Automatic Classification of Real-Time Diseased Cotton Leaves and Plants Using a Deep-Convolutional Neural Network

Chitranjan Kumar Rai (chitranjankrai@gmail.com)
Dr B R Ambedkar National Institute of Technology, Jalandhar

Research Article

Keywords: Convolution neural network, Image classification, Deep learning, Image processing, Plant leaf disease

Posted Date: March 14th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1440994/v1

Abstract

The automated detection and classification of plant diseases based on images of leaves is a significant milestone in agriculture. In this paper, the concept of deep learning was used to identify and predict cotton plant disease status using real-time images of leaves and plants. The models were trained using a database of 2293 images of cotton leaves and plants. The data included four distinct classes of leaves, plants disease combinations, and their respective categories. Python version 3.6.9 is used to implement the developed model. The model includes a deep learning package such as Keras, TensorFlow, and a Googlecolab cloud-based jupyter notebook as the development environment. For classifying leaves and plant diseases in cotton plants, our model attained an accuracy of 97.98%. The proposed technique outperformed the recent approaches indicated in earlier literature for relevant parameters. As a result, the technique is intended to reduce the time spent identifying cotton leaf disease in significant production regions and human error and time spent determining its severity.

Introduction

India is the world's largest producer of cotton, which is also regarded as 'The king of fibres' and its biggest consumer. It directly supports about 60 million farmers and employs approximately 40 million people in the agricultural and industrial sectors. India's cotton production is estimated at around 26% of the global total [1]. The country's average per kg production is still lower than the world average. However, due to the pest and disease proneness of the crop, its production has decreased. Some common categories of disease and pests affecting cotton productivity and quality are a bacterial blight, leaf miner, and spider mite [2].

Even though agriculture is India's backbone, no technological advances have been researched in developing automation in agriculture. Due to various diseases and parasites, there are significant challenges in terms of productivity and quality [3]. Plant disorder has been considered one of the most significant hazards to food security [4]. This issue can be solved using the most accurate and precise diagnostic tools. The use of visual inspection has contributed significantly to identifying plant diseases, although it takes very much time. However, due to the nature of the conditions, their detection requires extensive training and experience. The evolution of digital camera has enabled the simultaneous identification of various plant diseases using images [5]. Due to the rapid emergence and development of digital image acquisition technologies, the field of image-based diagnosis has gained widespread acceptance. It is possible to identify the diseases in plants by examining leaf image's shape and color features. However, due to the complexity of the acquired image, it requires a specific preprocessing step to extract its features [6]. Computer vision technology is also commonly utilized to obtain plant morphological features and classify and recognize plants diseases [7, 8]. Artificial neural networks (ANNs) are a specific tool for detecting disease in plants based on the successes of current neuroscience research [9, 10]. By modelling the human brain, it can make fundamental decisions. These techniques, however, have low detection capability due to the complex image preprocessing and feature extraction. In such cases, deep learning is a type of machine learning that autonomously enables a computer to learn a

particular feature avoiding complicated image preprocessing without human intervention. The Convolution neural network (CNN), an end-to-end deep learning approach, is widely used in image recognition among the various deep learning networks. It is a supervised deep learning system with high classification accuracy and is capable of learning and classifying objects. Since the early days of their development, deep CNNs have been used for image analysis. Due to their improved capabilities, they became widely used in the 2010s.

AlexNet was one of the CNNs that significantly impacted image classification [11]. In 2016, [12] proposed an image-based deep learning model to detect and classify diseases in plants. Since then, various deep learning models have been reported for disease diagnosis in different plants [13–17], but they failed miserably when these models were tested on the real-time dataset. The author [18] proposed a CNN model for the automatic identification of plants based on their leaves. Silva et al. [19] modified the CNN model to estimate the defoliation level of the soybean plant. The researchers proposed a faster and more accurate R-CNN architecture to identify leaf diseases in sugar beet [20] automatically. There are few studies that have focused on the application of deep learning algorithms to evaluate the severity of plant infections. The developed model is a VGG16 and ResNet50 architecture trained and tested on apple rot and coffee leaf images from the Plant Village dataset with a transfer learning approach that yields 90.4% and 95.24% accuracy[20–22]. The author in [23] proposed an optimal deep learning network model called VGG and AlexNetOWTBn for identifying and classifying healthy and diseased leaves on a large image dataset. The authors [24] proposed novel techniques for detecting plant leaf diseased based on a Deep CNN using different training epochs, batch sizes, and dropouts and obtained a classification accuracy of 96.4%. The authors [25] employed a deep CNN to identify four cucumber diseases based on their symptoms (i.e., downy mildew, anthracnose, powdery mildew, and target leaf spots). These researches indicated that CNNs had made significant contributions to identifying and detecting plant diseases in various situations.

Implementation of real-time cotton leaf and plant disease detection and classification remains problematic because of environmental factors such as the shadow of one leaf on the other, light, and background soil. Unfortunately, no viable CNN models for real-time diagnosis of cotton leaves and plant diseases exist, which would be extremely useful in cotton production. This study aimed to develop disease classification models for cotton leaves and plants using a deep learning approach known as a deep convolutional neural network (DCNN). Based on the datasets, the DCNN was trained to classify fresh and diseased cotton leaves and plants.

Materials And Methods

All experiments are carried out on the platform, with the following specifications: Intel (R) core i3-4005 CPU @ 1.70 GHz Processor, 8.00 GB RAM, 64-bit Window 10 Operating System, and Google Colab Pro with Python 3.6.9 as the coding environment. Fig. 1 depicts the flowchart of our proposed research. In this article, a CNN motivated by the AlexNet architecture [11] is proposed for the classification of fresh and diseased cotton leaves and plants.

Dataset Description

In the proposed work, datasets of 2293 real-time images of cotton leaves and cotton plants are used. These datasets were collected from Kaggle, the online free database platform. The images are categorized among four classes: diseased cotton leaf (DCL), diseased cotton plant (DCP), fresh cotton leaf (FCL), and fresh cotton plant (FCP). Based on the category, the image dataset is labelled to its appropriate classes. Fig. 2 shows the sample of the image dataset taken in the actual field condition. Table 1 shows the number of images belonging to each class.

Table 1 Dataset Description

Categories	Number of Images
DCL	346
DCP	922
FCL	511
FCP	514
Total	2293

Data Augmentation

Data augmentation refers to the creation of new data sets from existing datasets. This procedure is used to augment the available data and increase its size to provide a larger and more accurate dataset and compensate for the additional data collection costs. To improve the training process, we propose introducing random transformations such as rescaling, rotation by 40°, width and height shifting the image by 20% each, shear and zooming the image by 20% each, and horizontal flipping with fill mode as nearest. The preceding strategies enlarge the dataset, which aids in preventing overfitting during the training stage [26]. Fig. 3 shows the image sample after the data augmentation technique is applied.

The architecture of Convolution Neural Network

The architecture of the CNN model that we have proposed is shown in Fig. 4. The architecture consists of four convolution layers, four pooling layers, and fully connected layers. With a SoftMax activation function, the final layer serves as an output layer. To change the images into a 1D array, a flatten acting as a hidden layer is used, resulting in improved performance and more accessible data handling. Each

convolutional layer has a size of 3x3, and each max-pooling has a length of 2x2. The input image size and its feature map vary, as shown in Fig. 4.

Detail of proposed Convolution Neural Network model

The proposed model is a sequential model with a series of convolution and max-pooling layers that converts the input image into a feature set that can be processed further. In the first layer, the input image of size 224x224 is convolved with 32 filters of size 3x3 resulting in the dimension of 222x222x32. A max-pooling layer with a filter size of 2x2 makes up the second layer that will half the convoluted image size, i.e., 111x111x32. Similarly, the third layer performs a convolution operation with 64 filters of size 3x3, accompanied by a fourth max-pooling layer with a filter size of 2x2. As a result, the image size will be reduced to 54x54x64. The fifth layer again performs a convolution operation with 128 filters of size 3x3, followed by a sixth pooling layer that will reduce the convolved image's size to 26x26x128. A combination of convolution layers with 256 filters, each with size 3x3 and a max-pooling layer with a filter of size 2x2, is again used in the seventh and eighth-layer. Thus, the resulting image dimension will be reduced to 12x12x256. A dropout of 0.5 is used after the last max-pooling layer. As the image dimension is reduced, a flattening layer is added to flatten the previous layer's output. This layer will offer the feature set for each image as output. The model comprises two dense layers that serve as the artificial neural network's hidden layer. These dense layers are interconnected, with every input neuron linked to every other hidden neuron. These layers have 128 and 256 neurons with ReLU as the activation function. Two dropouts of 0.1 and 0.25 are used after the dense layers. The last layer of the model will be the output layer with Softmax as the activation function. A fully connected layer with four output neurons will predict the class labels and evaluate the proposed model's performance (Accuracy, Precision, Recall, and F1 Score).

Training and Testing

The database was randomly split into three datasets: training, validation, and testing. For training and testing the proposed CNN model, over 2,293 image samples of fresh and infected cotton leaves and plants were taken in the field's actual condition were used. Furthermore, the training dataset is divided into training and validation to determine model overfitting. Out of 2293 image samples, 2095 images were included for the training and validation process, and the remaining 198 images were preserved to test the model's performance in classifying new images. The complexity of these images is evident, with multiple facets contributing to it. Some of these include the appearance of numerous leaves and other parts of the plants and the irrelevant objects in the background, such as shoes. The adam optimization with a learning rate of 0.0001, batch size 32, and categorical cross-entropy as loss function is used to train the network. In addition, the model is trained using the three different train, validate, and test ratios, using four different max-pooling layers configuration and finally using three different numbers of epochs.

Visualization approaches are used to expose CNN feature maps in order to identify how CNNs learn features for discriminating across classes. This experiment can better appreciate the differences in feature maps produced from several diseased cotton leaves and plant images. Fig. 5 and

6 shows an example image from the dataset, together with the visualisation results of different convolution layers and different max-pooling layers.

Performance Parameter

The performance of our model was calculated using metrics such as accuracy, precision, recall, and F1 score. These metrics score between 0 and 1, with 1 being the best and 0 being the worst.

Accuracy: Accuracy is the proportion of accurately predicted images (TP+TN) to the total number of predictions (TP+TN+FP+FN). Accuracy can be evaluated using Equ. (1)

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision: Precision is expressed in the instances' correctly predicted (TP) to the total number of predictions as positive (TP+FP). Mathematically, it can be calculated using Equ. (2).

$$Precission = \frac{TP}{TP + FP} \quad (2)$$

Recall: Recall, also known as sensitivity, is the ratio of the instances correctly predicted (TP) to the total number of actual cases (TP+FN). Recall can be computed using Equ. (3).

$$Recall\ (Sensitivity) = \frac{TP}{TP + FN} \quad (3)$$

F1 Score: The F1-score compares precision and recall into a single measure. Mathematically it is the Harmonic mean of precision and recall and is computed using Equ. (4).

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (4)$$

Where TP: True Positives FP: False Positives TN: True Negatives FN: False Negatives.

Results

Keeping the learning rate fixed at 0.0001, we have split the database into three datasets: the training set, the validation set, and the testing set. The entire database is configured into three different training, validation, and testing ratios: 6:3:1, 7:2:1, and 8:1:1. In these configurations, the four pooling layers of the

CNN model was fixed to max-pooling, and the model was run for 100 epoch. The model attains the maximum training accuracy and testing accuracy of 96.36% and 94.95%, respectively, for dataset configuration of 8:1:1 [12], as depicted in Table 2. Figure 7 depicts the training accuracy and loss for all three formats of dataset configuration run for 100 epochs.

Table 2
Performance of the model after running for 100 epochs with three different dataset configurations

Dataset Configuration (%)			**Accuracy (%)**		**Loss (%)**	
Train	Validate	Test	Training	Testing	Training	Testing
60	30	10	89.89	72.49	19.20	56.60
70	20	10	95.96	86.90	11.85	55.07
80	**10**	**10**	**96.36**	**94.95**	**11.52**	**16.16**

In the second configuration, the dataset split ratio and the number of training epochs were constant to 8:1:1 and 100, respectively. The pooling layers configuration was shuffled accordingly, as shown in Table 3. Here the model shows overfitting when all the four pooling layers are taken as average pooling. With all the four pooling layers as max-pooling, the model offers maximum training and testing accuracy of 96.36% and 94.95%, the best among all the four combinations of pooling layers shown in Table 3. Figure 8 also shows the training accuracy and loss of all four combinations of pooling layers.

Table 3
Performance of the model after running for 100 epochs with four different pooling layer configurations

Pooling Layers Configuration	**Accuracy (%)**		**Loss (%)**	
	Training	**Testing**	**Training**	**Testing**
Avg.- Avg.-Max-Max	95.40	94.44	13.01	17.25
Max-Max- Avg.- Avg.	94.71	83.84	15.34	47.06
Avg.- Avg.-Avg.-Avg.	94.76	96.97	14.48	15.04
Max-Max-Max-Max	**96.36**	**94.95**	**11.52**	**16.16**

Four pooling layers and dataset split ratio were fixed in the third configuration to max-pooling and 8:1:1. Three different epochs were used for training, i.e., 100, 300, and 500. The proposed model shows the best training and testing accuracy of 99.73% and 97.98% for 500 epochs, as shown in Table 4. Figure 9 also shows the training accuracy and loss of all three combinations of epochs.

The proposed system's average testing accuracy and loss are 98% and 10.74%, respectively. The best combination of configuration obtained from the above-trained models is when the split ratio of the train, validate and test dataset is taken as 8:1:1 with Max-Max-Max-Max as the pooling layer and 500 as the

number of iterations. The training accuracy of the model is 99.73%, the testing accuracy is 97.98%, the training loss and testing loss is 0.92% and 8.38%, respectively, as shown in Table 4.

Table 4
Performance of the model after pooling layer fixed to max-pooling and dataset split ratio to 8:1:1.

Number of Epochs	Accuracy (%)		Loss (%)	
	Training	Testing	Training	Testing
100	96.36	94.95	11.52	16.16
300	99.37	96.97	2.31	7.68
500	**99.73**	**97.98**	**0.92**	**8.38**

The test results are prepared in the form of a confusion matrix used to evaluate the proposed model's performance. Figure 10 shows the confusion matrix of the final test results. It contains information about the actual class led in rows and information about the predicted class, represented in columns. All the correct predictions are on the diagonal, and incorrect predictions are off the diagonal. The darker the color, the more accurate the prediction is in the associated class. The model's classification accuracy can be visually evaluated based on these test results. According to the confusion matrix, the model is more prone to confusion in recognizing between diseased cotton plants and fresh cotton leaves when compared to other classes. Among the 83 images in the diseased cotton plant's testing set, two images were wrongly identified as fresh cotton leaf and fresh cotton plant. In addition, of the 43 images in the testing set of fresh cotton leaf, two images were incorrectly identified as diseased cotton leaf and fresh cotton plant. This misclassification may be due to the appearance of numerous leaves and other parts of the plants in the image. However, other classes are well distinguished.

Table 5 summarises the model's classification matrices for class names DCL, DCP, FCL, and FCP for accuracy, recall, precision, and F1-score matrices, which indicated the quality of predictions to investigate performance for each class.

Table 5
Classification report on the test image dataset

Class Name	Accuracy	Precision	Recall	F1 Score	Test images
DCL	1	0.97	1	0.98	29
DCP	0.98	1	0.98	0.99	83
FCL	0.95	0.98	0.95	0.96	43
FCP	1	0.96	1	0.98	43
Overall	0.98	0.98	0.98	0.98	198

The results of the proposed model are compared to those of other studies is represented in Table 6. The majority of the research reported in Table 6 employed datasets collected in a laboratory condition and considered only the plant's leaf as the input image. In our proposed model for disease classification, we have used the real-time dataset of cotton leaves and plants. We have considered both the leaves and the plants in the image dataset. The performance of our model is higher than the other model tabulated in Table 6.

Table 6
Comparison of proposed techniques with different plants disease detection methods

Author	Objective	Year	Image Count	Techniques Used	Learning rate	Number of iteration	Accuracy
Sukhvir et al.	Soybean leaf disease detection	2018	4775	K-mean Cluster	-	-	85.65
Ozguven et al.	Sugar beet leaf disease detection	2019	155	Updated FRCNN	0.001	150,000	95.48
Uday Pratap Singh et al.	Classification of Mango Leaves disease	2019	1070	MCNN	0.01	100	97.13
M Azath et al.	Cotton leaf disease detection	2021	2400	CNN	-	100	96.4
Proposed Technique	**Classification of Cotton leaf and plant**	**2021**	**2293**	**DCNN**	**0.0001**	**500**	**97.98**

Conclusions

The evolution of the agricultural sector has been widely influenced by the emergence of intelligent farming. This concept involves the use of various electronic platforms and methods to collect and analyze agriculture produce data. Machine learning and image processing have shown promising potential in detecting and classifying diseases in plants and leaves. Despite advancements in computer algorithms and techniques, there are still areas of research that need further development. The DCNN architecture was used to automatically classify diseases in cotton leaves and plants by modifying the parameters of the AlexNet architecture. On a test dataset of around 200 images, the proposed model achieves an accuracy of 97.98%. This study also shows that this technique can be utilized with minimal data to achieve its intended results. The current model outperforms the models given in prior studies. Based on current data, it was determined that the proposed DCNN model might be utilised to classify plant diseases correctly. In the future, we would like to improve disease classification accuracy and detect diseases in diverse crops using different deep learning algorithms. Future studies will also focus on identifying plant diseases in plant components other than leaves, such as flowers, fruits, and stems. This

model might potentially be used to diagnose plant leaf problems. Furthermore, we intend to analyse the training process more in-depth without labelled images.

Declarations

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Chockalingam SMM (2015) A profile of Indian cotton: at a glance. 1–6
2. Parikh A, Raval MS, Parmar C, Chaudhary S (2016) Disease detection and severity estimation in cotton plant from unconstrained images. Proceedings – 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016 594–601
3. Kumar R, Tiwari RK, Jeevalatha A, et al (2021) Potato apical leaf curl disease: current status and perspectives on a disease caused by tomato leaf curl New Delhi virus. Journal of Plant Diseases and Protection 128:897–911
4. Thangaraj R, Anandamurugan S, Kaliappan VK (2021) Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. Journal of Plant Diseases and Protection 128:73–86
5. Xiao BX, Wang CY, Guo XY, Wu S (2014) Image acquisition system for agricultural context-aware computing. International Journal of Agricultural and Biological Engineering 7:75–80
6. Vishnoi VK, Kumar K, Kumar B (2021) A comprehensive study of feature extraction techniques for plant leaf disease detection. Springer US
7. Sengar N, Dutta MK, Travieso CM (2018) Computer vision based technique for identification and quantification of powdery mildew disease in cherry leaves. Computing 100:1189–1201
8. Gupta H, Pahuja R (2019) Estimating Morphological Features of Plant Growth Using Machine Vision. 10:
9. Şekeroğlu B, Inan Y (2016) Leaves Recognition System Using a Neural Network. Procedia Computer Science 102:578–582
10. Hati S, G S (2013) Plant Recognition from Leaf Image through Artificial Neural Network. International Journal of Computer Applications 62:15–18
11. Krizhevsky A, Hinton GE ImageNet Classification with Deep Convolutional Neural Networks. 1–9
12. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Frontiers in Plant Science 7:
13. Jiang P, Chen Y, Liu BIN (2019) Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. IEEE Access 7:59069–59080

14. Zhou G, Zhang W, Chen A, et al (2019) Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion. IEEE Access 7:143190–143206
15. Sun J, Yang Y, He X, Wu X (2020) Northern Maize Leaf Blight Detection under Complex Field Environment Based on Deep Learning. IEEE Access 8:33679–33688
16. Azath M, Zekiwos M, Bruck A (2021) Deep Learning-Based Image Processing for Cotton Leaf Disease and Pest Diagnosis. Journal of Electrical and Computer Engineering 2021:
17. Zhang X, Qiao Y, Meng F, et al (2018) Identification of maize leaf diseases using improved deep convolutional neural networks. IEEE Access 6:30370–30377
18. Lee SH, Chan CS, Wilkin P, Remagnino P (2015) Deep-plant: Plant identification with convolutional neural networks. Proceedings - International Conference on Image Processing, ICIP 2015-Decem:452–456
19. da Silva LA, Bressan PO, Gonçalves DN, et al (2019) Estimating soybean leaf defoliation using convolutional neural networks and synthetic images. Computers and Electronics in Agriculture 156:360–368
20. Esgario JGM, Krohling RA, Ventura JA (2020) Deep learning for classification and severity estimation of coffee leaf biotic stress. Computers and Electronics in Agriculture 169:
21. Wang G, Sun Y, Wang J (2017) Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning. Computational Intelligence and Neuroscience 2017:
22. Vallabhajosyula S, Sistla V, Kolli VKK (2021) Transfer learning-based deep ensemble neural network for plant leaf disease detection. Journal of Plant Diseases and Protection. https://doi.org/10.1007/s41348-021-00465-8
23. Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture 145:311–318
24. Geetharamani G, J. AP (2019) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Computers and Electrical Engineering 76:323–338
25. Ma J, Du K, Zheng F, et al (2018) A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Computers and Electronics in Agriculture 154:18–24
26. Hawkins DM (2004) The Problem of Overfitting. Journal of Chemical Information and Computer Sciences 44:1–12

Figures

Figure 1

Flow chart of real-time disease classification of cotton leaves and plants

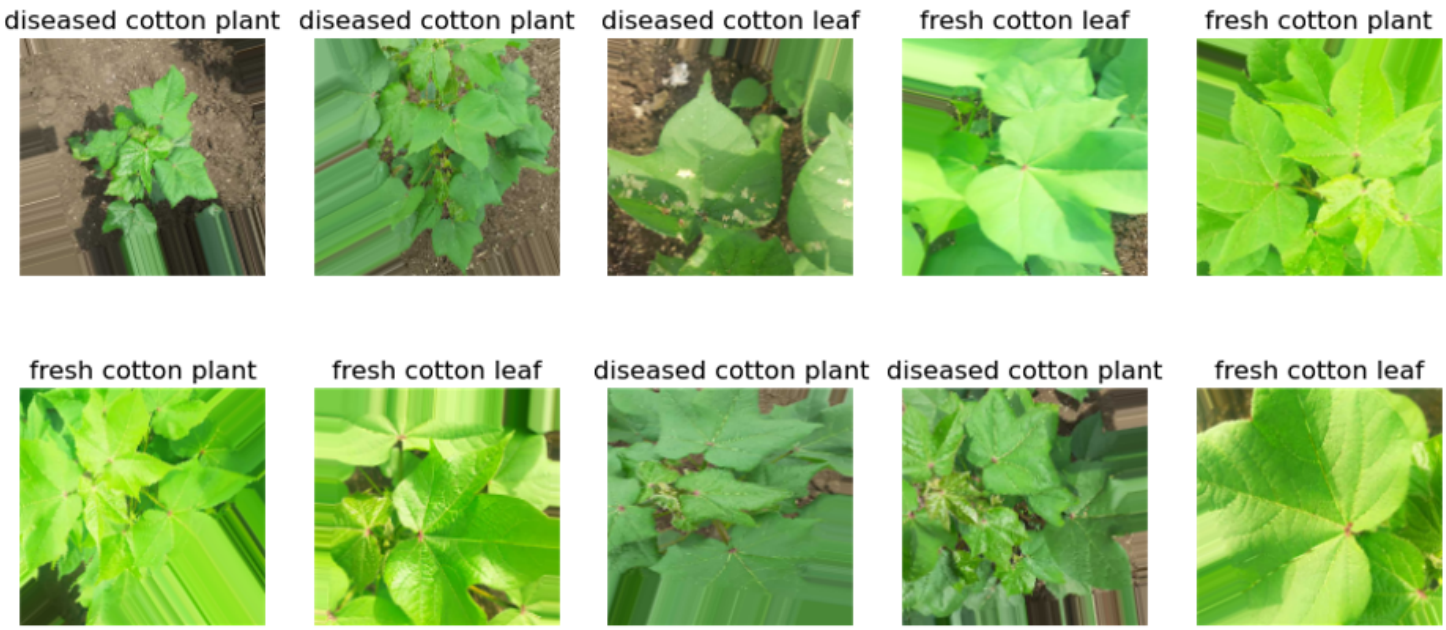

Figure 2

Sample of Images

Figure 3

Sample of cotton leaf image after the data augmentation process

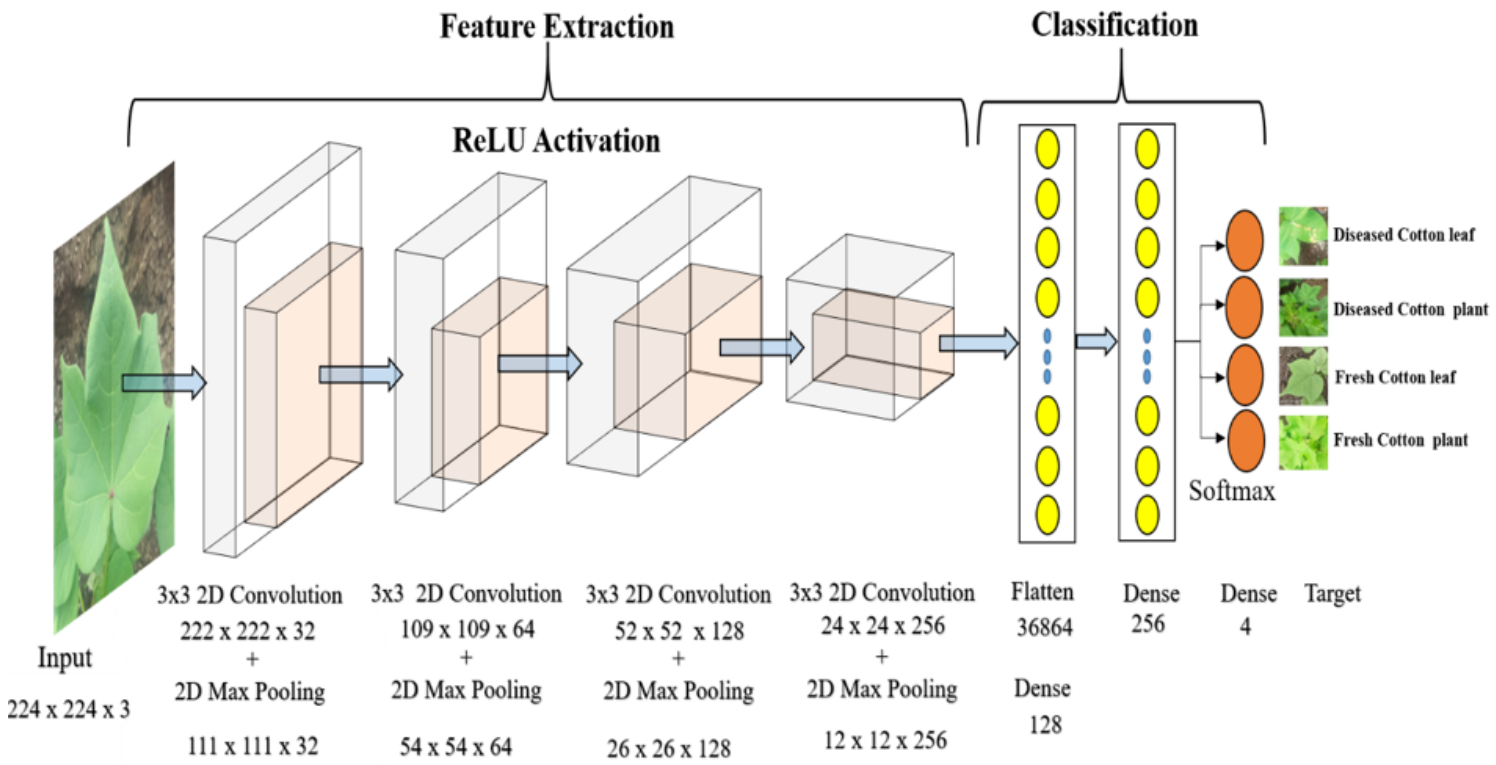

Figure 4

Proposed CNN architecture for training

Figure 5

Visual representation of different convolution layers

Figure 6

Visual representation of different max-pooling layers

Figure 7

Training accuracy and training loss of the model for three different dataset configurations.

Figure 8

The model's training accuracy and loss for four different pooling layer configurations.

Figure 9

The model's training accuracy and loss for three different epochs (a) 100 epochs (b) 300 epochs (c) 500 epochs

Figure 10

Confusion Matrix based on the test dataset using 500 epochs