is a time-consuming process due to many reasons.

All the pre-trained architectures' layers are non-trainable except the last layer, whose weights can be updated during training. We can add our custom layers instead of that last layer for more parameters of training. This study modifies all eight TL-based pre-trained architectures and added custom layers on top of non-trainable layers to train the fine-tuned model. Eight pre-trained deep CNN models namely, VGG16, VGG19 [40], InceptionResNetV2 [41], NasNetLarge[42], InceptionV3 [43], Xception [44], ResNet50 [45], and DenseNet121 [46] were explored in this work. A comparative summary is presented in Table 4. Customization of these pre-trained model's was done by training only the last layer of those models. A Fully Connected (FC) layer of 512 artificial neurons were added on top of the non-trainable layers, then another FC layer of 128 artificial neurons, followed by another FC layer of 64 artificial neurons and finally the output layer with four artificial neurons, representing four classes namely, Diseased Cotton Leaves, Diseased Cotton Plants, Fresh Cotton Leaves, and Fresh Cotton Plants, after flattening (i.e. conversion of data into one dimensional array of feature vector for feeding the FC layer for elucidation). Doing this, features that the eight architectures learned during acquiring identification of object features and patterns belonging to thousands of multiple

categories have been retained. Those features and pre-trained weights are then applied to detect cotton plant and leaf disease in the final layers by customizing the architectures. Those pre-trained models are rarely implemented in diseased cotton plant and leaves recognition. Hence, this study tries to experiment with those models and their competence in this task is analyzed. A rudimentary overview of the customization of eight pre-trained architectures is depicted in Fig. 3.

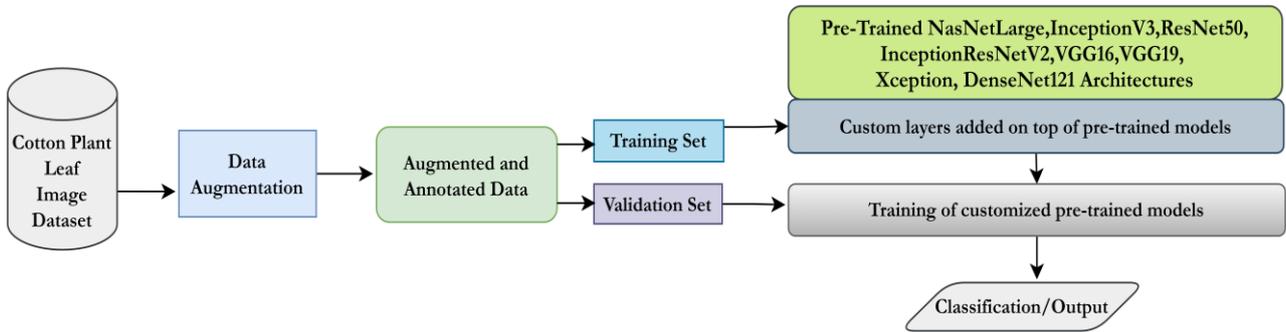


Fig. 3. Rudimentary overview of the training of customized pre-trained architectures.

Table 4. A comparative synopsis of eight pre-trained architectures utilized in this study.

Pre-Trained Architectures	# Parameters	Top Five Accuracy*	Top Five Error rate	Depth	Major Contribution	Year
NasNetLarge	88.9 Million	96%	-	-	Allows transferability by designing a novel search span.	2018
VGG16	138.3 Million	90.01%	7.33%	23	Architectural simplicity with increased testing accuracy compared to previous models.	2014
VGG19	143.6 Million	90.0%	7.31%	26		
Xception	22.9 Million	94.5%	0.05%	126	It is a linear pile of depth-wise differentiable layers of convolution with residual association making it a flexible architecture to modify.	2017
InceptionV3	23.8 Million	94.1%	3.5%	159	Manages the issue of the representational constraint, Balances out the network, Able to represent higher dimensional attributes.	2015
InceptionResNetV2	55.8 Million	95.3%	3.52%	572	A hybrid form of Inception model. Uses split transformation combine concept with residual association.	2017
ResNet50	25.6 Million	92.1%	3.57%	50	Learns only residual information of each layers for detection task. Reduced complexity.	2015
DenseNet121	8 Million	92.3%	5.19%	121	Reduced the vanishing gradient problem and no. of parameters.	2017

\*Top 5 Accuracy states the architecture’s performance during validation on ImageNet dataset [47] and depth determines the model’s topologic depth.

### 3.5 Experimental Setup

A 16 GB RAM laptop with Intel Core i7-8750 CPU consisting of eight cores clocking at 2.20 GHz and running an NVIDIA GeForce GTX 1050-Ti graphics were used for the experiments in this study. The PC was running Windows 10 using Anaconda Python and the TensorFlow and Keras deep learning framework.

### 3.6 Proposed DCPLD-CNN Architecture

A stack five-convolution layer-based CNN model with MaxPooling in each convolution layer followed by dense layers of unit 512, 128, 64 artificial neurons and the final output dense layer with unit 4 artificial neurons is suggested in this paper from scratch. This is a stack CNN centered method for Diseased Cotton Plant Leaf Detection (DCPLD-CNN), without any transfer learning techniques. The architecture and input and output shape of each Convolution, MaxPooling, and FC layers are depicted in Table 5. The arrangement of DCPLD-CNN model is presented in Fig. 4.

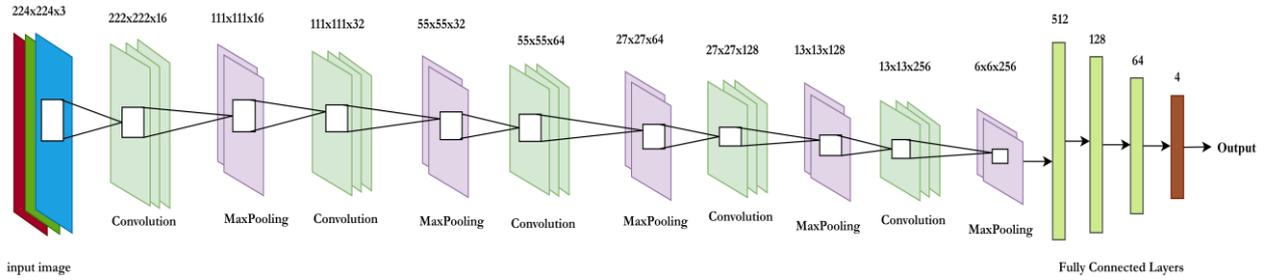


Fig. 4. DCPLD-CNN Model Architecture.

**Model Parameters:**

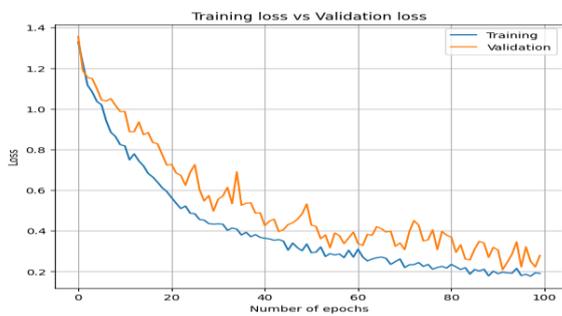
Kernel size = 3; Strides = (1, 1); padding = ‘SAME’; Input Image Shape = (224, 224, 3); MaxPooling pool size = (2,2); The batch size =32; Activation function = ‘Relu’ (i.e. in all the convolutional as well as dense layers apart from the final layer); Output layer’s Activation function = ‘softmax’; Optimizer = ‘Adam’; Learning rate = 0.0001. To lessen the overfitting, Dropout value of 0.10 was used in the convolutional layers and at a level of 0.30 in every dense layer.

Table 5. DCPLD-CNN Network Architecture with filter size, shape of the output tensor, and no. of trainable parameters.

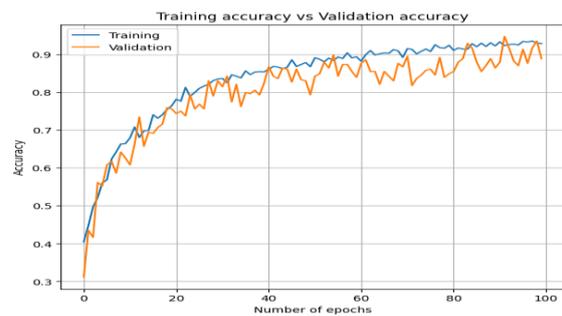
Type of Layers	Filter	Output Shape	# Parameters
Convolution_Layer_1	16	(None, 222, 222, 16)	448
Max_Pooling_Layer_1	16	(None, 111, 111, 16)	0
Convolution_Layer_2	32	(None, 111, 111, 32)	4640
Max_Pooling_Layer_2	32	(None, 55, 55, 32)	0
Convolution_Layer_3	64	(None, 55, 55, 64)	18496
Max_Pooling_Layer_3	64	(None, 27, 27, 64)	0
Convolution_Layer_4	128	(None, 27, 27, 128)	73856
Max_Pooling_Layer_4	128	(None, 13, 13, 128)	0
Convolution_Layer_5	256	(None, 13, 13, 256)	295168
Max_Pooling_Layer_5	256	(None, 6, 6, 256)	0
Dense_Layer_1	512	(None,512)	4719104
Dense_Layer_2	128	(None,128)	65664
Dense_Layer_3	64	(None,64)	8256
Dense_Layer_4	4	(None,4)	260
Total parameters: 5,185,892			

**4. Results**

In this segment, an evaluation based on this study's findings of the proposed DCPLD-CNN model and eight custom TL-based pre-trained models for the diseased cotton plant and leaf recognition is presented. For this study's comparative analysis, the dataset was trained using batch sizes of 32 and 64, for 100 iterations for every customized pre-trained model-based architecture. For the proposed built from scratch DCPLD-CNN model's performance evaluation, training was done for 100 and 500 iterations for two batch sizes of 32 and 64. In Table 6, a comprehensive summary of this study's outcome is presented. For DCPLD-CNN model a state of the art validation accuracy of 98.77% was achieved for 500 iterations and 88.89% for 100 iterations with 32 batch size. The validation accuracy and loss curve for 100 and 500 iterations of the DCPLD-CNN model is presented in Fig. 5-8. The validation accuracy curve of all eight customized pre-trained models for 100 iterations with a 32 batch size is presented in Fig. 9-12.

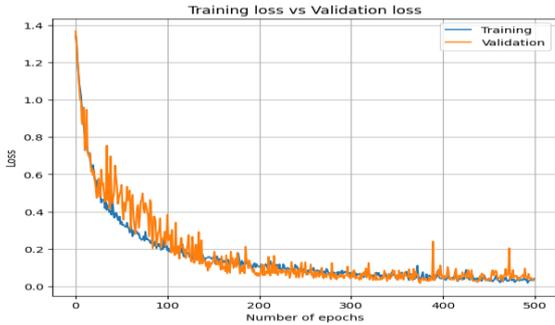


(a) Loss metric curve –DCPLD-CNN-100 iterations-32 batch size

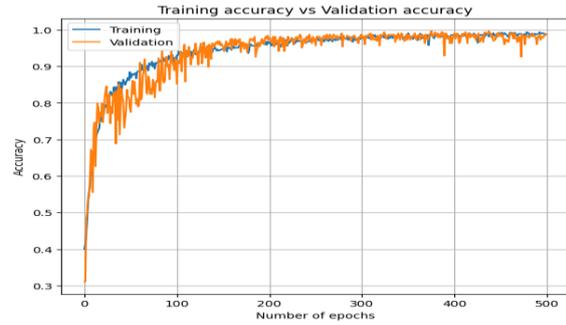


(b) Accuracy metric curve – DCPLD-CNN-100 iterations-32 batch size

Fig. 5. (a) Loss metric curve and (b) Accuracy metric curve of the proposed DCPLD-CNN Architecture. Batch Size=32, Iterations =100.

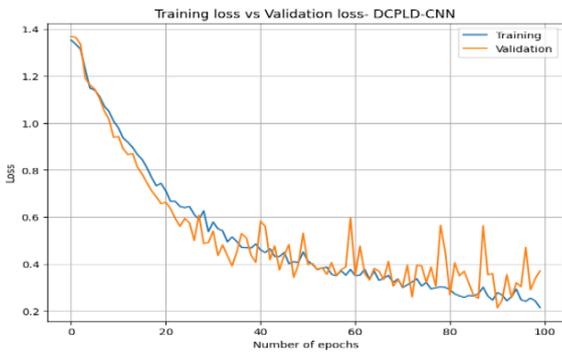


(a) Loss metric curve –DCPLD-CNN-500 iterations-32 batch size

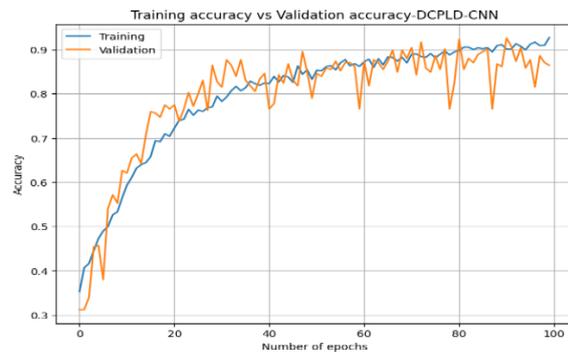


(b) Accuracy metric curve – DCPLD-CNN-500 iterations-32 batch size

Fig. 6. (a) Loss metric curve and (b) Accuracy metric curve of the proposed DCPLD-CNN Architecture. Batch Size=32, Iterations =500.

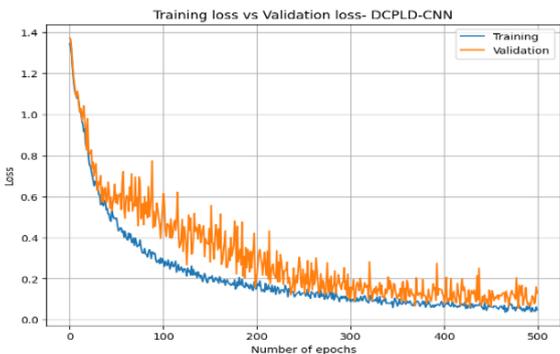


(a) Loss metric curve –DCPLD-CNN-100 iterations -64 batch size

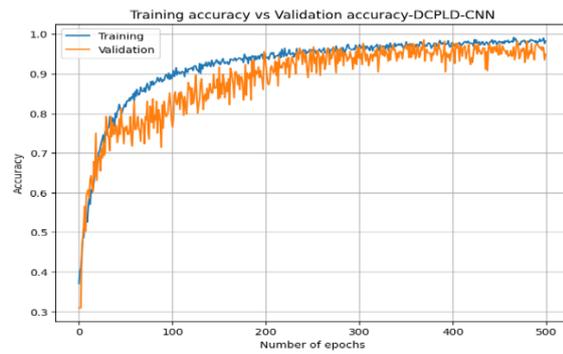


(b) Accuracy metric curve – DCPLD-CNN-100 iterations -64 batch size

Fig. 7. (a) Loss metric curve and (b) Accuracy metric curve of the proposed DCPLD-CNN Architecture. Batch Size=64, Iterations =100.

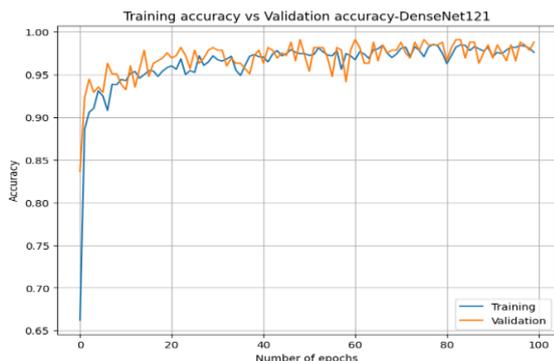


(a) Loss metric curve –DCPLD-CNN-500 iterations -64 batch size

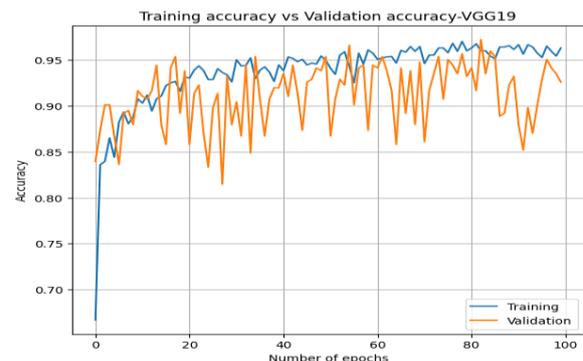


(b) Accuracy metric curve – DCPLD-CNN-500 iterations -64 batch size

Fig. 8. (a) Loss metric curve and (b) Accuracy metric curve of the proposed DCPLD-CNN Architecture. Batch Size=64, Iterations =500.



(a) Accuracy metric curve-Custom-DenseNet121



(b) Accuracy metric curve-Custom-VGG19-CNN

Fig. 9. (a) Loss metric curve and (b) Accuracy metric curve of the Custom-DenseNet121-CNN and Custom-VGG19-CNN Architecture. Iterations =100.

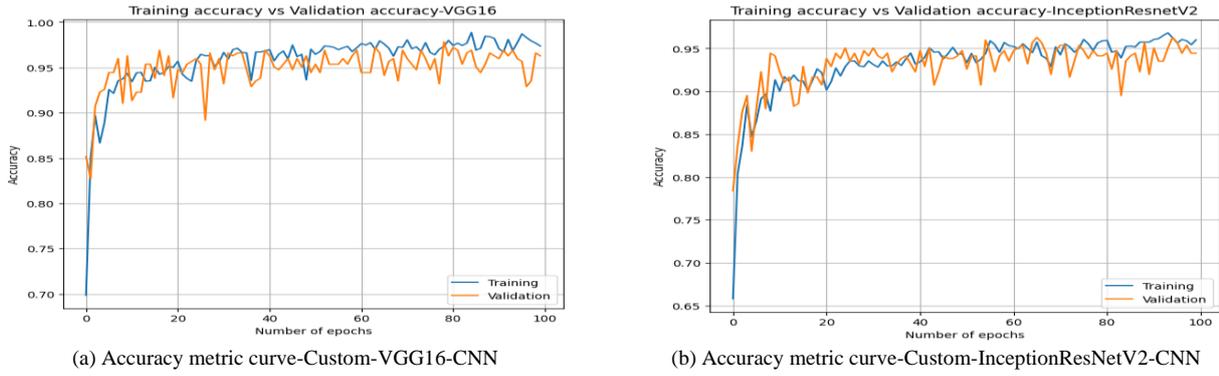


Fig. 10. (a) Loss metric curve and (b) Accuracy metric curve of the VGG16-CNN and Custom--InceptionResNetV2-CNN Architecture. Iterations =100.

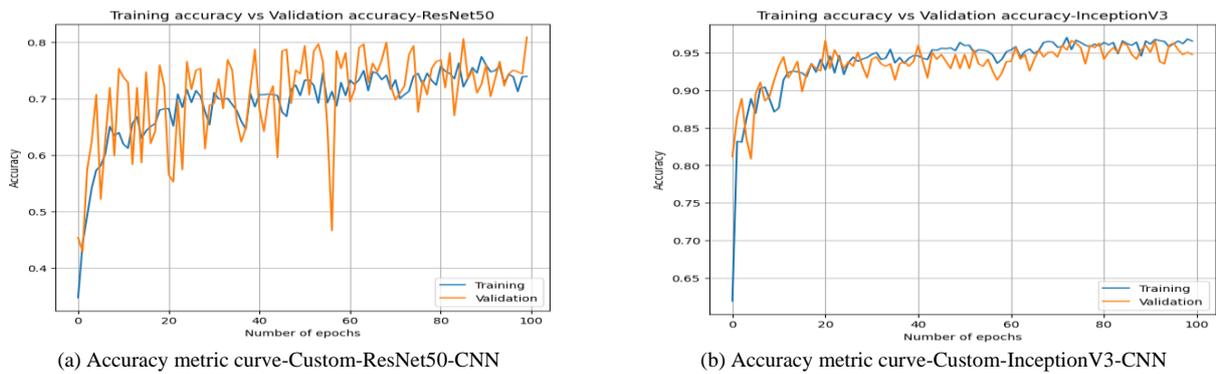


Fig. 11. (a) Loss metric curve and (b) Accuracy metric curve of the ResNet50-CNN and Custom—InceptionV3-CNN Architecture. Iterations =100.

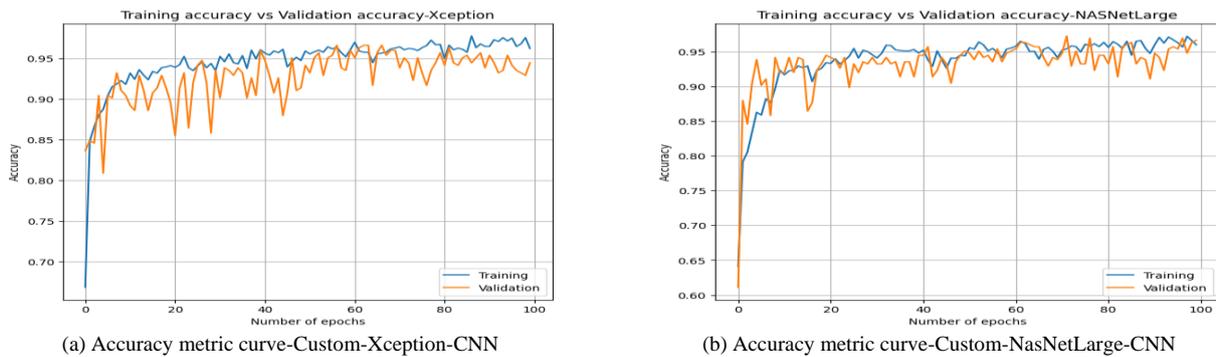


Fig. 12. (a) Loss metric curve and (b) Accuracy metric curve of the ResNet50-CNN and Custom—InceptionV3-CNN Architecture. Iterations =100.

A Confusion Matrix (CM) is an analytical method that illustrates how correctly a model has identified the various classification labels and is used during the classification task's performance assessment. A CM is a summary of the prediction findings for a classification problem; see Fig. 13. The CM for the DCPLD-CNN model is presented in Fig. 14-15, and CM of the eight customized pre-trained architectures are presented in Fig. 16-23, which demonstrates each class's classification accuracy distribution.

		Predicted Data	
		Positive (P)	Negative (N)
Actual Data	Positive (P)	True Positive (TP)	False Negative (FN)
	Negative (N)	False Positive (FP)	True Negative (TN)

Fig. 13. Structure of a Confusion Matrix (CM). Here, TP = prediction of true-positive occurrences, TN = prediction of true-negative occurrences, FP = prediction of false-positive occurrences, and FN = prediction of false-negative occurrences.

$$\text{ACCURACY} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{1}$$

$$\text{PRECISION} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{2}$$

$$\text{RECALL} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F-MEASURE} = \frac{2 * \text{PRECISION} * \text{RECALL}}{\text{PRECISION} + \text{RECALL}} \tag{4}$$

Using the abovementioned evaluation metrics presented through Equations (1)-(4), the models evaluation performance was calculated. The ratio of the correct segment of the test outcomes is called precision. The recall is the proportion of positive inspections aptly predicted to all supervision in the correct classes. To assess the model's overall performance, the F1 score was provided.

Diseased Cotton Leaves	0.99	0.01	0.00	0.00
Diseased Cotton Plants	0.00	1.00	0.00	0.00
Fresh Cotton Leaves	0.01	0.00	0.98	0.01
Fresh Cotton Plants	0.00	0.02	0.00	0.98
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 14. Class-wise Confusion Matrix of DCPLD-CNN for 32 batch size and 500 iterations of training

Diseased Cotton Leaves	0.99	0.01	0.00	0.00
Diseased Cotton Plants	0.00	1.00	0.00	0.00
Fresh Cotton Leaves	0.01	0.00	0.98	0.01
Fresh Cotton Plants	0.00	0.02	0.00	0.98
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 15. Class-wise Confusion Matrix of DCPLD-CNN for 64 batch size and 500 iterations of training

Diseased Cotton Laeves	1.00	0.00	0.00	0.00
Diseased Cotton Plants	0.00	1.00	0.00	0.00
Fresh Cotton Leaves	0.01	0.02	0.96	0.01
Fresh Cotton Plants	0.00	0.02	0.00	0.98
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 16. Class-wise Confusion Matrix of Custom-DenseNet121-CNN Architecture.

Diseased Cotton Leaves	0.94	0.00	0.02	0.04
Diseased Cotton Plants	0.01	0.95	0.01	0.03
Fresh Cotton Leaves	0.00	0.02	0.97	0.01
Fresh Cotton Plants	0.00	0.00	0.00	1.00
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 17. Class-wise Confusion Matrix of Custom-NasNetLarge-CNN Architecture.

Diseased Cotton Leaves	0.94	0.01	0.04	0.01
Diseased Cotton Plants	0.03	0.96	0.01	0.00
Fresh Cotton Leaves	0.02	0.01	0.97	0.00
Fresh Cotton Plants	0.01	0.06	0.01	0.92
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 18. Class-wise Confusion Matrix of Custom-Xception-CNN Architecture.

Diseased Cotton Leaves	0.90	0.02	0.06	0.02
Diseased Cotton Plants	0.05	0.95	0.00	0.00
Fresh Cotton Leaves	0.00	0.00	1.00	0.00
Fresh Cotton Plants	0.00	0.02	0.04	0.94
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 19. Class-wise Confusion Matrix of Custom-InceptionV3-CNN Architecture.

Diseased Cotton Leaves	0.75	0.14	0.07	0.04
Diseased Cotton Plants	0.09	0.79	0.05	0.07
Fresh Cotton Leaves	0.04	0.01	0.87	0.08
Fresh Cotton Plants	0.03	0.06	0.08	0.83
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 20. Class-wise Confusion Matrix of Custom-ResNet50-CNN Architecture

Diseased Cotton Leaves	0.93	0.03	0.03	0.01
Diseased Cotton Plants	0.03	0.96	0.00	0.01
Fresh Cotton Leaves	0.02	0.02	0.95	0.01
Fresh Cotton Plants	0.01	0.00	0.02	0.97
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 21. Class-wise Confusion Matrix of Custom-InceptionResNetV2-CNN Architecture

Diseased Cotton Leaves	1.00	0.00	0.00	0.00
Diseased Cotton Plants	0.00	1.00	0.00	0.00
Fresh Cotton Leaves	0.04	0.00	0.96	0.00
Fresh Cotton Plants	0.02	0.01	0.00	0.97
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 22. Class-wise Confusion Matrix of Custom-VGG16 -CNN Architecture

Diseased Cotton Leaves	0.97	0.01	0.01	0.01
Diseased Cotton Plants	0.05	0.87	0.06	0.02
Fresh Cotton Leaves	0.00	0.00	0.95	0.05
Fresh Cotton Plants	0.00	0.01	0.05	0.94
	Diseased Cotton Leaves	Diseased Cotton Plants	Fresh Cotton Leaves	Fresh Cotton Plants

Fig. 23. Class-wise Confusion Matrix of Custom-VGG19 -CNN Architecture

Table 6. Summary of the performed experiments performance in diseased cotton plant and leaf recognition in this study.

Proposed Model	Evaluation Metric				Training	
	Accuracy	Precision	Recall	F1-Score	Epochs	Batch Size
DCPLD-CNN	88.89%	0.8627	0.886	0.874	100	32
	98.77%	0.9824	0.980	0.981	500	32
	86.42%	0.8456	0.799	0.821	100	64
	94.94%	0.9557	0.955	0.955	500	64
Custom-DenseNet121	98.60%	0.9886	0.988	0.988	100	32
	98.46%	0.9858	0.985	0.985		64
Custom-NasNetLarge	96.60%	0.9689	0.932	0.950		32
	94.44%	0.9178	0.910	0.913		64
Custom-Xception	94.48%	0.9448	0.924	0.934		32
	94.75%	0.9451	0.942	0.935		64
Custom-InceptionV3	90.02%	0.8810	0.912	0.896		32
	94.75%	0.9466	0.951	0.948		64

Custom-ResNet50	80.86%	0.8028	0.816	0.809		32
	75.93%	0.7277	0.752	0.739		64
Custom-InceptionResNetV2	94.44%	0.9444	0.942	0.943		32
	95.37%	0.9521	0.953	0.952		64
Custom-VGG16	96.30%	0.9535	0.952	0.952		32
	98.15%	0.9803	0.986	0.983		64
Custom-VGG19	92.59%	0.9289	0.927	0.928		32
	93.21%	0.9388	0.934	0.967		64

## 5. Performance Comparison of the Study

This paper provides a meticulous study of implementing the CNN-based cotton leaf and plant disease recognition model. In the previous literature, there is a shortage of extensive study of utilizing CNN for this task. To the best of the author's knowledge no study was done previously regarding diseased cotton plant, leaves recognition using stack CNN and eight pre-trained architecture along with IDA. In Table 7, a comparative analysis is presented between this work and other existing studies.

Table 7. Comparison of this study with other existing study regarding diseased cotton plants leaves recognition (✓: Performed, x: Did Not Perform).

Ref.	Model	IDA	TL	Accuracy
[17]	SVM	x	x	Not mentioned
[18]	ANN	x	x	Relative Error = 0.051.
[37]	CNN	x	x	96%
This Study	DCPLD-CNN	✓	✓	98.77%
	Pre-Trained	✓	✓	98.60%

## 6. Discussion

In this paper, a novel Deep-CNN based architecture named DCPLD-CNN and eight pre-trained customized Deep CNN architecture was evaluated based on numerous experiments in diseased cotton leaf and plant recognition. IDA was implemented to expand the number of samples during training the DCPLD-CNN and eight customized pre-trained models. It is perceived that the DCPLD-CNN method accomplished better as the no. of iterations during training increased. A batch size of 32 performed better than a batch size of 64 in DCPLD-CNN model. The experiment's outcome indicates that, amongst the eight customized pre-trained models, DenseNet121, NasNetLarge, Xception, InceptionResNetV2, VGG16, VGG19, and InceptionV3 were extraordinary. However, the ResNet50 did not perform better as the Top 5 accuracies suggest in this recognition task. An accuracy chart is provided in Fig. 24. The experimented models' results show that CNN can be effectively used in cotton leaf and plant disease recognition tasks. The DCPLD-CNN model achieves an accuracy of 98.77%, which is a more significant and improved number compared to previous literature, which used conventional ML algorithms and various image preprocessing techniques. Fine-tuned Pre-trained architectures also show tremendous accuracy in this task, making them a significant choice for disease detection in the agricultural sector regarding plants and leaves.

There are numerous future research guidelines. Recognition of sternness of the disease can be classified. To do that, we need to expand the diseased crop dataset. There is a shortage of sufficient datasets in this field to attain a more robust and accurate outcome. More datasets with labeled cotton specific diseases and real environmental background are needed to deploy a DL based model for more vigorous detection disease types in cotton plant. Region-based segmentation of disease in cotton plants and leaves is another option for applying DL methodology. More extensive and cross-disciplinary exploration is essential to apprehend the dynamics influencing the recognition of plant diseases, such as the categories, volume of datasets, rate of learning, accurate depth of design architecture, lighting, state of the disease, etc.

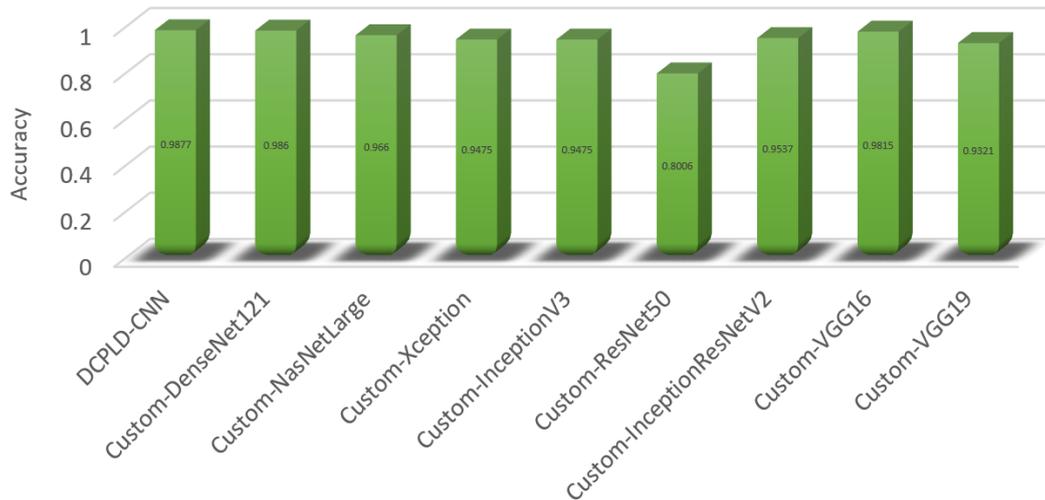


Fig. 24. Accuracy of the experimented models in this study.

## 7. Conclusion

Deep Learning's revolution has created a surge in many practical applications in image classification tasks regarding disease recognition in various plants. Going out of the traditional manual detection and diagnosis of diseased cotton plants is the way to move forward now, due to robust deep learning architectures classification and detection precision through automatic features extraction characteristics. This study shows that DL-based models, particularly CNNs, achieve greater accuracy in the cotton plant and leaves disease classification. Furthermore, experiments piloted in this study show that the advantage of using a pre-trained model, specifically if the no. of instances being used in training any model, is not very significant or low in volume. In this study, a CNN based novel DCPLD-CNN method is presented to recognize diseased cotton plants and leaves. DCPLD-CNN method accomplished an intricate accuracy of 98.77% with a 32 batch size and 94.94% with a batch size of 64. Eight pre-trained CNN models were also customized and experimented with within this study. Custom-DenseNet121 model attained the top accuracy of 98.60%, whereas all the other models returned great detection accuracy of over 90%. Only the Custom-ResNet50 model did not perform well compared to the other models based on transfer learning techniques. All the experimented methods could be applied to newer types of diseased plants for detection and classification even at an early stage with more real-world datasets and will significantly contribute to the agricultural sector of a country in a digital, fast and efficient way. In the future, the author aims to decrease all the models' processing and magnitude and turn it into a smartphone app that farmers can use to detect diseases flexibly without the need for external human supervision. The author also hopes to expand this model by including useful and practical recommendations and a consumer feedback module to effectively manage cotton plant and leaf diseases.

## Acknowledgements

The author would like to recognize the anonymous critics for their esteemed reviews. No Grant was provided for this research.

## References

- [1] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases," *Comput. Electron. Agric.*, vol. 72, no. 1, pp. 1–13, 2010.
- [2] A. Camargo and J. S. Smith, "An image-processing based algorithm to automatically identify plant disease visual symptoms," *Biosyst. Eng.*, vol. 102, no. 1, pp. 9–21, 2009.
- [3] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Comput. Electron. Agric.*, vol. 175, no. December 2019, p. 105527, 2020.
- [4] Y. M. Oo and N. C. Htun, "Plant Leaf Disease Detection and Classification using Image Processing," *Int. J. Res. Eng.*, vol. 5, no. 9, pp. 516–523, 2018.
- [5] A. Clément, T. Verfaillie, C. Lormel, and B. Jaloux, "A new colour vision system to quantify automatically foliar discoloration caused by insect pests feeding on leaf cells," *Biosyst. Eng.*, vol. 133, no. 0, pp. 128–140, 2015.
- [6] D. Story, M. Kacira, C. Kubota, A. Akoglu, and L. An, "Lettuce calcium deficiency detection with machine vision computed plant features in controlled environments," *Comput. Electron. Agric.*, vol. 74, no. 2, pp. 238–243, 2010.

- [7] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, "Image processing Based Detection of Fungal Diseases in Plants," *Procedia Comput. Sci.*, vol. 46, no. Icict 2014, pp. 1802–1808, 2015.
- [8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [9] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015.
- [10] S. Zhang, W. Huang, and C. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition," *Cogn. Syst. Res.*, vol. 53, pp. 31–41, 2019.
- [11] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, no. March, pp. 272–279, 2019.
- [12] H. Durmus, E. O. Gunes, and M. Kirci, "Disease detection on the leaves of the tomato plants by using deep learning," *2017 6th Int. Conf. Agro-Geoinformatics, Agro-Geoinformatics 2017*, 2017.
- [13] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access*, vol. 6, no. c, pp. 30370–30377, 2018.
- [14] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019.
- [15] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Deep learning for visual unDerstanDing: part 2 Generative Adversarial Networks," *IEEE Signal Process. Mag.*, no. January, pp. 53–65, 2018.
- [16] B. Cheng and E. T. Matson, "A feature-based machine learning agent for automatic rice and weed discrimination," *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.)*, vol. 9119, no. September 2014, pp. 517–527, 2015.
- [17] S. P. Patil and R. SZambre, "Classification of Cotton Leaf Spot Disease Using Support Vector Machine," *J. Eng. Res. Appl. www.ijera.com*, vol. 4, no. 5, pp. 92–97, 2014.
- [18] N. Shah and S. Jain, "Detection of Disease in Cotton Leaf using Artificial Neural Network," *Proc. - 2019 Amity Int. Conf. Artif. Intell. AICAI 2019*, pp. 473–476, 2019.
- [19] M. S. Al-Tarawneh, "An empirical investigation of olive leave spot disease using auto-cropping segmentation and fuzzy C-means classification," *World Appl. Sci. J.*, vol. 23, no. 9, pp. 1207–1211, 2013.
- [20] A. Adeel *et al.*, "Diagnosis and recognition of grape leaf diseases: An automated system based on a novel saliency approach and canonical correlation analysis based multiple features fusion," *Sustain. Comput. Informatics Syst.*, vol. 24, p. 100349, 2019.
- [21] S. Zhang, X. Wu, Z. You, and L. Zhang, "Leaf image based cucumber disease recognition using sparse representation classification," *Comput. Electron. Agric.*, vol. 134, pp. 135–141, 2017.
- [22] J. K. Patil, "Color Feature Extraction of Tomato Leaf Diseases," *Int. J. Eng. Trends Technol.*, vol. 2, no. 2, pp. 72–74, 2011.
- [23] H. Sabrol and K. Satish, "Tomato plant disease classification in digital images using classification tree," *Int. Conf. Commun. Signal Process. ICCSP 2016*, pp. 1242–1246, 2016.
- [24] M. P. Arakeri, M. Arun, and P. R. K., "Analysis of Late Blight Disease in Tomato Leaf Using Image Processing Techniques," *Int. J. Eng. Manuf.*, vol. 5, no. 4, pp. 12–22, 2015.
- [25] D. Ashourloo, H. Aghighi, A. A. Matkan, M. R. Mobasheri, and A. M. Rad, "An Investigation Into Machine Learning Regression Techniques for the Leaf Rust Disease Detection Using Hyperspectral Measurement," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 9, no. 9, pp. 4344–4351, 2016.
- [26] R. Kumar Sahoo, R. Panda, R. Chandra Barik, and S. Nath Panda, "Automatic Dead Zone Detection in 2-D Leaf Image Using Clustering and Segmentation Technique," *International Journal of Image, Graphics and Signal Processing(IJIGSP)*, vol. 10, no. 10, pp. 11–30, 2018.
- [27] T. Rumpf, A. K. Mahlein, U. Steiner, E. C. Oerke, H. W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance," *Comput. Electron. Agric.*, vol. 74, no. 1, pp. 91–99, 2010.
- [28] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi, "Cucumber leaf disease identification with global pooling dilated convolutional neural network," *Comput. Electron. Agric.*, vol. 162, no. December 2018, pp. 422–430, 2019.
- [29] S. Uğuz and N. Uysal, "Classification of olive leaf diseases using deep convolutional neural networks," *Neural Comput. Appl.*, vol. 5, 2020.
- [30] J. G. Arnal Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosyst. Eng.*, vol. 180, no. 2016, pp. 96–107, 2019.
- [31] J. Amara, B. Bouaziz, and A. Algergawy, "A deep learning-based approach for banana leaf diseases classification," *Lect. Notes Informatics (LNI), Proc. - Ser. Gesellschaft fur Inform.*, vol. 266, pp. 79–88, 2017.
- [32] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 299–315, 2017.
- [33] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, no. September, pp. 1–10, 2016.
- [34] Q. Wu, K. Zhang, and J. Meng, "Identification of Soybean Leaf Diseases via Deep Learning," *J. Inst. Eng. Ser. A*, vol. 100, no. 4, pp. 659–666, 2019.
- [35] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.
- [36] Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic Study of Automated Diagnosis of Viral Plant Diseases Using Convolutional Neural," *Int. Symp. Vis. Comput.*, no. II, pp. 638–645, 2015.
- [37] A. Jenifa, R. Ramalakshmi, and V. Ramachandran, "Cotton Leaf Disease Classification using Deep Convolution Neural Network for Sustainable Cotton Production," in *International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development, INCCES 2019*, 2019, pp. 2019–2021.
- [38] "Cotton Plant Disease Prediction," 2020. [Online]. Available: <https://indianaiproduction.com/>. [Accessed: 11-Jan-2021].
- [39] "Data augmentation Techniques." [Online]. Available: <https://iq.opengenus.org/data-augmentation/>. [Accessed: 14-Jan-2021].

- [40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
- [41] M. Längkvist, L. Karlsson, and A. Loutfi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2014.
- [42] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8697–8710, 2018.
- [43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016.
- [44] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017.
- [45] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," *Eur. Conf. Comput. Vis.*, vol. 9908 LNCS, pp. 630–645, 2016.
- [46] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017.
- [47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 9, no. 8, pp. 248–255, 2010.

## Authors' Profiles



**Md. Rayhan Ahmed** is currently serving as a Senior Lecturer in the Department of Computer Science and Engineering at Stamford University Bangladesh. He received his Bachelor of Science (BSc) degree from Ahsanullah University of Science and Technology (AUST) in 2014 and Master of Science degree (MSc) from United International University (UIU), Bangladesh in 2021. His research interest is Machine Learning, Artificial Intelligence, Deep Learning, Computer Vision, Data Science, Implementation of the Internet of Things (IoT) in real-world applications, and development of the android application.

**How to cite this paper:** Md. Rayhan Ahmed, " Leveraging Convolutional Neural Network and Transfer Learning for Cotton Plant and Leaf Disease Recognition", *International Journal of Image, Graphics and Signal Processing(IJIGSP)*, Vol.13, No.4, pp. 47-62, 2021.DOI: 10.5815/ijigsp.2021.04.04