Deep Learning Model for Disease Identification of Cotton Plants

Rajani Zambare
M.Tech. Student, Department of Technology, Shivaji University, Kolhapur, Maharashtra, India
rjzambare@gmail.com

Rashmi Deshmukh
Asst. Professor, Department of Technology, Shivaji University, Kolhapur, Maharashtra, India
rvm_tech@unishivaji.ac.in

Chetan Awati*
Asst. Professor, Department of Technology, Shivaji University, Kolhapur, Maharashtra, India
cja_tech@unishivaji.ac.in

Suresh Shirgave
Associate Professor, DKTE's, Textile and Engineering Institute, Ichalkaranji, Maharashtra, India
skshirgave@gmail.com

Sandeep Thorat
Asst. Professor, Government College of Engineering and Research, Avasari, Pune, Maharashtra, India
sathorat2003@gmail.com

Sheetal Zalte
Asst. Professor, Computer Science Department, Shivaji University, Kolhapur, Maharashtra, India
sheetal.zaltegaikwad@gmail.com

* Corresponding Author

Abstract: Disease management and disease prediction from plant images can be made accurately and efficiently with the deep learning technique. Early identification of plant diseases helps farmers improve plant production and economic growth. Hence, a system that automatically detects diseases is needed to revolutionize agricultural monitoring and allow plant leaves to be healed immediately after disease identification. An experimental study shows that the proposed CNN model technique for cotton plant disease detection is efficient. The proposed system takes a cotton plant image as an input and preprocesses it to get a digitized colour image. The leaf image is segmented, and relevant features are extracted from it. The other leaf image is classified with the CNN model. Experimentation shows that the proposed CNN model achieves higher accuracy using more than three layers and three hundred epochs for training. The model is optimized by adding a dense layer and flattening. The model accuracy obtained for classification is 99.38%.

Keywords: Cotton plant, CNN model, Machine Learning, Image classification, Prediction.

1. Introduction

Currently, with the speed of economic development of our country, the economic impact of agriculture is increasing. However, agriculture is a large industry and plays a significant role in our country. Therefore, automated disease detection systems are needed as they could revolutionize the monitoring of large crop areas and help leaves recover as soon as the disease is detected. Furthermore, many diseases affect cotton and many other crops that affect many farms. Therefore, this process reduces the time it takes to train a model for automatic cotton leaves or plant disease detection, and it also reduces the time it takes to identify the disease in plants using the trained model. Moreover, it shows that CNN is a better option for image classification as it provides higher accuracy.

The proposed system uses the CNN model as it is used in profound learning studies for quick and easy training and learning of data sets. The CNN model is easy to use and operates on low parameters. A complex neural network (CNN) processes input datasets like images, shapes, gradients, image patterns, illusions, illustrations, etc., in structured data arrays. CNN can efficiently work with any design pattern like lines, circles, gradients, or even eyes and facial expressions.

These features of convolutional neural networks make them more robust and easier to use in computer vision and image classification.

2. Literature Review

This paper discusses the automatic plant leaf disease classification [1] based on the OPNN classifier. In this investigation, a professional programmed diagnosis of maize plants was developed to avoid the problem of understanding the complex structure of OPNN. For building such a system, researchers use PNN classification and ANN. The proposed method Achieved 95.5% accuracy and is highly related to the existing method.

Convolution neural networks (CNNs) provide a quick and accurate diagnosis in image recognition. In this paper [2], the authors discuss using some techniques to detect particular diseases of the cassava crop. For training and validation model uses the dataset with 21397 images of the cassava plant. After training and testing the model, the evaluation shows that the projected process can attain high correctness stages. This experiment proves that the CNN model technique is feasible for plant disease detection and presents a suitable alternative to AI solutions for small farmers.

This paper [3] investigates the utilization of DL models. This paper studies the models like Alexnet and VGinet pre-trained on object groupings to seek texture cataloguing difficulties such as plant diseases. Investigation correlated to natural farming is significant because of its likely financial effect on rural efficiency and quality. In this context, the authors propose a method of feature extraction based on deep learning to identify plant species and classify plant leaf diseases. Leaf disease is characterized by image processing techniques where feature extraction plays a vital role because the use of appropriate structures guides classification accuracy. Furthermore, many machine learning techniques exist to detect and identify plant diseases, with in-depth learning progressing with upgraded performance and correctness. Therefore, this paper [4] introduces a prototype for detecting and recognizing crop leaf diseases. The model works on the CNN base Alexnet model. Also, it clearly explains the comparison of the proposed model with other CNN models. The proposed model offers 96.76% accuracy, higher than VGG-16 and Lenet-5.

This work [5] proposes a solution for paddy plants related diseases using CNN. CNN is a likely choice because it can train a model with a minimal training set, and also, in a deep learning algorithm, CNN Model Provides higher accuracy with faster convergence. The model uses ten convolutional layers for classification, and it takes 20 epochs for training. The learning rate of this model is 0.0001The Proposed CNN model Obtained 94.12% accuracy and confusion matrix. It examines that discovering plant or leaf diseases is a favourite subject of all deep learning and AI researchers. In agriculture, detecting plant or leaf diseases is an essential issue of investigation as they can show benefits by observing broad areas of production and thus identify disease manifestations.

This paper [6] introduces the DL model through CNN to train crop disease classifiers through image classification. Artificial intelligence drones are used as crop monitoring systems equipped with and implemented cameras for monitoring purposes. For training, the CNN model using the dataset contains 16,257 colour images. Furthermore, the dataset for training the CNN model has been categorized into ten categories of unhealthy crop leaves.

This paper [7] deals with issues in the field of palm disease and their time-series seriousness. Dates are an economically important crop for Pakistan as they are rich in protein. Therefore, image processing techniques try to recognize a specific and dangerous disease like a sudden decline syndrome. This paper presents a mechanism for four different stages of the disease identifying the disease at given stages of infected dates. Deep learning techniques diagnose diseases based on surface, and colour removal approaches. The system recognized the disease with 99% accuracy at stage 4 and 95.6 % at stage 3, and it gave 91% accuracy at stages 1 and 2, respectively. Subsequently, the convolutional neural network statistically analyses the results for further processing. This application helps macro and micro farmers and stakeholders across the country and worldwide.

The research [9] presents the Rice disease mechanism to get primary and a precise diagnosis of the disease. Image processing methods based on Deep Learning (DL) are the best solution for accurately and precisely identifying various diseases in rice plants. Twelve different rice disease images are used to train the CNN model for this research. A mean shift algorithm is used for preprocessing the input image. Lightweight CNN architecture is proposed to identify various diseases of rice plants, along with various sophisticated CNN architectures. Experimental results show that the proposed system detects rice disease with 95.4 % accuracy.

The system mainly focuses on microbial diseases, which threaten food security, and due to limited infrastructure and knowledge, it is difficult to identify. However, with the help of AI and automatic detection, it is easy to identify diseases. This paper [10] intends to distinguish the graph and mango leaf diseases and classify them into different disease categories. The Convolutional neural network is used to train the model to identify whether the diseases are present or not. For feature extraction and classification, the researchers use a design called AlesNet. The system is developed with MATLAB, which uses a dataset of 8,438 images of graph and mango leaves collected from the Plant Village dataset. As a result, the proposed system gives better accuracy.

This paper [11] examines accessible tactics to solve the problem debated with available datasets. This paper presents a study of various suggested methods, such as image editing based on classification, which are essential stages of disease identification. The paper's authors clearly describe the meaning of plant disease from the point of AI and DL. This paper shows that researchers used various techniques to detect disease promptly and reduce costs.

In traditional techniques, human experts in agriculture have been hired to find discrepancies between tomato plants due to pests, diseases, weather, and lack of nutrition. Then, to create more predictive accuracy, deep learning-based classification is introduced. This paper [13] provides an overview of recent tomato leaf disease detection work using image processing, machine learning, and deep learning methods. Furthermore, discuss general public and private datasets, methods used, and deep learning frameworks for diagnosing tomato leaf diseases.
Artificial intelligence-based deep learning methods are essential in detecting disease through images of plant leaves. However, finding diseases with small datasets using DL approaches is challenging. Transfer learning is one of the furthermost prevalent DL algorithms for successfully diagnosing plant diseases with minimal visual input. The Transfer Learning-Based Deep Convulsion Neural Network Model is proposed in this paper [14] for diagnosing tomato leaf diseases. This system result shows that the Adam Optimizer performs more accurately than SGD and RMSprop optimizers.

Recognition and identification of crop disease in the early stage is a significant concern of all farmers worldwide. At present-day, crop diseases are self-identified by farmers, who are inclined to make mistakes due to time-consuming, subjective, and human involvement. Early detection and detection of the disease will help farmers reduce the use of pesticides, reduce environmental impact and maximize profits by minimizing losses. DL algorithms are beneficial for accurately identifying crop diseases at an early stage. Research [8] aims to create a system to recognize and categorize grape diseases based on RGB leaf images.

Deep learning methodology models self-learn image features from datasets and obtains high image classification accuracy. This model provides higher accuracy than traditional methodologies. Furthermore, these model models obtain a higher accuracy rate and perform faster than traditional models. So-called object-scale adaptive convolutional neural network (OSA-CNN) is introduced in this article [15], which combines OBIA with CNN, and is proposed for HSR image classification.

3. Methodology

The following section gives the details of the proposed system.

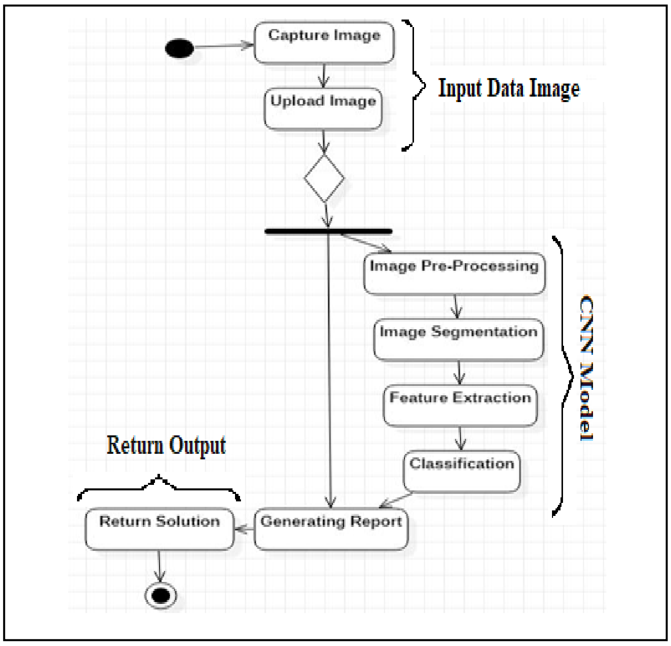

Figure 1. Proposed System Architecture

Figure 1 contains three parts, inputting data image, use of CNN model, and generation of the report.

Input Data Image
Input Data Image contains two parts 1. Capture image, and 2. Upload Image.
Capture Image: The end-user can capture a picture of a leaf or plant with the help of a device that contains a camera. The user can capture images in any image file format like jpeg, jpg, png, etc. For Training CNN Model, the researcher must take a dataset of healthy and disease leaves and plants to train the CNN model.
Upload Image: After capturing the image, the user can upload the image to a web page to get the result through the web application. For Training Purposes, the researcher has to give a training and testing dataset path for image preprocessing and image segmentation.

3.1 CNN Model

Image Preprocessing: Data preprocessing is a crucial step, and most profound learning researchers spend time in data preprocessing before creating a model. Data preprocessing includes external search, missing value treatment, and removal of unwanted or noisy data. In deep learning, the preprocessing image term represents the lowest level of abstraction for the operation of images. These operations do not improve image information content; they decrease it if entropy is regarded as a measure of information content. Instead, Pre-processing aims to modify image data to reduce unwanted noise or enhance specific image characteristics crucial for further processing.
Segmentation of Image: Image segmentation is a deep learning method for splitting a digital image into multiple segments, also referred to as "image objects" or "image regions." (Sets of

pixels). In more detail, image segmentation identifies each pixel in an image based on its label since pixels with identical labels share specific characteristics.

Feature Extraction: To perceive an object, the extraction of features procedure expects a fundamental function. Various attributes can be used to analyze a picture, including shade, surface (how the shade is distributed in the picture, the brilliance, and its hardness), morphology, edges, etc. Unfortunately, the information is too large to handle by calculation. Furthermore, it is expected to be depressingly dreary so that the data will be transformed into a limited number of capabilities of representations. The capacity set is constructed by converting the information data into highlights.

Several characteristics distinguish these large data sets from small ones: they are large and have plenty of variables. Therefore, computing resources are needed to process these variables. However, by selecting, combining, and eliminating variables from a large data set, features relevant to the problem can be extracted, thus reducing the amount of data significantly. The features of these datasets are easy to process, but they still contain accurate and original information about the data.

Classification: The procedure first computes the square of the data points provided. In this case, the classification algorithms used on the training data look for similar patterns (identical number sequences, words, or emotions, in subsequent data sets.

Return Output

After Training and building a CNN model, it can easily detect images and show the class of the given image of a leaf or plant. According to the classification of the data image, the report is generated, and the result will show that the data belongs to which category or class. This report is used to classify the result and generate output for the end-user. The end-user gets the solution on the web page as per the class.

Algorithm: Pseudo Code for the Convolutional Neural Network

1. Import libraries
2. Load the dataset into train_data_pata and validation_data_path from the storage drive
3. Predefine the methods/procedures to show the image preprocessing and segmentation results.
4. Use Preprocessing attributes for image preprocessing.
5. Validation dataset for rescaling
6. Creating CNN model
7. Compile CNN model with Adam optimizer
8. Train the CNN model.
9. Summarize the result of the CNN model and check the accuracy.

Figure 2. Pseudocode for CNN

Figure 2 shows different steps involved in the prediction of cotton plant disease. In the first step, required libraries like Keras and matplotlib. The image is processed with various attributes like rescale, rotation angle, width, height, etc. Convolutional layers with max-pooling of pool_size = (2, 2), input_shape [150, 150, 3], and, kernel_size 3 in each layer is used. The model uses a dense layer with four units and a softmax activation function to get the output.

Implementation of Algorithm

The proposed system uses real-time datasets that include images of diseased and healthy cotton plants or leaves. Some images are used for training, and the remaining are used for testing. The methodology uses the following steps which are involved in detecting cotton disease using a convolutional neural network (CNN):

Image Preprocessing
The image preprocessing is done and sent as input to the CNN model. For Pre-processing, the training dataset required Image Data Generator. The Image Data Generator creates new variants from old or input images at each epoch. The Image Data Generator used the following attributes to preprocess the training dataset:

Random Rotations: It randomly rotates the image between the angles 00 to 3600. This attribute takes the integer value.

Fill Mode: When the image rotates, some pixels will travel outdoor of the image and leave the vacant zone that requires to be occupied. This vacant zone will fill with the help of the fill_mode argument. For example, fill_mode = nearest.

Random Shift: Most of the time, the object is not placed in the center of the image, and to overawed this issue, there is a need to move the pixels of the image either horizontally or vertically. Adding some constant value in all pixels centralizes the object in the image. Two arguments are used to centralize the object as height_shift_range – it is used for vertical shift and width_shift_range – it is used for the horizontal shift.

Random Flips: Flip the image is always used to demonstrate the effect of image argumentation. Two arguments as, horizontal_flip and vertical_flip, are used. The value of these arguments is either true or false.

Random Brightness: It is used to change the brightness of the image. However, sometimes images are unclear due to the light or brightness of the image. Therefore, it becomes imperative to train the dataset of the CNN model in different light conditions with the help of random brightness. Random brightness used brightness_range argument with float value. If the value is less than 1.0, it refers to the darkness, and if the value is more significant than 1.0, it refers to the image's brightness.

Random Zoom: It uses to zoom in and out the image, taking a float value. zoom_range argument is used for achieving the purpose of random zoom.

Rescale: The image is formed by a pixel having a value of 0 to 255. Here, 0 refers to black, and one refers to the white colour. However, the colour image contains three red, green, and blue maps, and 255 is the maximum value for the image, so it rescales by 1. /255 to transfer every pixel value from 0, 255 - > [0, 1]. Rescaling images contributes to reducing the loss of images during preprocessing.

Shear Range: This attribute is used for the computer so that the computer or AI can see how humans see the object through their eyes from a different angle. The shear_range argument is used with a float value to assign sheer range in preprocessing. With the above attributes, we can perform preprocessing on the image and get the result in figure 3.

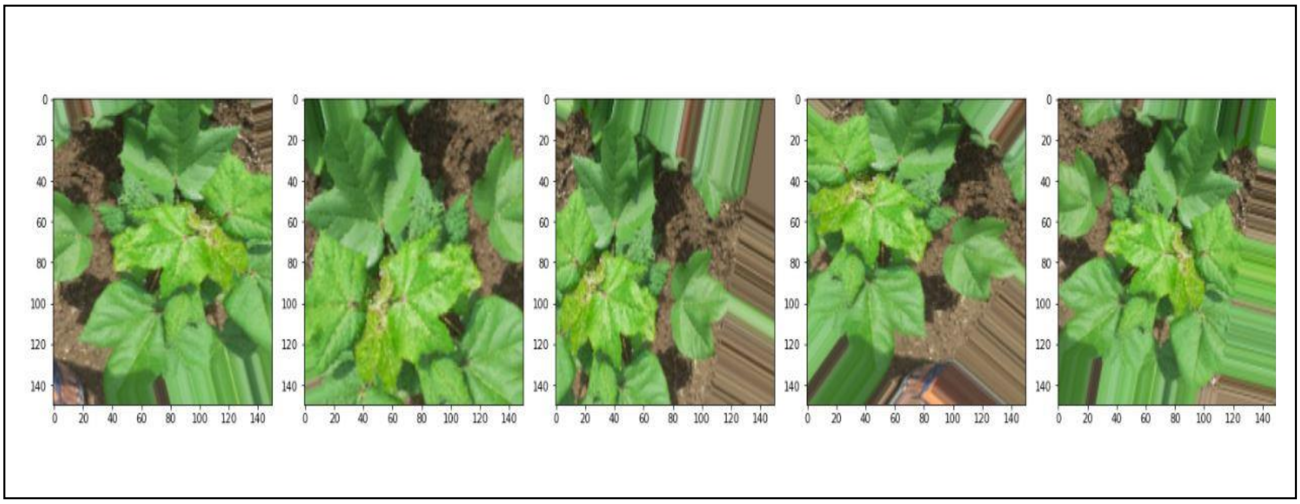

Figure 3. Use of image preprocessing step

Image Segmentation:
It divides images into separate segments based on their characteristics and qualities. Image segmentation's primary purpose is to make image analysis easier. In image segmentation, an image is divided into sections with the same properties. As shown in the algorithm, we can use the feature standardization arguments of ImageDataGenerator to perform Image segmentation on the input dataset.

Feature Standardization: Standardizing pixel values throughout the whole dataset is also possible. This is known as feature standardization, similar to the standardization commonly done for each column in a tabular dataset. The featurewise_center and featurewise_std_normalization parameters on the ImageDataGenerator class can be used to accomplish feature standardization. By default, they are set to True, and creating an instance of ImageDataGenerator with no arguments has the same effect. The image segmentation results are shown in figure 4.

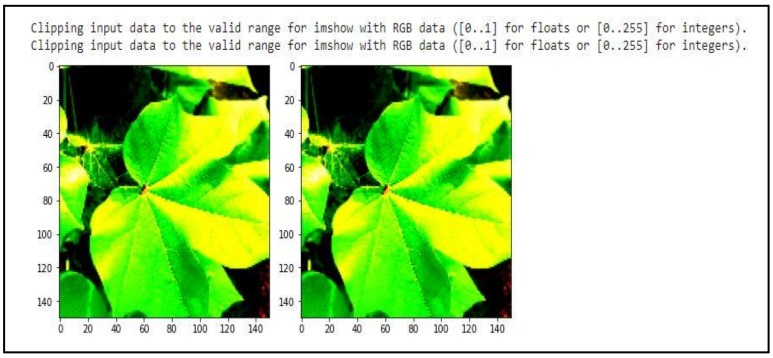

Figure 4. Use of Image Segmentation step

Feature Extraction: Feature extraction in the CNN model requires Convolution Conv2D and Max pooling. Extracting the features from the input images uses kernel size and filters. The algorithm shows that the convolutional layer uses a different range of filters with batch size. For example, the batch size in the convolutional layer is started from 1 to 256 and for specifying the colour images, apply batch size is 3. Max pool layer extracts more relevant features from the previous step. In this layer, the pool is defined to reduce the size of the image and then sent to the next convolutional layer. After every convolutional layer, the number of parameters is changed because of kernel size, batch size (which represents the colour image channel), and filter size. In the CNN model, the parameters consist of two parts: 1. Weight and 2. Bias

$$p = w + b \qquad \text{..Parameter formula}$$

$$w \ = k^2 * f * c \quad \text{..Weight formula}$$

$$b = f \qquad \text{.. Bias}$$

Here, p: parameter, w: weight, k: kernel size, f: filter size, c: channel size / batch size, b: filter size. Consider convolution layer one Kernel size (k) is 3, filter size (f) is 32, and for colour image batch size (c) is 3. Hence the values of w and b are Weigh formula is w =k^2*f*c = 32 * 32 * 3= 864. Bias b=f = 32. Hence $p = w + b$ = 864 + 32 = 896 for convolution layer 1.
Similarly, to get the total number of parameters used in the CNN model need to add parameters from each layer used in the CNN model.
Tp = Σ P_c; Here, Tp is the total number of parameters. Pc is the number of parameters from each convolutional layer.
The CNN model summary in figure 5 shows the number of parameters of each layer with the total number of parameters used in the CNN model.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      896

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 conv2d_3 (Conv2D)           (None, 15, 15, 256)       295168

 max_pooling2d_3 (MaxPooling  (None, 7, 7, 256)        0
 2D)

 dropout (Dropout)           (None, 7, 7, 256)         0

 flatten (Flatten)           (None, 12544)             0

 dense (Dense)               (None, 128)               1605760

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 256)               33024

 dropout_2 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 4)                 1028

=================================================================
Total params: 2,028,228
Trainable params: 2,028,228
Non-trainable params: 0
```

Figure 5. CNN model summary

Classification: Here, initially, an image is flattened, then the ReLu activation function is applied to the input layer, and finally, the SoftMax activation function is applied to receive the accurate output. The ReLu activation function is usually used to improve the performance of the CNN.

The input image is flattened into a single long feature vector in a flattened image. Flatten is linked to the input layer for classification. It is used before the image pass so that the activation functions can classify it. Figure 6 shows the image classification result.

```
[[1. 0. 0. 0.]] --->>> 0
Disease Detected in leaf Image
[[1. 0. 0. 0.]] --->>> 0
Disease Detected in leaf Image
[[0.972 0.028 0.    0.   ]] --->>> 0
Disease Detected in leaf Image
[[0. 1. 0. 0.]] --->>> 1
Disease Detected in plant Image
[[0.    0.956 0.    0.044]] --->>> 1
Disease Detected in plant Image
[[0.003 0.    0.997 0.   ]] --->>> 2
Healthy Leaf Image
[[0. 0. 1. 0.]] --->>> 2
Healthy Leaf Image
[[0.    0.    0.999 0.   ]] --->>> 2
Healthy Leaf Image
[[0.    0.008 0.    0.992]] --->>> 3
Healthy Plant Image
[[0.    0.001 0.    0.999]] --->>> 3
Healthy Plant Image
```

Figure 6. Image Classification

Input Layer: The ReLu activation function is utilized in most CNN models to simplify the image and improve classification accuracy. The Input layer of classification is always hidden. All fully connected networks use the input layer as a hidden layer.
ReLu activation: The complete form of ReLu is Rectified Linear Unit. Because it does not trigger all neurons at once, the ReLu activation function is applied in the input layer of CNN. As a result, ReLu generates results in $(0, \infty)$.
Output Layer: The output layer of the model aims to determine the probabilities of identifying the class of each data item. The SoftMax activation function is applied in the output layer for data classification, which is sigmoid yet very useful for classifying the data. SoftMax plays a crucial function in the CNN Model, helping identify images and improve accuracy.

4. Results and Discussion

4.1 Experimental Setup

For training the model, we have taken the images from Kaggle. We have used a small dataset to find out the working of the CNN model with fewer parameters. Further validation and normalization of the data are done to train the dataset. Trained data is used for further processing after being labelled. To normalize the dataset, scale and reshape functions are used. Then pooling layer is used to eliminate all negative pixels, and final images are formed.

4.2 Model Summary: The model uses an Adam optimizer and a sequential model with layers of conv2d and max-pooling, then applies a 50 percent dropout to the first input layer before flattening it. The second input layer of the model uses 10 percent dropout and a dense layer, and the output layer of the CNN model uses 25 percent dropout and a dense layer. As a result, the proposed CNN model achieves higher accuracy. By researching other CNN and deep learning models, it was found that more than three layers are best for creating a CNN model. The Adam optimizer gives more than 99.2% accuracy, and it observed that the model required more than 100 epochs for the colour images to train and get higher accuracy. Optimizing the model by adding a dense layer and flattening it to deliver the best possible output is possible.

After running the training and testing dataset on the created model, we found the desired output, as shown in the figure image classification. The proposed model gives 99.38% accuracy. We use a colour image dataset for training and testing, so the model required 500 epochs to improve accuracy. Figure 7 shows the model accuracy for the dataset which applies to the model:

```
61/61 [==============================] - ETA: 0s - loss: 0.0752 - accuracy: 0.9739
Epoch 493: val_accuracy improved from 0.98765 to 0.99383, saving model to /content/drive/My Drive/cotton plant disease prediction/v3_pred_cott_dis.h5
61/61 [==============================] - 27s 443ms/step - loss: 0.0752 - accuracy: 0.9739 - val_loss: 0.0265 - val_accuracy: 0.9938
```

Figure 7. Training dataset result

The term "accuracy" refers to the ability to classify diseases appropriately. For example, we may have many images of diseased leaves or cotton plants. We are classifying the diseased leaf or plant by utilizing the classifier. We may be able to obtain accurate results through classification. The rate of getting the correct output is accurate. Accuracy is calculated with accuracy in percentage = 100* ((Number of correct classified leaves or plants) / (Total number of leaves or plants in Datasets))

The following graph in figure 8 shows the CNN model accuracy for the training and testing dataset.

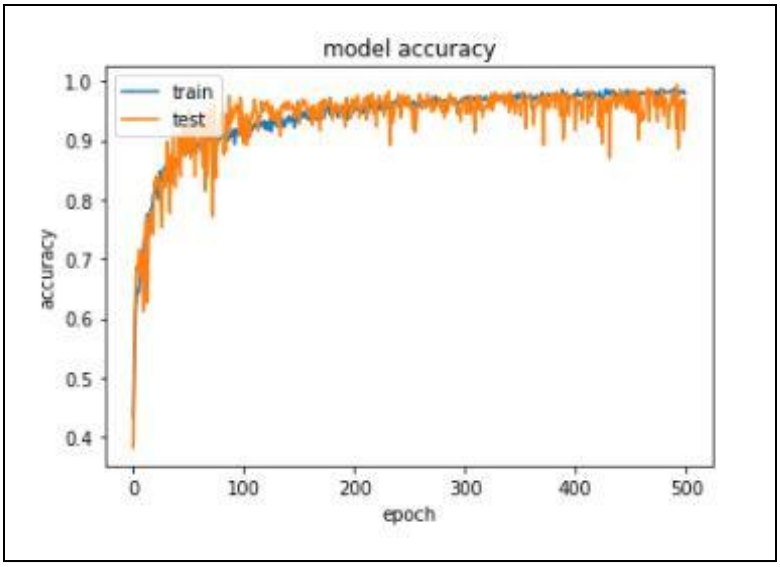

Figure 8. CNN Model Accuracy

The proposed CNN model and dataset are used in this research to show that classifiers can be used to compare and predict the accuracy of any comparison dataset. The prediction of healthy or diseased leaves was 99.38% accurate. This model is capable of identifying diseased and healthy leaves or plants. Hence model can predict the class of the input image easily.

5. Conclusion

To improve the production and yield of the cotton plant, it is necessary to detect plant diseases. It is better if disease detection is done at an early stage. The existing method uses disease detection of plants manually or with the necked eye. Deep learning plays an essential role in image processing to detect infected cotton leaves.

The proposed study presents a CNN model to solve the problem of the identification of diseases in cotton leaves. We have to use real-time data set containing images of cotton leaves from a Kaggle web-based data science environment. Initially, images are reprocessed and segmented based on their characteristics and quality with the help of the teacher standardization method. Furthermore, relevant features are extracted with a two-dimensional convolutional layer and Max-pooling with a convolution layer of kernel size 3. This results in a compressed image. Finally, the SoftMax function is applied to get the classification of the image. Images are classified into infected or not infected classes. Experimentation shows that the proposed model achieves 99.38% accuracy and is more effective in solving the cotton plant disease identification problem. In the future, we want to develop an application for farmers using intelligent devices with disease detection features. The application can include the identification of diseases in fruits, vegetables, and other plants.

References

[1] A. Eisha, S. Neeraj, G. Kamal, “OPNN: Optimized Probabilistic Neural Network-based Automatic Detection of Maize Plant Disease Detection”, 2021 6th International Conference on Inventive Computation Technologies (ICICT), DOI: 10.1109/ICICT50816.2021.9358763, (2021)
[2] A. Ammar Alhaj, C. Bronislav, J. Roman, K. Rasin, K. Said, “Classification of Plant Diseases Using Convolutional Neural Networks”, Artificial Intelligence in Intelligent Systems. CSOC, LNNS 229, PP 268-275, (2021)
[3] B. Stefania, M. Serban, O. Bogdan, M. Raul, T. Romulus, “Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture”, IEEE Access, (Volume: 9), (25 November 2021).
[4] B R Pushpa, A. Adarsh, A V Shree Hari, “Plant Disease Detection and Classification Using Deep Learning Model”, 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), DOI: 10.1109/ICIRCA51532.2021.9544729, (2021)
[5] G K Sagarika, SJ Krishna Prasad, S Mohana Kumar, “Paddy Plant Disease Classification and Prediction Using Convolutional Neural Network”, 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT) DOI: 10.1109/RTEICT49044.2020.9315634, (2020)
[6] L. K. Guan, H. C. Soon, T. M. Keng, L. Chung Fan, Y. Min, T. K. Tze Kin, “Optimization of Crop Disease Classification using Convolution Neural Network”, 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), DOI: 10.1109/IICAIET51634.2021.9573934, (2021)
[7] M. Aurangzeb, M. Javed Ahmed, R. Mirza Abdur, G. Sajid Habib, “Date Palm Disease Identification Using Features Extraction and Deep Learning Approach”, 2020 IEEE 23rd International Multitopic Conference (INMIC), DOI: 10.1109/INMIC50486.2020.9318158, (2020)
[8] M. Rajinder Kumar, D. Nagaraj V., “Early detection and identification of grape diseases using convolutional neural networks”, Journal of Plant Diseases and Protection, (Volume- 129), pp 521-532; https://doi.org/10.1007/s41348-022-00589-5, (2022)
[9] Prottasha M. S. I., Zarin T., Reza S. M. S.and Hossain D. A., “A Lightweight CNN Architecture to Identify Various Rice Plant Diseases in Bangladesh”, 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), pp. 316-320, DOI: 10.1109/ICICT4SD50815.2021.9396927, (2021)
[10] Rao U Sanath, R Swathia, V Sanjana; L Arpitha; K Chandrasekhar, Chinmayi, Naik Pramod Kumar, “Deep Learning Precision Farming: Grapes and Mango Leaf Disease Detection by Transfer Learning”, International Conference on Computing System and its Applications (ICCSA- 2021), Global Transitions Proceedings, 2(2), Pages 535-544, (2021)

[11] Raina Sakshi, Gupta Abhishek, “A Study on Various Techniques for Plant Leaf Disease Detection Using Leaf Image”, 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), DOI: 10.1109/ICAIS50930.2021.9396023, (2021)
[12] S. Nandhini, K. Ashokkumar, “Analysis on Prediction of Plant Leaf diseases using Deep Learning”, 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), DOI: 10.1109/ICAIS50930.2021.9395751, (2021)
[13] Thangaraj, R., Anandamurugan, S., Pandiyan, P. et al., “Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion”, J Plant Dis Prot 129, 469–488, https://doi.org/10.1007/s41348-021-00500-8, (2022)
[14] Thangaraj Rajasekaran, S. Anandamurugan, Kaliappan Vishnu Kumar, “Automated tomato leaf disease classification using transfer learning-based deep convolution neural network”, Journal of Plant Diseases and Protection, (Volume 128), pages73–86, (2021)
[15] Wang Jie, Zheng Yalan, Wang Min, Shen Qian, Huang Jiru, “Object-Scale Adaptive Convolutional Neural Networks for High-Spatial Resolution Remote Sensing Image Classification”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (Volume: 14), 283 – 299, DOI: 10.1109/JSTARS.2020.3041859, (2020)
[16] A. B. Yusuf, R. M. Dima, S. K. Aina, “Optimized Breast Cancer Classification using Feature Selection and Outliers Detection”, Journal of the Nigerian Society of PhysicalSciences, 298–307, DOI:10.46481/jnsps.2021.331, 2021
[17] V. Umarani, A. Julian, J. Deepa, “Sentiment Analysis using various Machine Learning and DeepLearning Techniques”, J. Nig. Soc. Phys. Sci. 3 (2021) 385–394, DOI:10.46481/jnsps.2021.308, (2021)