Cotton Plant Disease Prediction using Resnet50

MSc Research Project
Data Analytics

Ria Arora
StudentID:x20231385

School of Computing

National College of Ireland

Supervisor: Prof. Prashanth Nayak

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ria Arora
Student ID:	20231385
Programme:	MSc Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Prashanth Nayak
Submission Due Date:	15/12/2022
Project Title:	Cotton Plant Disease Prediction using Resnet50
Word Count:	7251
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Ria Arora
Date:	15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Cotton Plant Disease Prediction using ResNet50

Ria Arora
X20231385

Abstract

Plant diseases have become a major problem and have affected the economy adversely. Deep learning has helped on a large scale in detecting plant infections. Cotton is a widely grown crop and it is very important to detect the disease in it. Transfer learning plays a major role in the detection of infections which helps farmers save their plants from getting destroyed. Transfer learning has been used in this research which uses a previously trained model on a new one. ResNet50 transfer learning model has been chosen and data augmentation and fine-tuning have been performed which improves the performance of transfer learning models. The best part about ResNet50 is that it has over 23 million trainable parameters. One advantage of using the ResNet50 architecture is that it has shown strong performance in a wide range of tasks, including image classification, object detection, and semantic segmentation. This is due to its ability to learn deep, hierarchical representations of data, which allows it to capture complex patterns and features in the input. Google's Keras is a high-level deep learning API for creating neural networks. It is built in Python and is used to make neural network construction simple. It also allows for the calculation of various backend neural networks. Transfer learning has been proven to be the best method for disease detection. The best results are achieved with the data augmentation in the ratio 1:3 and fine-tuning the model. An accuracy of 96.9% has been achieved.

Keywords: Deep Learning, Transfer learning, Cotton Disease Detection, ResNet50.

Table of Contents

1 Introduction

Cotton is beneficial and has several advantages. It also contributes to the economic development of several countries. Hence, saving plants from getting destroyed due to diseases is important. It is very difficult for farmers to detect the disease with the naked eye and thus, a reliable and cost-effective method is required for disease detection. Cotton plants and leaves usually get diseased because of weather, wind, or insects. Crop rotation is a good practice to save crops from getting infected. Transfer learning will be used in this paper for finding the infections in cotton leaves and plants. By using this deep learning technique, we can flourish productivity by controlling diseases. ResNet50 model will be used. The residual neural network has fifty layers which include one max pool layer, one average pool layer, and 48 convolutional layers. If there are enough training samples available, the accuracy is high. The primary goal of writing this study is to assist farmers in adopting artificial intelligence to preserve their crops. Farmers can identify weeds and illnesses, assess plant quality, and forecast animal output. Transfer learning is a technique that involves using a pre-trained model as a starting point for training a new model on a different task. This can be useful for a variety of reasons, including reducing the amount of data and time needed to train a new model. In the context of predicting cotton plant diseases, transfer learning could potentially be used to help train a model to identify and classify different diseases based on images of cotton plants. By starting with a pre-trained model that has already been trained on a similar task, such as image classification, the new model can potentially learn more quickly and accurately than if it were trained from scratch on the cotton plant disease data. Of course, the effectiveness of transfer learning will depend on the quality and similarity of the pre-trained model and the new task. It may also be necessary to fine-tune the pre-trained model to better adapt it to the specific characteristics of the cotton plant disease data.

1.1 Research Questions and Objectives

The following research questions have been answered in this paper:

1. How is transfer learning effective in detecting cotton plant infections?
2. How is ResNet50 a good model for detecting plant diseases?
3. Can data augmentation and fine-tuning help in improving the accuracy of the model?

In this project, the main goal is to identify cotton plant and leaf diseases so that farmers can take immediate action to save them on time. The table below shows the objectives of this paper and the model and techniques used to get the best results:

Objective 1: Identifying the deep learning methods used in the identification of cotton plant diseases.

Objective 2: Implementation, evaluation, and the results of ResNet50.

Objective 3: Performing data augmentation in the ratio 1:1 and evaluating the results.

Objective 4: Performing data augmentation in the ratio 1:2 and evaluating the results.

Objective 5: Performing data augmentation in the ratio 1:3 and evaluating the results.

Objective 6: Fine-tuning the model to get better accuracy.

Data augmentation is the process used to increase the amount of data by altering the existing data. It helps increase the number of training data and therefore helps in increasing the accuracy.

In the fine-tuning method, a model that has already been trained for a given task is tuned to make a similar task.

2 Related Work

2.1 Transfer Learning models used for cotton disease identification

Deep CNN and transfer learning approaches have been used in a paper which has helped in reducing the estimation cost (Hassan, et al., 2021). Several transfer learning models like MobileNetV2, InceptionV3, EfficientNetB0, and InceptionResNetV2 have been compared and the best accuracy has been calculated. The calculations came out to be amazing and proved that CNN can be reliable in detecting plant diseases. In many rural places, farmers still use the primitive method of finding plant diseases with the naked eye which is time-consuming and needs attention. Some widely used classifiers like Support Vector Machines, Decision Trees, Random Forests, logistic Regress, naive Bayes, etc. can be used after feature extraction. The transfer learning approach on the other hand does not need a lot of implementation cost and gives promising results even with a small amount of data. The dataset used in the paper consists of 54,305 images of healthy and infected leaves. Firstly, the models were implemented with the colored leaf images and then the grayscale images of the same dataset were used, and the accuracy was calculated. The dataset contains 38 classes of leaf images. EfficientNetB0 has given the best results with 99.56 percent of accuracy. It takes not much time for training.

In a research paper, cotton bruises have been detected with the help of Deep convolutional networks. As cotton is grown in most of the world, it is also a very beneficial crop (Caldeira, et al., 2021) and must be protected at all costs. It is cultivated on around 2.5 percent of the world's arable land and is raised in about 100 different nations. To get the best solution to save the better way but GoogleNet also gave satisfactory results.

Algorithm	Overall Accuracy
SVM	80.30%
NFC	71.10%
RNA	76.60%
KNN	78.80%

Figure 1. Accuracy score of different machine learning algorithms.

India is an agriculture-based country (Rajasekar, et al., 2022), and it is mandatory to grow healthy crops. Cotton is one of the major crops grown but its leaves get spoilt or diseased easily which indirectly affects the economy. ResNet model is used in this paper which is pretrained on Xception and ImageNet components. The images are pre-processed, and the size of the images is 224*224 for ResNet. The dataset has 60,000 images. The 152-layer ResNet has been used. Training accuracy and validation accuracy achieved are 0.95 and 0.98 respectively.

No of Layers	Plain	ResNet
18 Layers	27.94	27.88
34 Layers	28.54	25.03

Figure 2. Result of plain and ResNet

The use of neural networks has been portrayed in the paper (Sagar & Dheeba, 2020). Different transfer learning architectures like ResNet50, InceptionV3, VGG16, InceptionResNet, and DenseNet169 have been evaluated. The paper proves that ResNet gives the best accuracy and has helped identify the diseases in less time achieving an accuracy of 0.98. The old methods of detecting leaves and plant diseases are time-consuming and costly. Modern methods must be adapted to ensure that diseases are detected at an early stage. This will be beneficial for the farmers as it will prevent the crops from getting spoiled. In larger fields where the crops are grown widely, the automated method will be of great help. The transfer learning method is quite advantageous as one does not need to start the learning process from the scratch. The model has already learned the previous learnings, and this can be used in solving another problem of a similar pattern. A dataset containing 54,000 images has been taken which includes diseased, healthy leaves, and leaves that are under controlled conditions. The images include different plant species. Image augmentation is performed, and all the images are resized to 256*256 pixels. Image zooming, flipping, changing the brightness, and shearing are done in the dataset which helped in getting much better accuracy. The images were randomly rotated from 0 to 45 degrees. Dropouts are done to avoid overfitting. ResNet50 has given the best recall and F1 score as well.

Bacterial leaf spot disease in bell peppers causes a lot of damage. (Thakur, et al., 2021) Used transfer learning over different models to identify disease in bell pepper plants. Currently, these diseases are only diagnosed in the lab, which is an expensive method, and to overcome this drawback, transfer learning has been taken into consideration. The results have been calculated using the three models ResNet50, VGG16, and Inception V3. Data has been altered in various ways to increase accuracy. A total of 2442 images of healthy and diseased bell pepper plant leaves have been taken and have been divided into 70 percent train data and 30 percent test data. For the VGG model, images have been resized to 224*224 and the color space is changed to the BGR model. The loss function used is Binary cross-entropy. ResNet50 is also resized to 224*224 and height shift, rotation, and flipping are done to the dataset. A batch size of 32 is taken and a total of 20 epochs are taken. The ReLu activation function is used, and the Adam optimizer is used. For the InceptionV3 model, the images are resized to 256*256, and then the data is augmented. Here again, Adam optimizer is used. VGG16 has a total of 21,203,778 parameters. The VGG16 has given an accuracy of 99.7 percent whereas ResNet50 is shown to be 98.59 percent accurate. InceptionV3 on the other hand is 95.77 percent accurate. The paper proves that the model can be used in real-time to reduce the incurred cost and give the best results. In the future, the models can be applied to real field images with background noise.

Classifier	Performance Metrics			
	Accuracy	*Loss*	*F1-score*	*AUC*
Vgg16	**99.72%**	**0.0116**	**0.9977**	**0.998**
ResNet50	99.31%	0.0165	0.9942	0.994
InceptionV3	95.77%	0.1240	0.9685	0.953

Figure 3. Performance comparison of VGG16, ResNet50 and InceptionV3

(Mohana Saranya, et al., 2021) proposed a paper on Tomato leaf disease identification using Deep CNN. Instead of normal data augmentation, deep convolutional generative adversarial networks are used. Results are obtained in the models like ResNet50, InceptionV3, and VGG16. The DCGAN architecture has two major components discriminator and a generator. The discriminator discriminates if the image is real or fake and the generator gives new images with the given input image. Batch size, optimizers, and drop-off values are taken, and performance was calculated. The dataset has 39 classes and 16,000 diseased images and 1591 healthy images. ResNet50 has given a very good accuracy as compared to the other models. DCGAN has helped in increasing the accuracy. The RMS prop hyperparameter optimization with 0.5 dropouts gave a good accuracy.

Cotton is one of the significant crops in India. (Bhatheja & Jayanthi, 2021), have used transfer learning and its models like InceptionV3, VGG19, ResNet152V2, and ResNet50. ResNet152V2 gave 98.36 percent and is shown to be a good model. Training is done using 12GB NVIDIA. The results came out to be promising results after running 10 epochs. But after running 30 epochs, the accuracy tends to improve more. Transfer learning has become very popular for disease detection. Execution time is reduced, and accuracy is increased.

Deep transfer learning is used by (De Luna, et al., 2019) to check tomato disease. There can be errors in the result if we try to detect the diseases with the naked eye. Three transfer learning models ResNet50, VGG16, and, InceptionV3 have been used to check the performance. A total of 1200 images have been taken and all the images have been resized to 224*224 for ResNet50 and VGG16 and 229*229 for InceptionV3. Dataset is divided into 20 percent validation and 80 percent training. Adam has been used as an optimizer. The highest accuracy achieved is 95.75 percent which is for VGG16.

(Turaev, et al., 2020) apply transfer learning to determine the quality of fruits and vegetables. GoogleNet, VGG16, VGG19, NasNetMobile, AlexNet, ResNet18, ResNet101, and ResNet50 are among the pre-trained models selected. The collection comprises photos of twelve fruits and vegetables totaling 4200 images, with a class total of 60. These characteristics are considered while evaluating the health of fruits and vegetables: nutritional value, texture, and color. Pre-trained models are used instead of constructing the CNN from scratch to save time and money. Each model is fine-tuned by removing the final three layers. CNN series and DAG (Directed Acyclic Graph) CNN models are used. The data is split into 60 percent for training and 40 percent for validation and testing. The greatest validation accuracy was achieved by VGG16 (91.50), followed by ResNet18 (91.37).

(Rautaray, et al., 2020) used transfer learning and convolutional neural networks (CNNs) to detect and classify diseases in paddy crops. They used four models (VGG16, LeNet-5, ResNet, and DensNet) and found that the use of transfer learning increased accuracy to 92%. This is higher than past work, which had an accuracy of only 70%. When using a single-layer CNN, accuracy was 72%, while using two layers increased accuracy to 75%. KNN and SVM models had an accuracy of 70%. The authors found that the VGG16 model was the most effective and could quickly identify the type of disease affecting a rice plant. This technique is useful for farmers, as it allows for the early detection of diseases and can help with decision-making.

2.2 Cotton Leaf disease detection using CNN

Food production is necessary as India's economy is largely dependent on agriculture (Kumbhar, et al., 2019). Convolutional neural networks help a lot in disease prediction. Images can be uploaded and with the help of digitalized color images, diseases can be detected. In this paper, two kinds of cotton diseases are predicted namely, bacterial blight and Alternaria macrospora. Images are resized to 128*128 pixels. The resized images are passed through different layers in CNN which are the hidden layer, pooling, and flattening layer. Once CNN is performed, the disease is the predicted using softmax layer. The dataset is divided into 70:30 train and test sets and cross-validation is applied. The activation function used is ReLu which helps in removing negative weights. A web application is developed where when the user uploads an image, data is pre-processed, CNN is performed, and the results are declared. Xampp server is used to deploy the Web Application in the computer. The training dataset has 513 photos, whereas the testing dataset contains 207 images. The training accuracy is 80, while the testing accuracy is 89 percent.

Reducing crop yield loss is important and can be achieved by early detection of diseases. The quality of crops can be preserved by addressing biological factors such as diseases, pests, and weeds. This can be done by regularly monitoring the fields and taking appropriate action based on the severity of the disease (Menon, et al., 2021). In this research, various CNN models were analyzed. First, pre-trained models were used, and new layers were added to the frozen layers. The transfer learning approach was used with VGG16, Xception, InceptionV3, ResNet152, and MobileNetV2. Two datasets were used: the Plant Village Dataset, consisting of 54,303 leaf images, and the Plantdoc Dataset, consisting of 2,598 images. Images of various crops, including tomatoes, bell peppers, apples, corn, and potatoes, were used. For the first dataset, six epochs were used for training to avoid under-fitting and over-fitting. MobileNetV2 and VGG16 were the most accurate models with 92% accuracy, while ResNet152 was the least accurate. For the second dataset, three epochs were considered, but the higher noise level resulted in less accurate results compared to the first dataset. The Xception model had the highest accuracy of 65%.

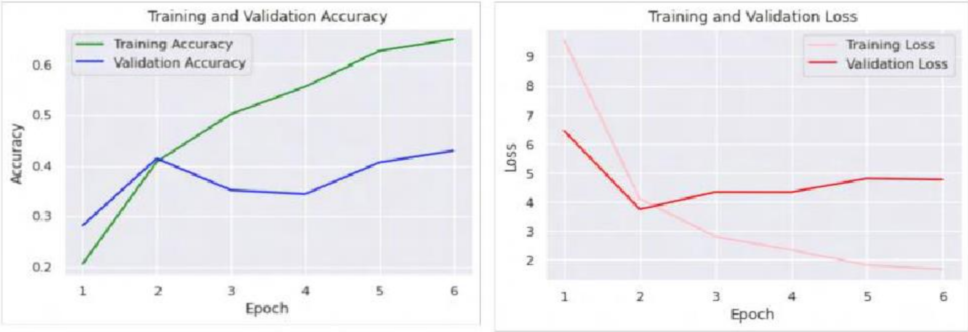

Figure 4. Exception model results when trained with the PlantDoc dataset

A study shows that CNN is a good model for disease identification of cotton plants (Zambare, et al., 2022). The dataset taken is quite small. When the Adam optimizer is used, the accuracy tends to be higher which is 99.38 percent. Color image dataset is used for training and testing and so 500 epochs were required for getting better results.

(Jenifa, et al., 2019) proposed a paper on cotton disease classification using a deep convolution neural network. The multi-layered CNN is one of the best models and is widely used and also is easy to implement. In this paper, 500 training images and 100 test images are used and images are resized to 512*512. MATLAB is used for the implementation. 96 percent accuracy is achieved and cotton diseases like Target spot, Bacterial blight, Cercospora, and Ascochyta blight have been detected accurately.

(Yadhav, et al., 2020) used CNN algorithms for plant disease detection and classification. The project is implemented in a Raspberry Pi kit. They used unsupervised learning, and the feature extraction is done using CNN. For optimization, the K-means clustering algorithm is used. The Adam optimizer is found to be better than the other optimizers. The accuracy achieved was greater than 95 percent. Different types of activation functions and optimizers are used, and the results are calculated. The processed picture is then grouped in MATLAB using a clustering method to determine the impacted region.

(Mohanty, et al., 2016) used deep learning for plant disease prediction. They used a dataset consisting of 54,306 images of healthy and diseased leaves. Altogether, 26 kinds of diseases have been detected in this paper. An accuracy of 99.35 percent has been achieved. Images have been resized to 256*256 pixels. The results are calculated with colored images, grayscaled images, and then segmented images. The AlexNet and GoogleNet models have been chosen for evaluation. GoogleNet has been seen performing much better as compared to AlexNet. Training and testing sets have been distributed in the ratio 80-20, 60-40, 50-50, 40-60, and 20-80.

(Matboli & Atia, 2022) used deep learning techniques to develop models for identifying fruit diseases. They compared five transfer learning models using a dataset of 5000 images, with 4500 for training and 500 for testing. They found that the CNN model, which included nine layers after top layer elimination and normalization, had the highest accuracy. The model achieved 96.96% accuracy on the apple dataset and 99.16% accuracy on the citrus dataset. InceptionV3, VGG16, and MobileNet also performed well, with dense layers and varying numbers of layers.

2.3 Cotton Crop Disease Detection using Machine Learning Algorithm

A paper by (Chopda, et al., 2018) used a Decision tree classifier for cotton crop disease prediction. Parameters like soil moisture and temperature have been taken to check the plant situation. Cotton diseases like Wilt and Anthracnose have been predicted. The real-time temperature data will be provided to the server where the Decision Tree classifier will display the output to the user. Arduino Uno Board receives the information from the sensors and then the decision tree comes into the picture. To build the decision tree classifier, training data is results. The results turned out to be quite promising.

(Maniyath, et al., 2018) proposed a paper on plant disease detection using machine learning. The proposed study contains several steps of implementation, including dataset construction, feature extraction, training the classifier, and classification. The images are resized to the same size. A histogram of an Oriented gradient is used for feature extraction. Random forest classifier is used as gives more accuracy even with less amount of data and gave the accuracy of 70 percent. This accuracy can be increased more with a large number of datasets. The images are taken with a plain background.

(Sarangdhar & Pawar, 2017) used the regression method for cotton leaf detection. Soil quality is monitored to analyze the condition of the plant. The Support Vector Machine based regression method is used to detect five kinds of cotton diseases namely Fusarium wilt, Mildew, Alternaria, Bacterial Blight, and Cereospra. The proposed method gave an accuracy of 83.26 percent. In total, 900 cotton leaf images are used out of which 271 images are used for testing and 629 images are used for training. SVM is found to be performing well.

(Jadhav, et al., 2019)proposed a paper on plant health prediction using machine learning. OpenCV library is used SVM is used for disease prediction. Images are taken and sent to the server after which segmentation is done and then the model is trained using SVM. After this, the model is tested using the test data and the results are evaluated. This helped to check if the plant is diseased or not.

A survey on plant disease prediction using machine learning by (Gokulnath & Usha Devi, 2020), has shown that machine learning has helped farmers a lot in crop disease prediction. Some of the challenges in agriculture are soil defects, irrigation techniques, weather conditions, water treatment, etc. Plats get diseases due to fungi and bacteria. The paper mentions the techniques used in plant disease detection which are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning methods. It also explains deep learning techniques and image processing techniques for disease prediction. The purpose of this work is to conduct a thorough investigation of the various computational approaches used in the plant disease diagnosis and classification system. Many clever algorithms were crucial in completing the required job.

(Domingues, et al., 2022) used machine learning for disease and pet prediction in the crops. Various ML techniques have been used and compared in this paper and tomato crops have been mostly focused on. After data processing and feature extraction, models are used for regression and classification. SVM performs well and can be used for regression models. Randon forest on the other hand gives more accuracy with less amount of data. Images with plain backgrounds give better results as compared to images that have field backgrounds and a lot of colors in them. Deep learning techniques need a large amount of data. So, to conclude, deep learning techniques are good for a large number of images and machine learning methods give better results with a smaller number of images.

3 Methodology

3.1 Cotton Plant Disease Detection Methodology Approach

A CRISP-DM methodology is a widely used approach to data mining. It stands for Cross-Industry Standard Process for Data Mining, and it provides a structured way of thinking about the data mining process. It consists of six main phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The goal of the CRISP-DM methodology is to ensure that data mining projects are carried out in a systematic and organized manner so that they can be completed efficiently and effectively.

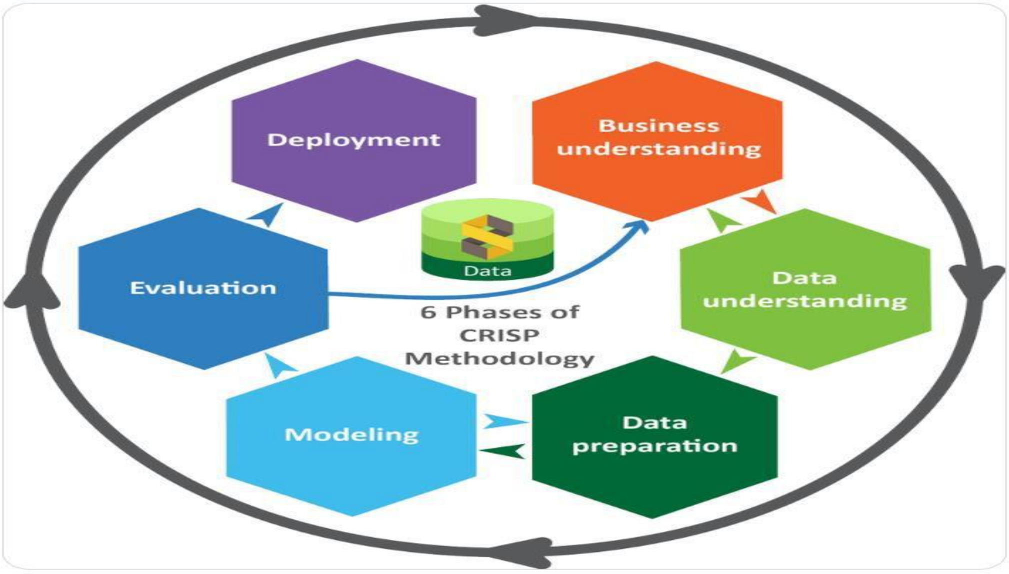

Figure 5. CRISP-DM methodology

The motive of this paper is to detect the diseases in cotton plants and leaves before it's too late and help the farmers. Necessary steps can be taken to prevent the diseases from spreading and in this way the economy can also be saved. A transfer learning model named ResNe50 has been used in this paper for disease identification as ResNet50 promises improvement in accuracy and requires less time to train.

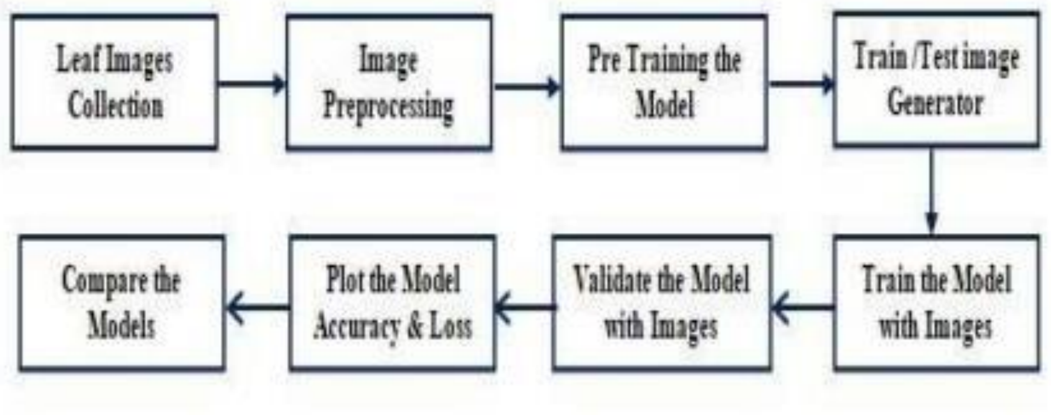

Figure 6. Block diagram for the model

3.2 Dataset Collection

The dataset has been provided by an Indian scientist Aksak Zade and is available publicly. The dataset[1] consists of four classes- fresh cotton leaf, diseased cotton leaf, fresh cotton plant, and diseased cotton plant.

Training set	Qty	Validation Set	Qty
Diseased Cotton leaf	288	Diseased Cotton leaf	55
Diseased cotton plant	815	Diseased cotton plant	101
Fresh Cotton leaf	427	Fresh Cotton leaf	80
Fresh cotton plant	421	Fresh cotton plant	88

Figure 7. Dataset description

Some samples were taken from the provided dataset.

Figure 8. Diseased Cotton Leaf **Figure 9. Diseased Cotton Plant**

Figure 10. Fresh Cotton Plant

Figure 11. Fresh Cotton Leaf

[1] https://drive.google.com/drive/folders/1vdr9CC9ChYVW2iXp6PlfyMOGD-4Um1ue

Dataset link:
https://drive.google.com/drive/folders/1vdr9CC9ChYVW2iXp6PlfyMOGD-4Um1ue

3.3 Data pre-processing

Image processing is the method that deals with altering images with the help of computer algorithms. The dataset is taken from open source and will not give a good accuracy, hence, to improve the accuracy of the model, data processing is performed. Some data processing steps include resizing the image and removing the noise from the data. The data is prepared in such a manner that it is easier for the model to understand it very clearly. There are four classes of images, and all the images are labeled correctly data is divided into the training set and test set. The original size of the images is 691*691, in the pre-processing step, all the images are resized to 224*224.

3.4 Data Exploration

Once the data is loaded into the model, the next step in the data exploration process is to pre-process the data. This typically involves a series of steps such as normalizing the data, removing any missing or erroneous values and transforming the data in a way that is suitable for the model. After pre-processing the data, the next step is to split the data into training and testing sets.

3.5 Data Augmentation

This is an important step and helps a lot in improving accuracy. This is done by a method called data augmentation. In this paper, data augmentation is performed in the ratio 1:1, 1:2, and 1:3 and the accuracy has been calculated for all three. Data augmentation helps in easier prediction for the cotton disease because it increases the data which helps in better evaluation. The data is prepared in such a manner that it is easier for the model to understand it very clearly. There are four classes of images, and all the images are labeled correctly data is divided into the training set and test set. Roboflow has been used to perform data augmentation. Roboflow is a developer framework for Computer Vision that enables improved data collecting, pre-processing, and model training procedures. It supports a variety of annotation formats. Image orientations, resizing, contrasting, and data augmentations are all phases of data pre-processing.

3.6 Initializing the deep learning models

ResNet50 is a convolutional neural network that is trained on the ImageNet dataset. It is a deep learning model that can recognize and classify objects in images with high accuracy. It was developed by researchers at Microsoft and is widely used in computer vision applications. The ”50” in its name refers to the fact that it has 50 layers, which makes it a deeper network than many other models.

The architecture of ResNet50 is based on the concept of residual learning, which is the idea that it is easier for a network to learn the residual, or the difference between the desired output and the current output, rather than the desired output itself. To achieve this, ResNet50 uses skip connections, which are connections that bypass one or more layers, allowing the input to be directly added to the output of a layer further down the network. This helps to alleviate the problem of vanishing gradients, which occurs when the gradients of the network become very small, making it difficult for the network to learn.

The architecture of ResNet50 consists of four main components: the convolutional layers, the activation layers, the pooling layers, and the fully connected layers. The convolutional layers are responsible for extracting features from the input data, such as edges and textures. The activation layers apply a non-linear function to the output of the convolutional layers, allowing the network to learn more complex patterns. The pooling layers down-sample the data, reducing the spatial dimensions of the data while retaining the most important information. Finally, the fully connected layers learn to classify the data based on the features extracted by the convolutional and pooling layers.

Overall, the ResNet50 architecture is a deep neural network that is designed to be both deep and accurate. Its use of residual learning and skip connections helps to improve its performance on a wide range of tasks, such as image classification, object detection, and semantic segmentation.

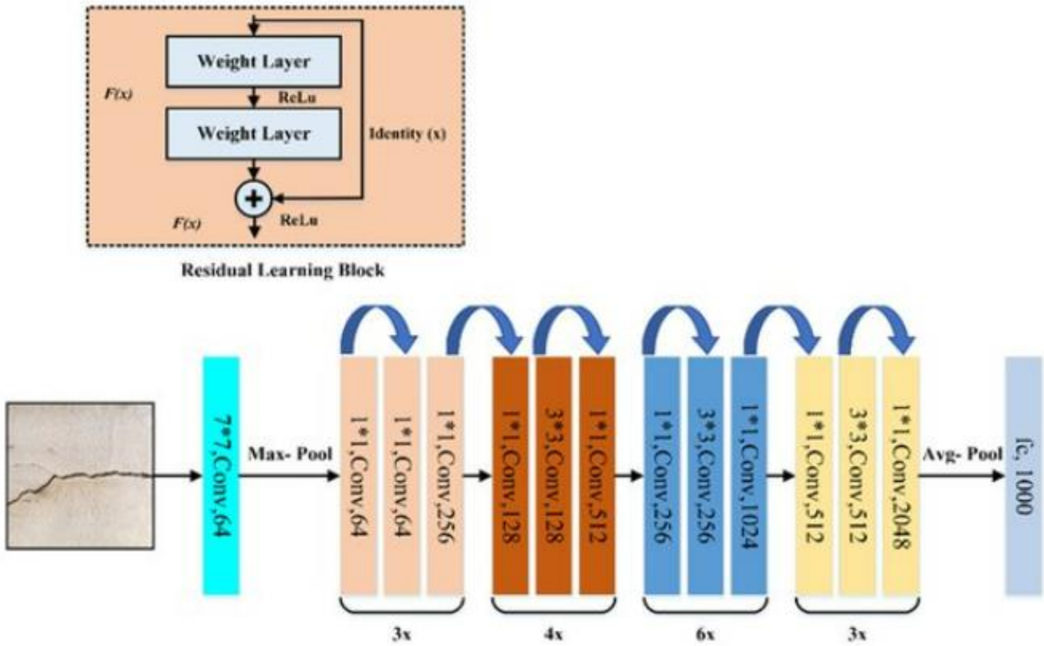

Figure 12. ResNet50 model architecture

4 Design Specification

The below flow chart shows the steps carried out to implement the project. It highlights the detailed roadmap for how the model will be built and used. Data is divided into a 70% training set, a 20% validation set, and a 10% testing set for the 1:1 dataset. For the 1:2

dataset, it is divided into 82% training set, 12% validation set, and 6% testing set whereas for 1the :3 dataset, it is 87% training set, 8% validation set, and 4% testing set. 14,000 images of train sets and 194 images of test sets have been taken for the baseline dataset. After performing the data augmentation in the ratio 1:2, 27,000 train images and 195 test images are taken. After data augmentation 1:3, 41,000 train images and 195 test images are taken.

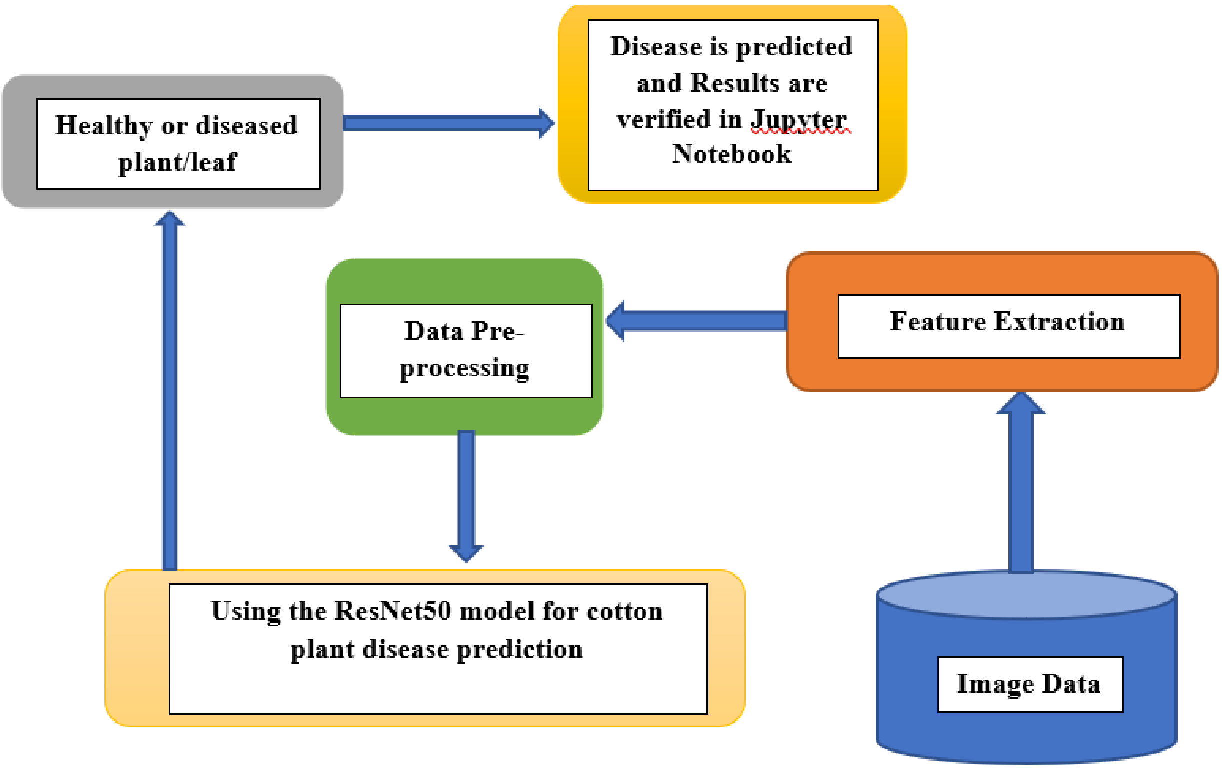

Figure 13. Design Specification Flow Diagram for Detecting Cotton Disease

5 Implementation

The ResNet50 model reads all the images and predicts the disease in cotton plants and leaves.

5.1 Project setup by Hardware and Software

Experimental setup: The performance of the model is checked using Jupyter Notebook under Anaconda IDE. To perform deep learning tasks, the Python programming language provides pre-defined libraries such as Keras, TensorFlow, and PyTorch for working with image data. For plotting and data visualization, Python offers libraries like pandas, Matplotlib, and seaborn is used.

5.2 ResNet50 model

Step 1: Data collection and preparation- The first step is to collect a large dataset of images of cotton plants with and without different types of diseases. The images need to be pre-processed by resizing and normalizing the pixel values (224*224).

Step 2: Defining the ResNet50 model- The next step is to define the ResNet50 model using the Keras functional API. This involves specifying the input layer, the residual blocks, the global average pooling layer, and the output layer. We can use transfer learning to fine-tune a pre-trained ResNet50 model on the cotton plant disease dataset.

Step 3: Compile the model- After defining the model, you need to compile it by specifying the optimizer, here Adam is used and the loss function which is categorical_crossentropy for this research, and the metrics to be used for evaluating the model's performance.

Step 4: Model training- Once the model is compiled, it is trained on the prepared dataset by specifying the number of epochs and the batch size. The number of epochs taken is 20 and the callback function is used. Callback functions are often used in event-driven programming to specify the behavior of a program when a certain event occurs, such as a user clicking a button or a network request completing. Here data augmentation techniques are used such as random horizontal flips and random crops to increase the diversity of the training data and improve the model's generalization performance.

Outputs per training example: 1,2,3
90° Rotate: Clockwise, Counter-Clockwise
Rotation: Between -15° and +15 °
Bounding Box: Rotation: Between -30° and +30°

Step 5: Model evaluation- After training, the performance of the model is evaluated on a separate validation dataset, and check the metrics such as accuracy and top-5 error rate.

Step 6: Use the trained model- Once the model is trained and evaluated, it is used for making predictions on new unseen images of cotton plants. The model can classify the images into different classes of diseases or healthy plants and provide probabilities for each class.

6 Evaluation

The model metrics that are calculated are loss, accuracy, validation loss, and validation accuracy. Loss and accuracy are metrics that are used to evaluate the performance of a machine-learning model. Loss is a measure of how well the model can make predictions on a given dataset, while accuracy is a measure of how well the model's predictions match the true values in the dataset. Validation loss and validation accuracy are similar to the loss and accuracy, but they are computed on a separate dataset called the validation set. This dataset is typically used to evaluate the model's performance during the training process, and it is distinct from the training set and the test set. To calculate loss, one typically uses a loss function, which is a mathematical function that takes in the predicted values from the model and the true values from the dataset and outputs a numeric value that represents how well the model is performing. For example, the mean squared error loss function calculates the difference between the predicted and true values for each data point, squares the difference, and then takes the average of all of these squared differences. To calculate accuracy, one typically compares the model's predictions to the true values in the dataset and counts the number of times that the model's predictions match the true values. This count is then divided by the total number of data points in the dataset to give the overall accuracy. Validation loss and validation accuracy are calculated in the same way, but using the validation set instead of the full dataset. This allows one to evaluate the model's performance on a separate set of data, which can help to prevent overfitting and ensure that the model is generalizing well to new data.

6.1 Stage 1: Calculating metrics for baseline dataset

The below metrics are obtained after running the baseline dataset. The overall evaluation accuracy achieved on the validation dataset is approximately 78%. We can notice that the loss and validation loss are more or less the same across all epochs. This suggests that the model is more likely to be a perfect fit.

```
Epoch 1/20
61/61 [==============================] - 31s 503ms/step - loss: 1.2868 - accuracy: 0.4936 - val_loss: 1.1249 - val_accuracy: 0.5556
Epoch 2/20
61/61 [==============================] - 18s 297ms/step - loss: 0.9862 - accuracy: 0.6053 - val_loss: 1.1933 - val_accuracy: 0.6111
Epoch 3/20
61/61 [==============================] - 18s 297ms/step - loss: 0.9024 - accuracy: 0.6412 - val_loss: 0.7966 - val_accuracy: 0.7778
Epoch 4/20
61/61 [==============================] - 18s 297ms/step - loss: 0.8691 - accuracy: 0.6627 - val_loss: 0.8158 - val_accuracy: 0.6667
Epoch 5/20
61/61 [==============================] - 18s 298ms/step - loss: 0.9322 - accuracy: 0.6361 - val_loss: 1.0051 - val_accuracy: 0.6667
Epoch 6/20
61/61 [==============================] - 18s 297ms/step - loss: 0.8368 - accuracy: 0.6730 - val_loss: 0.6658 - val_accuracy: 0.7778
Epoch 7/20
61/61 [==============================] - 19s 305ms/step - loss: 0.8500 - accuracy: 0.6694 - val_loss: 0.8086 - val_accuracy: 0.7222
Epoch 8/20
61/61 [==============================] - 18s 303ms/step - loss: 0.7391 - accuracy: 0.7176 - val_loss: 0.6687 - val_accuracy: 0.7778
Epoch 9/20
61/61 [==============================] - 18s 300ms/step - loss: 0.7539 - accuracy: 0.6914 - val_loss: 0.5906 - val_accuracy: 0.7778
Epoch 10/20
61/61 [==============================] - 18s 296ms/step - loss: 0.6641 - accuracy: 0.7381 - val_loss: 1.0657 - val_accuracy: 0.7222
Epoch 11/20
61/61 [==============================] - 18s 298ms/step - loss: 0.9751 - accuracy: 0.6674 - val_loss: 0.8098 - val_accuracy: 0.7778
Epoch 12/20
61/61 [==============================] - 18s 298ms/step - loss: 0.7192 - accuracy: 0.7258 - val_loss: 0.7715 - val_accuracy: 0.7778
Epoch 13/20
61/61 [==============================] - 19s 307ms/step - loss: 0.7212 - accuracy: 0.7283 - val_loss: 1.2519 - val_accuracy: 0.7222
Epoch 14/20
61/61 [==============================] - 23s 369ms/step - loss: 0.6548 - accuracy: 0.7509 - val_loss: 0.7284 - val_accuracy: 0.7778
Epoch 15/20
61/61 [==============================] - 18s 295ms/step - loss: 0.6957 - accuracy: 0.7278 - val_loss: 0.5299 - val_accuracy: 0.8333
Epoch 16/20
61/61 [==============================] - 18s 296ms/step - loss: 0.7172 - accuracy: 0.7191 - val_loss: 0.7328 - val_accuracy: 0.7778
Epoch 17/20
61/61 [==============================] - 18s 294ms/step - loss: 0.6455 - accuracy: 0.7509 - val_loss: 0.6202 - val_accuracy: 0.7778
Epoch 18/20
61/61 [==============================] - 18s 294ms/step - loss: 0.6164 - accuracy: 0.7565 - val_loss: 0.4560 - val_accuracy: 0.8333
Epoch 19/20
61/61 [==============================] - 18s 295ms/step - loss: 0.6614 - accuracy: 0.7576 - val_loss: 0.9876 - val_accuracy: 0.7222
Epoch 20/20
61/61 [==============================] - 18s 294ms/step - loss: 0.7661 - accuracy: 0.7355 - val_loss: 0.5443 - val_accuracy: 0.7778
```

Figure 14. Loss and accuracy metrics

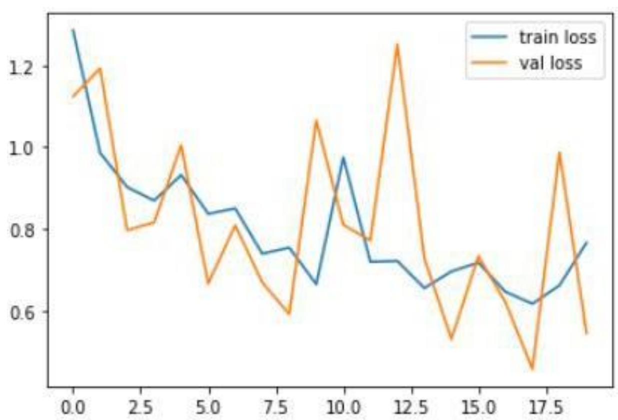

Figure 15. Training and validation accuracy

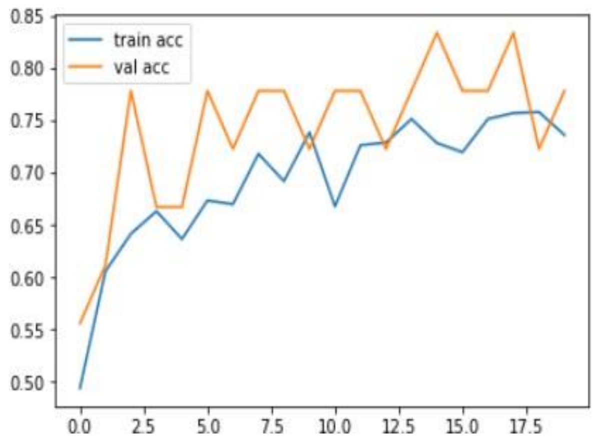

Figure 16. Training and validation loss

6.2 Stage 2: Fine-tuning the data augmentation dataset in the ratio 1:1

After performing the data augmentation in the ratio 1:1, and fine-tuning it, the validation accuracy of 99%. The model seems to be a good fit. In this step, the number of trainable parameters was increased thereby increasing the accuracy at the cost of production and the speed at which the model trains. The validation loss and loss for this dataset indicate the fact that the model is not overfitting. Figure 17 shows the training progress and the validation accuracy over 20 epochs.

	loss	accuracy	val_loss	val_accuracy
0	1.989164	0.444900	6.794571	0.169753
1	1.013295	0.656586	9.147050	0.151235
2	0.753111	0.754997	3.106287	0.222222
3	0.565374	0.843157	0.532191	0.839506
4	0.479142	0.868273	0.241044	0.969136
5	0.438823	0.895951	0.190389	0.984568
6	0.388978	0.905177	0.184790	0.984568
7	0.374618	0.912865	0.189572	0.978395
8	0.329812	0.927729	0.169246	0.987654
9	0.319666	0.933367	0.175377	0.984568
10	0.295712	0.942081	0.167204	0.987654
11	0.260245	0.953357	0.157817	0.990741
12	0.239819	0.961046	0.180034	0.978395
13	0.249912	0.959508	0.170274	0.987654
14	0.229966	0.962071	0.152931	0.987654
15	0.211353	0.968734	0.172531	0.984568
16	0.219399	0.966171	0.162934	0.990741
17	0.206364	0.976422	0.157891	0.990741
18	0.224981	0.970784	0.153508	0.987654
19	0.200212	0.978473	0.141225	0.993827

Figure 17. Loss and accuracy metrics for the 1:1 dataset

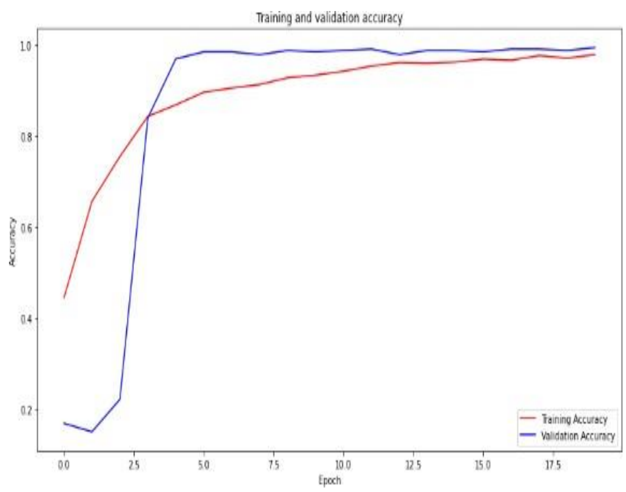

Figure 18. Training and validation accuracy

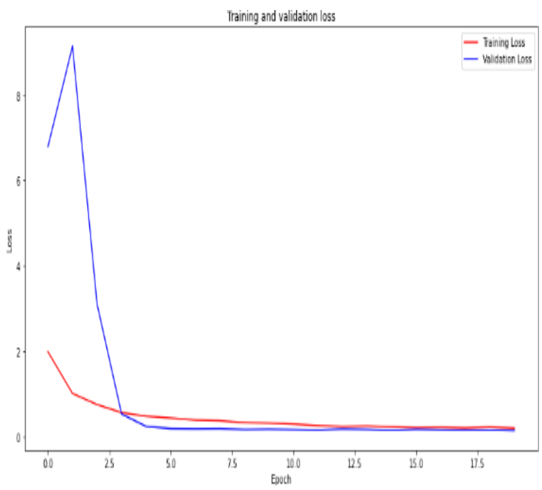

Figure 19. Training and validation loss

6.3 Stage 3: Fine-tuning the data augmentation 1:2

This stage introduces more noise to the model in an attempt to train the model to be more robust. We can notice that the noise in the dataset has reduced the model performance by a small fraction. The model was scheduled to train over 20 epochs initially but was cut short by the callback function to prevent the model from overfitting. We can notice that the validation accuracy for the model is at 96% and is not too far behind the training accuracy.

	loss	accuracy	val_loss	val_accuracy
0	1.640672	0.501662	5.878012	0.271318
1	0.846501	0.718138	0.366229	0.917313
2	0.605705	0.822682	0.289902	0.935401
3	0.508092	0.853343	0.272865	0.945736
4	0.414887	0.892870	0.240135	0.953488
5	0.377371	0.911710	0.233439	0.968992
6	0.349994	0.919837	0.238258	0.961240
7	0.344022	0.928334	0.265207	0.950904
8	0.276126	0.944588	0.302661	0.963824
9	0.279057	0.949760	0.284512	0.961240
10	0.282365	0.946805	0.260505	0.966408

Figure 20. Metrics for 1:2 dataset

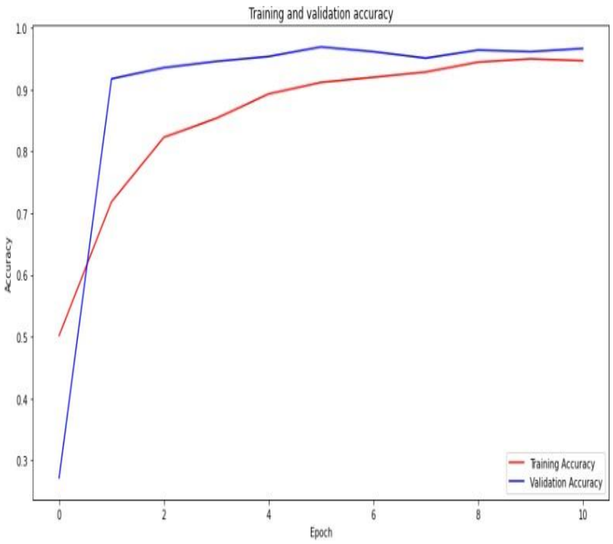

Figure 21. Training and validation accuracy

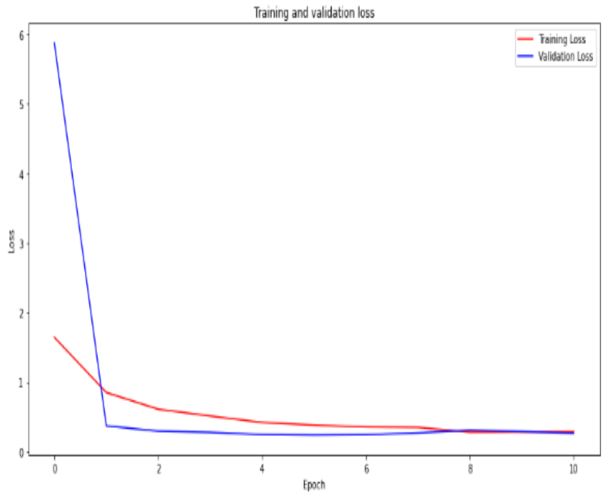

Figure 22. Training and validation loss

6.4 Stage 4: Fine-tuning the data augmentation 1:3

The final stage of model building involves introducing more noise to the data and recording the model behavior. In this stage, the size of the dataset is increased by 3 times. It can still be noticed that the model was performing well with a validation accuracy of 96% while having a very low loss.

	loss	accuracy	val_loss	val_accuracy
0	0.242541	0.954545	0.249024	0.963855
1	0.205628	0.969902	0.221153	0.965577
2	0.184319	0.971437	0.216771	0.969019
3	0.177677	0.974509	0.235439	0.969019
4	0.142515	0.984951	0.236770	0.967298
5	0.141136	0.986179	0.239100	0.962134
6	0.129488	0.989558	0.239502	0.963855
7	0.142740	0.985872	0.223353	0.969019

Figure 23. Metrics for 1:3 dataset

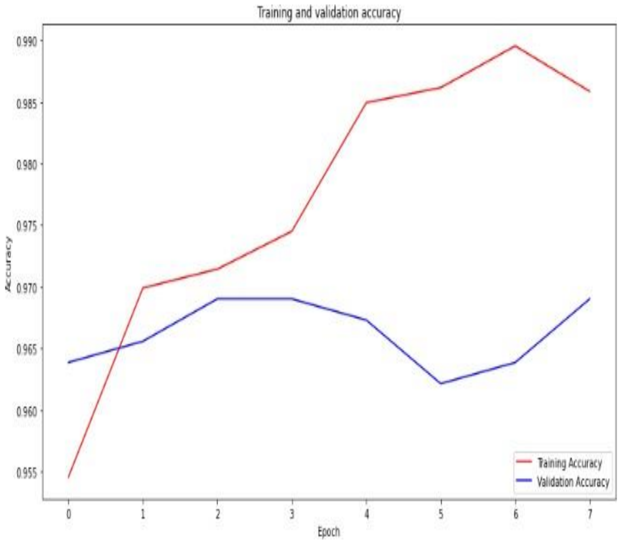

Figure 24. Training and validation accuracy

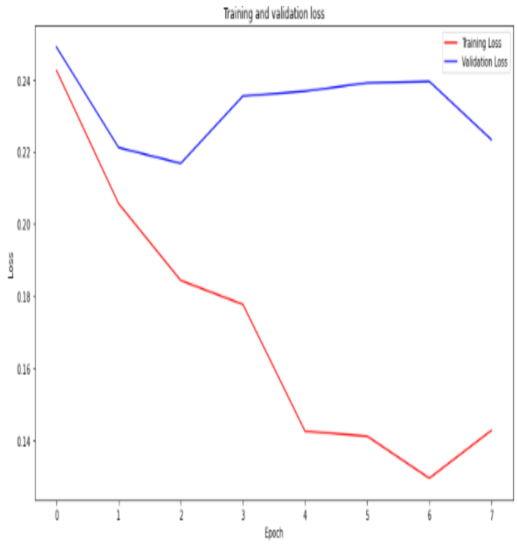

Figure 25. Training and validation loss

Hence, we can conclude that the model trained on 1:3 data augmentation is the best performing considering the amount of noise present in the training. It also makes the model robust and is likely to work better in real-time data.

6.5 Discussion

The focus of this research is on the application of Deep Learning and Transfer Learning techniques. The findings of this study are based on the use of Deep Learning models and represent a novel and effective approach. To carry out the research, a baseline dataset has been taken and it is fine-tuned with the augmented data in the ratio 1:1, 1:2, and 1:3. Fine-tuning the data augmentation 1:3 gives the accuracy of 96.9% in the seventh epoch itself which is a very good accuracy.

7 Conclusion and Future Work

In conclusion, the ResNet50 model is a powerful deep-learning tool that can be used for cotton plant disease detection. By training the model on a large dataset of images of cotton plants with and without different types of diseases, it is possible to achieve high accuracy and a top-5 error rate on the validation dataset. The trained model can be used to classify images of cotton plants into different classes of diseases or healthy plants, and to perform object detection and segmentation to identify the diseased regions in the plants.

However, there are also limitations to using the ResNet50 model for cotton plant disease detection. The model relies on a large and diverse dataset for training, and its performance may degrade when applied to images that are significantly different from the training data. The model also requires significant computational resources for training and inference and may not be practical for use on mobile or edge devices.

Overall, the ResNet50 model is a promising approach for cotton plant disease detection, but there is still much room for improvement and further research in this area. There are many potential avenues for future work on cotton plant disease detection using the ResNet50 model. Some possibilities include:

1. Improving the model's performance: One potential direction is to improve the model's performance by using more advanced deep learning techniques such as assembling multiple models, using larger and more diverse datasets, and incorporating domain-specific knowledge into the model.
2. Developing a real-time cotton plant disease detection system: Another direction is to develop a real-time cotton plant disease detection system that can be deployed in the field and used by farmers or agricultural workers to quickly and accurately identify diseased plants. This could involve designing a user-friendly interface for capturing and uploading images and implementing efficient algorithms for running the ResNet50 model on mobile or edge devices.
3. Investigating the use of other deep learning models: In addition to the ResNet50 model, many other deep learning models have been developed for image classification and object detection tasks. It would be interesting to investigate the use of other models such as VGGNet, Inception, or Mask R-CNN for cotton plant disease detection and compare their performance with the ResNet50 model.
4. Studying the impact of cotton plant diseases on yield and quality: Another potential direction is to study the impact of cotton plant diseases on the yield and quality of cotton crops and use the results to develop strategies for mitigating the effects of diseases on cotton production. This could involve collecting and analyzing data on cotton plant diseases and their effects and developing predictive models for estimating the impact of diseases on yield and quality.
5. Collaborating with agricultural experts: Finally, it would be valuable to collaborate with agricultural experts and organizations to integrate the cotton plant disease detection system into existing agricultural practices and to promote its adoption by farmers and other stakeholders. This could involve conducting field trials, providing

training and support, and fostering a community of users and contributors to the system.

References

Bhatheja, H. & Jayanthi, N., 2021. *Detection of Cotton Plant Disease for Fast Monitoring Using Enhanced Deep Learning Technique.* s.l.:s.n.

Caldeira, R. F., Santiago, W. E. & Teruel, B., 2021. *Identification of cotton leaf lesions using deep learning techniques.* s.l.:s.n.

Chopda, J., Raveshiya, H., Nakum, S. & Nakrani, V., 2018. *Cotton Crop Disease Detection using Decision Tree Classifier.* s.l.:s.n.

De Luna, R. G., Dadios, E. P., Bandala, A. A. & Vicerra, R. R. P., 2019. *Tomato Fruit Image Dataset for Deep Transfer Learning-based Defect Detection.* s.l.:s.n.

Domingues, T. B. T. a. F. J., 2022. Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey.

Gokulnath, B. V. & Usha Devi, G., 2020. *A survey on plant disease prediction using machine learning and deep learning techniques.* s.l.:s.n.

Hassan, S. M. et al., 2021. *Identification of plant-leaf diseases using CNN and transfer-learning approach.* s.l.:s.n.

Islam, F., Hoq, M. N. & Rahman, C. M., 2019. *Application of Transfer Learning to Detect Potato Disease from Leaf Image.* s.l.:s.n.

Jadhav, A. S. P. G. H. a. R., 2019. Plant health prediction using machine learning.

Jenifa, A., Ramalakshmi, R. & Ramachandran, V., 2019. *Cotton Leaf Disease Classification using Deep Convolution Neural Network for Sustainable Cotton Production.* s.l.:s.n.

Kumbhar, S. et al., 2019. *Farmer Buddy-Web Based Cotton Leaf Disease Detection Using CNN.* s.l.:s.n.

Maniyath, S. R. et al., 2018. *Plant disease detection using machine learning.* s.l.:s.n.

Matboli, M. A. & Atia, A., 2022. Fruit Disease's Identification and Classification Using Deep Learning Model.

Menon, V., Ashwin, V. & Deepa, R. K., 2021. *Plant Disease Detection using CNN and Transfer Learning.* s.l.:s.n.

Mohana Saranya, S. et al., 2021. *Optimized Deep Convolutional Generative Adversarial Network for Tomato Leaf Disease Identification.* s.l.:s.n.

Mohanty, S. P., Hughes, D. P. & Salathé, M., 2016. *Using deep learning for image-based plant disease detection.* s.l.:s.n.

Rajani Zambare, R. D. C. A. S. S. S. T. S. Z., 2022. Deep Learning Model for Disease Identification of Cotton Plants.

Rajasekar, V. et al., 2022. *Detection of Cotton Plant Diseases Using Deep Transfer Learning.* s.l.:s.n.

S.S Rautaray, M. P. R. S., 2020. Paddy crop disease prediction—a transfer learning technique. *International Journal of Recent Technology and Engineering.*

Sagar, A. & Dheeba, J., 2020. *On Using Transfer Learning For Plant Disease Detection,* s.l.: s.n.

Sarangdhar, A. A. & Pawar, V., 2017. *Machine learning regression technique for cotton leaf disease detection and controlling using IoT.* s.l.:s.n.

Sarangdhar, A. A. & Pawar, V. R., 2017. Machine learning regression technique for cotton leaf disease detection and controlling using IoT.

Thakur, P., Chug, A. & Singh, A. P., 2021. *Plant Disease detection of Bell Pepper Plant Using Transfer Learning over different Models.* s.l.:s.n.

Turaev, S., Almisreb, A. A. & Saleh, M. A., 2020. *Application of Transfer Learning for Fruits and Vegetable Quality Assessment.* s.l.:s.n.

Yadhav, S. Y., Senthilkumar, T., Jayanthy, S. & Kovilpillai, J. J. A., 2020. Plant Disease Detection and Classification using CNN Model with Optimized Activation Function. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020.*