Disease Detection in Rubber Leaves using Transfer Learning with GAN

03CS7914 Project (Phase II)

CHN20MT006 CHN20CSIP02 Karthika G Nath
chn20csip02@ceconline.edu
M. Tech. Computer Science & Engineering (Image Processing)

Department of Computer Engineering
College of Engineering Chengannur
Alappuzha 689121
Phone: +91.479.2165706
http://www.ceconline.edu
hod.cse@ceconline.edu

College of Engineering Chengannur
Department of Computer Engineering

C E R T I F I C A T E

This is to certify that, this report titled ***Disease Detection in Rubber Leaves using Transfer Learning with GAN*** is a bonafide record of the work done by

CHN20MT006 CHN20CSIP02 Karthika G Nath

Fourth Semester M. Tech. Computer Science & Engineering (Image Processing)

student, for the course work in **03CS7914 Project (Phase II)**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, M. Tech. Computer Science & Engineering (Image Processing) of **APJ Abdul Kalam Technological University**.

Guide	Senior Guide	Coordinator
Raji A Asst.Professor Electronics & Communication Engineering	Dr. Smitha Dharan Principal	Ahammed Siraj K K Associate Professor Computer Engineering

Head of the Department

July 16, 2022

Dr. Manju S Nair
Associate Professor
Computer Engineering

Permission to Use

In presenting this project dissertation at College of Engineering Chengannur(CEC) in partial fulfillment of the requirements for a postgraduate degree from APJ Abdul Kalam Technological University, I agree that the libraries of CEC may make it freely available for inspection through any form of media. I further agree that permission for copying of this dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the Head of the Department of Computer Engineering. It is understood that any copying or publication or use of this dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to CEC in any scholarly use which may be made of any material in this project dissertation.

Karthika G Nath.

Statement of Authenticity

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at College of Engineering Chengannur(CEC) or any other educational institution, except where due acknowledgement is made in the report. Any contribution made to my work by others, with whom I have worked at CEC or elsewhere, is explicitly acknowledged in the report. I also declare that the intellectual content of this report is the product of my own work done as per the **Problem Statement** and **Proposed Solution** sections of the project dissertation report. I have explicitly stated the major references of my work. I have also listed all the documents referred, to the best of my knowledge.

Karthika G Nath.

Acknowledgements

Primarily, I thank Lord Almighty for his eternal support through out my Project work.

I express my sincere thanks to **Dr. Smitha Dharan**, Principal, College of Engineering Chengannur for extending all the facilities required for doing my project. My heartfelt words of gratitude to **Dr. Manju S Nair**, Associate Professor and Head of Department of Computer Engineering, for providing constant support.

I express my gratitude to my project co-ordinator **Ahammed Siraj K K**, Associate Professor in Computer Engineering and my project guides **Dr. Smitha Dharan**, Principal and **Raji A**, Assistant Professor in Electronics and Communication Engineering who played a great role for valuable suggestions and expert guidance.

Now I would like to express my grateful thanks to **Dr. Shaji Philip**, Scientist, Rubber Research Institute of India(RRII) Kottayam, who helped me lot in gathering different information, collecting rubber leaf image dataset from RRII plantations and guiding me from time to time.

I would also like to express my gratitude to my parents for their constant moral and economic support. Last but not the least goes to all my friends who have directly or indirectly helped me for the completion of this project.

Karthika G Nath

Abstract

Agriculture is considered to be the backbone of the economic system for developing countries. An essential field of research in the agriculture industry is the identification of plant diseases. Plant diseases are seen as a problem in the agriculture sector because they affect the quality, management, and output of crops. Therefore, it is essential to identify and detect these diseases as early as possible. The use of deep learning techniques has significantly improved the accuracy and precision of the diagnosis of plant diseases. This work presents, a deep learning-based method to detect diseases in rubber leaves using a Deep Convolutional Generative Adversarial Network (DCGAN) which generate synthetic images of rubber plant leaves. Then a pretrained DenseNet121 model based on transfer learning is used to categorise the diseases in rubber leaves.

Contents

List of Figures

Chapter 1

Introduction

In many countries like India, the majority of people rely heavily on agriculture as a source of income. Agricultural crops may lose a significant proportion of yield as a result of diseases that affect the leaves, fruits and stems of plants. For agricultural productivity to be increased in a sustainable manner, rapid and precise plant disease identification is essential. The traditional method of identifying and diagnosing leaf diseases involves professional observation using naked eyes. It is not practical to monitor large farms this way since it takes time and competence. Nowadays, identifying and classifying plant diseases is a difficult task. The symptoms of plant diseases can be seen in several plant components, although leaves are the most frequently seen area for spotting an infection. Automatic disease identification offers a way to solve agricultural issues quickly and efficiently. The rapid advancement of deep learning techniques has considerably aided in the early diagnosis of plant diseases. The quantity and caliber of labelled training data determine how accurate deep learning models are. Deep learning is a branch of machine learning techniques built on artificial neural networks and representation learning.

Rubber plant (Hevea brasiliensis) is one of the most cultivated plant which yields economic benefit to farmers. Every year, a large amount of latex are produced due to the increasing of rubber-based products in the market. Therefore it is necessary to maintain the quality and quantity. A variety of diseases can affect the growth of rubber trees, which reduces the production and quality of natural rubber. The diseases in rubber leaves affect the tree and causes problems such as, reduce the latex yield, damage the tapping panel and affect bark, kill branches, lengthen immaturity period etc. Thus, care should be given to the rubber trees, so that they are free from diseases. Most commonly seen rubber leaf diseases are Powdery Mildew (PM), Corynespora Leaf Fall (CLF), Colletotrichum Leaf Spot (CLS), Bird's eyespot, Abnormal Leaf Fall (ALF), Algal spot.

1. Powdery Mildew(PM):-Fungus appears as white dusty colonies on leaf surfaces.
2. Corynespora Leaf fall(CLF):- Spots with railway track symptoms which infect young and old leaves.
3. Bird's eyes spot:- Numerous small circular spots scattered on leaf surfaces.
4. Abnormal leaf fall(ALF):- On leaves dull grey, circular spots appear which enlarge and become irregular.
5. Algal spot:- Small Translucent spots usually on the upper surface of leaves.

This project work involves the classification of healthy rubber leaves and two types of colletotrichum diseases, namely Traditional colletotrichum and New colletotrichum which is also known as Circular spot. Colletotrichum is one of the most severe diseases of Hevea brasiliensis. The fungal pathogens are responsible for Colletotrichum disease in rubber leaves.

- Traditional Colletotrichum:
  It mainly infects tender and immature leaves , as initially seen by many brown conical or elevated spots. As the disease progresses, the infected leaves become visibly wrinkled and twisted, eventually falling off the petiole.[7]

Figure 1.1: Traditional Colletotrichum

- New Colletotrichum(Circular spot):
  Appear as concentric rings that occur generally along the leaf margins, and occasionally in the middle of a mature leaf. These big lesions may combine to form recognizable larger patches of varying sizes. The central portion of this spot is light brown and papery.[7]

Figure 1.2: New Colletotrichum (Circular spot)

This project mainly focuses on a deep learning based framework to recognize the rubber plant diseases by investigating the rubber leaf images. Here a GAN model is used to generate synthetic images and augment these images to the dataset for training purpose. Then a pre-trained Convolutional Network using transfer learning is used for the detection of disease from leaf images. Transfer learning is the reuse of a previously learned model on a new problem which uses the knowledge learned from a prior assignment to increase prediction about a new task.

1.1 Proposed Project

This project presents a fully automated deep learning based framework to recognize the rubber leaf diseases by investigating the leaf images.

1.1.1 Problem Statement

Automatic detection and classification of diseases in rubber leaves using deep learning based approach.

1.1.2 Proposed Solution

To develop a Deep learning based approach for disease detection in rubber leaves using the Deep Convolutional Generative Adversarial Network (DCGAN) to generate synthetic images of rubber plant leaves. Thereafter, a DenseNet121 convolutional neural network model using transfer learning is used to classify the diseases in rubber leaves.

Chapter 2

Report of Preparatory Work

Plant disease diagnosis is important for crop management and production. Manual monitoring of plant disease is a difficult task. It requires tremendous amount of time, work and expertise. Therefore, fast and accurate plant disease detection methods are important. To effectively automate the detection of plant diseases , recently the deep-learning-based detection has been proposed and used successfully.

2.1 Literature Survey Report

1. Abbas, Amreen, Sweta Jain, Mahesh Gour, and Swetha Vankudothu. **Tomato plant disease detection using transfer learning with C-GAN synthetic images.** Computers and Electronics in Agriculture vol.187, p.106279, 2021 [1]

   Here,a deep learning-based method is used for tomato disease detection. The proposed model is trained and tested using the publicly available tomato PlantVillage dataset. The dataset consists of 16,012 images of tomato plant leaves which includes nine categories of infected leaf diseases, and one category is for healthy leaves. A Conditional Generative Adversarial Network (C-GAN) is utilized to generate synthetic images of tomato plant leaves. Thereafter, transfer learning with the DenseNet121 model is trained on synthetic and real images to classify the tomato leaves images into ten categories of diseases. This method helps farmers in the classification of diseases that affect tomato plants by simply taking an image of diseased leaves. Conditional Generative Adversarial Network (C-GAN) is used as a data augmentation strategy to increase the dataset size and prevent the network from overfitting. The generator model and discriminator model are two adversarial networks in the C-GAN model. It consists of an input layer, an embedding layer, and a dense layer followed by an input layer, reshape layer, and concatenate layer, which is in turn followed by four convolutional layers. There are 771,454 trainable parameters in total in the model. To train C-GAN, the discriminator model is given a real tomato image with a corresponding label, whereas the generator model is given the noise and a label. The generator model creates a fictitious representation of tomato leaves. The discriminator model is then fed with the generated fake image. The discriminator tries to tell the difference between actual and false images. Synthetic images are created when all of the C-GAN training has been completed. Then, for disease classification, a pre-trained DenseNet121 model was fine-tuned on tomato leaf pictures.

2. Zhang, Xihai and Qiao, Yue and Meng, Fanfeng and Fan, Chengguo and Zhang, Mingming. **Identification of maize leaf diseases using improved deep convolutional neural networks** , IEEE Access,vol. 6, pp.30370-30377, 2018. [2]

   In this study the diseases in maize leaves are identified using an improved deep convolutional neural network. It identified 8 categories of diseases in maize leaves namely Curvularia leaf spot, dwarf mosaic, gray leaf spot, northern leaf blight, brown spot, round spot, rust, and southern leaf blight, and a category representing healthy leaves. The proposed model used an improved GoogLeNet and Cifar10 models based on deep learning for leaf disease recognition. The input images were augmented to increase the size of the dataset and image preprocessing and labeling are done before the model is trained. Image normalization is used to format the size of the input and all the images are labeled with the appropriate disease acronym. GoogLeNet and Cifar10 network models are presented to increase the recognition accuracy of maize leaf diseases and improve the traditional identification methods with long convergence times and large numbers of model parameters. GoogLeNet makes use of sparse network architecture to mitigate the drawbacks of over-fitting and over-utilization of computational resources. It makes use of the pyramid model that expands the width and introduces the notion of "Inception Module". The goal of this module is to approximate the best solution using dense components. There are nine inception modules in total. The Cifar10 structure contains three convolutional layers, two fully linked layers, and a loss layer. A pooling layer and a Relu operation are applied after each convolutional layer in the model. The inception module of the GooLeNet model has been upgraded, while the Cifar10 model has been modified by adding dropout and ReLU between the two fully connected layers to improve the identification accuracy. The modified Cifar10 and GoogLeNet model is used to train the maize leaf image dataset. The revised approaches may have enhanced maize leaf disease accuracy and reduced convergence iterations, allowing for more efficient model training and recognition.

3. Ma, Juncheng and Du, Keming and Zheng, Feixiang and Zhang, Lingxian and Gong, Zhihong and Sun, Zhongfu. **A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network** , Comput. Electron. Agricult., vol.154, p. 18-24, 2018. [3]

   Here, a deep convolutional neural network (DCNN) was developed to undertake symptom-wise recognition of four cucumber diseases, namely anthracnose, downy mildew, powdery mildew, and target leaf spots. Cucumber leaf pictures obtained in the field were segmented to create the symptom images. A disease symptom segmentation method that integrated a comprehensive color feature with region growth was used to segment symptom images. Excess Red Index (ExR), H component of HSV color space, and b* component of L*a*b* color space make up the complete color feature (CCF), which performs powerful distinction of illness patches and clutter background. Then, to achieve robust and quick symptom image segmentation, an interactive region expanding method based on the color feature was applied. Data augmentation methods were used to extend the datasets created by the segmented symptom images to reduce the probability of overfitting. The DCNN performed well with the updated datasets containing about 14,208 symptom images. This model can be enhanced with additional properties such as those required for hyperspectral and thermal imaging.

4. Xiaoxiao, SUN and Shaomin, MU and Yongyu, XU and Zhihao, CAO and Tingting, SU. **Image recognition of tea leaf diseases based on convolutional neural network** 2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC). IEEE,p. 304-309 2018. [4]

   To identify tea leaf diseases, a CNN (Convolutional neural network) model is employed. Before being submitted to the network for training, the images of the diseased tea leaves were first preprocessed using image segmentation and enhancement methods. The edges are retrieved using the Soble edge extraction method, and certain interfering elements are removed using Gaussian image smoothing. Cutting a high-definition image of tea leaf disease can reduce background interference information in the image, enhancing the target area's visibility and simplifying the neural network's feature extraction process. The quantity and training quality of the dataset can both be increased by data augmentation.The learning rate and iteration numbers were continuously adjusted, and the dropout layer was correctly incorporated in the case of over-fitting in order to increase CNN's recognition accuracy. The convolution layer employs convolution filtering to go further into each little neural network building piece in order to extract more valuable, abstract characteristics. The Classification module, which uses a classifier, recognizes and categorizes the retrieved features. A classifier typically uses one or two layers of fully connected neural networks to do this. The ReLU linear function is employed as the activation function of neurons to improve network convergence. With the help of these CNN model improvements, this model achieves a high level of recognition accuracy.

5. Karthik, R and Hariharan, M and Anand, Sundar and Mathikshara, Priyanka and Johnson, Annie and Menaka, R **Attention embedded residual CNN for disease detection in tomato leaves**, Applied Soft Computing, vol.86, p.105933, 2020. [5]

   This work proposes two different deep architectures for detecting the infection in tomato leaves. The first architecture applies residual learning to learn significant features for classification. The second architecture applies attention mechanism on top of the residual deep network. The learning pattern of a CNN is generally based on aggregation of feature maps derived at multiple levels. It helps to selectively weigh the features different layers at the inception of a single layer. classification. The architecture of the proposed residual connection based on CNN consists of a sequence of three Residual Progressive Feature Extraction (RPFE) blocks, each set to learn progressive features. The attention model works on top of the RPFE CNN by retaining the context relevant features.

6. Kutty, Suhaili Beeran, Noor Ezan Abdullah, Hadzli Hashim, Aida Sulinda Kusim, Tuan Norjihan Tuan Yaakub, Puteri Nor Ashikin Megat Yunus, and Mohd Fauzi Abd Rahman. **Classification of Watermelon Leaf Diseases Using Neural Network Analysis** In 2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC), pp. 459-464. IEEE, 2013. [6]

   In this method, used a process to classify watermelon leaf diseases using neural network analysis. Here two types of watermelon leaf diseases such as Downey Mildew and Anthracnose have been tested in order to produce an automated model for detecting watermelon leaf diseases. A few of infected leaf samples were collected and they were captured using a digital camera with specific calibration procedure under controlled environment. The classification on the watermelon's leaf diseases is based on color feature extraction from RGB color model where the RGB pixel color indices have been extracted from the identified Regions of Interest (ROI). This work develops a classification system for watermelon leaf diseases using Neural Network Pattern Recognition Toolbox.

2.2 System Study Report

Rubber plant (Hevea brasiliensis) is one of the most important economic crop in the tropics. India is known to be the third topmost country in rubber production in the globe, of that Kerala ranks on the top for being the major contributor in rubber production . Every year, a large amount of latex are produced due to the increasing of rubber-based products in the market. A variety of diseases can affect the growth of rubber trees, which reduces the production and quality of natural rubber. Thus, a good care should be given to rubber trees in order to prevent the diseases and to get better yield . Deep learning is widely used in the field of agriculture, for plant disease detection and classification. The project aims to detect diseases in rubber leaves using deep learning-based method that utilizes the Deep Covolutional Generative Adversarial Network (DCGAN) and a pre-trained DenseNet121 model. DCGAN is used to generate synthetic images and DenseNet121 model is trained on synthetic and real images using transfer learning to classify the diseases.

Chapter 3

Project Design

In this method a deep learning network is applied to identify rubber leaf diseases. The proposed approach can be mainly divided into two parts; in the first part, synthetic images are generated using DCGAN for data augmentation. In the second part, a pre-trained DenseNet121 model is used for disease classification.

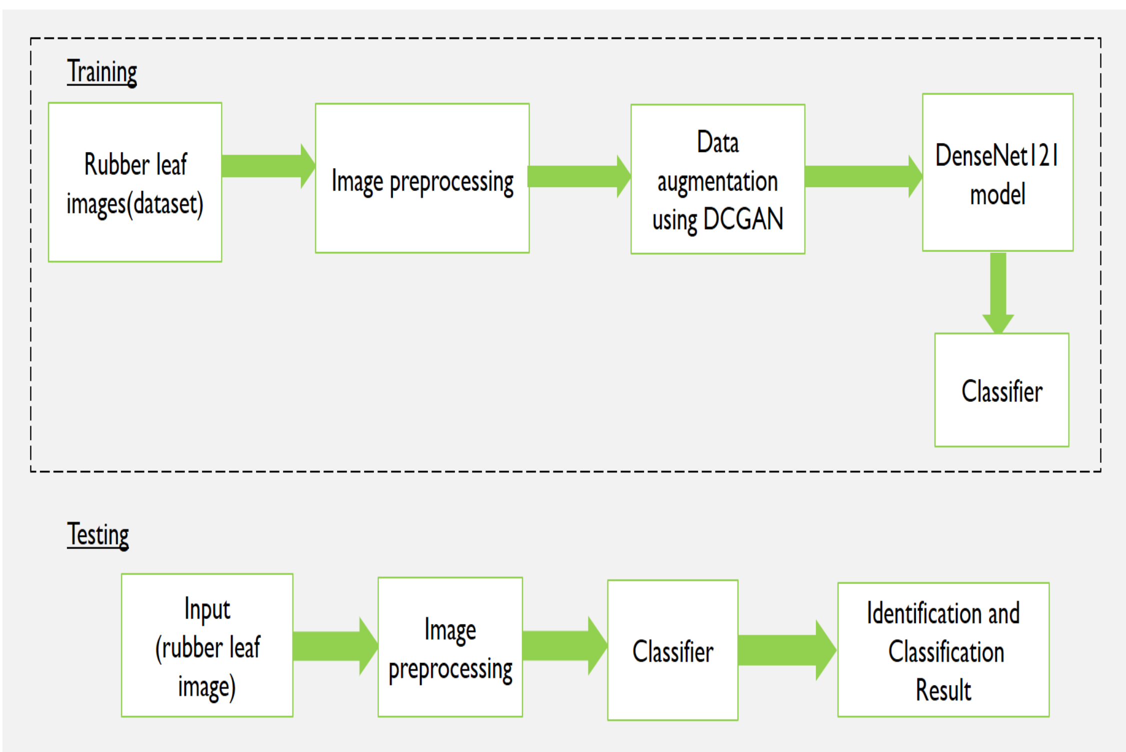

Figure 3.1: Block diagram of proposed method

3.1 Dataset

The dataset contains a total of 474 rubber leaf images which includes Traditional Colletotrichum, New colletrotrichum, and healthy leaves. The rubber leaf images were collected from four rubber plantations using android phones(OPPOA31,RealmeX2). For training the model, the dataset is divided into two sets i.e, Training set and Validation set. From Rubber Research Institute of India Kottayam, a total of 312 rubber leaves were collected which includes 104 leaf images of each category and these images are used in the training set. The remaining leaf images where collected from nearby rubber plantations(Malakkara,Kadammanitta and Chengannur). About 162 rubber leaf images were captured containing 54 images in each category . For validation, 132 images are used and 30 images are used for testing.

3.2 Image Pre-processing

In order to enhance the quality of input data and to extract meaningful information from the data several pre-processing techniques are performed on the acquired image. Image pre-processing are the steps taken to format the images before they are used by the model for training and inference. The aim of pre-processing is to improve the quality of the image so that it can be analysed in a better way. Pre-processing is required to clean image data for model input. It may also decrease model training time and increase model inference speed. Image sharpening is one of the pre-processing technique applied to the input image. Image sharpening is an effect applied to digital images to give them a sharper appearance. Sharpening enhances the definition of edges in an image and make the transitioning of features and details more significant. It uses a kernel which causes the pixel intensities to be higher and therefore more prominent to the human eye.
Unsharp filtering is an image sharpening technique that can be applied to the input image. It is also called unsharp mask filter which can sharpen an image or perform edge enhancement using a smoothing filter. The sharpening process works by utilizing a slightly blurred version of the original image. This is then subtracted away from the original image to detect the presence of edges, creating the unsharp mask. Unsharp Mask m(x , y) can be represented as:

$$m(x, y) = f(x, y) \text{ - } fb(x, y)$$

where, f(x,y) represents the original image and fb(x,y) represents blurred image. Then, this mask is added back to the original image resulting in enhanced high frequency components.

$$g(x, y) = f(x, y) + k \times m(x, y)$$

Here, k specifies weightage given to the mask being added. The above two equations can be combined into one as weighted average of original and blurred image.

$$\text{ie, } g(x, y) = (k + 1) \times f(x, y) \text{ - } k \times fb(x, y)$$

3.3 Data Augmentation using DCGAN

3.3.1 Generative Adversarial Network(GAN)

A Generative Adversarial Network (GAN) is a machine learning (ML) model developed in 2014. GANs are effective at generating high-quality synthetic images. It can be used for data augmentation technique. The performance of CNN models heavily depends on the available dataset for training. Insufficient amount of dataset results in overfitting. The main purpose of data augmentation is to enhance the size of the dataset. It generates a new set of data based on training data that look like training data. Two major components of GANs are Generator and Discriminator.

- Generator- It will generate data that is fake based on original(real) data. Its aim is to generate the fake image based on feedback and make the discriminator fool that it cannot predict a fake image.

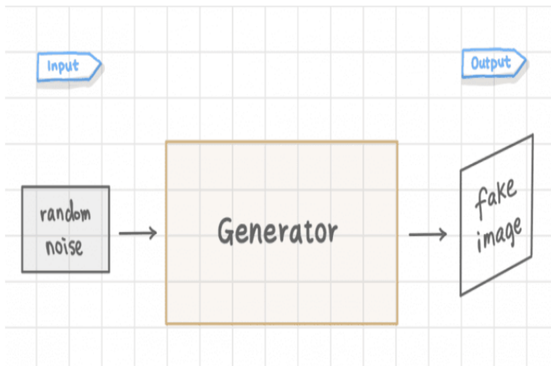

Figure 3.2: Generator model

- Discriminator- It is a simple classifier that predicts data is fake or real. It is trained on real data and provides feedback to a generator

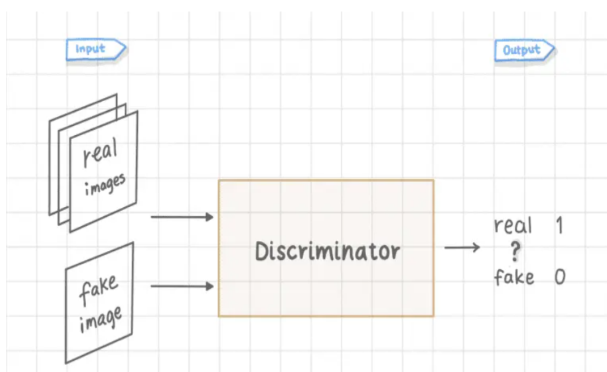

Figure 3.3: Discriminator model

The generator takes in random noise (random numbers) and generates an image. This generated image is fed then into the discriminator alongside a stream of images taken from the actual data. The discriminator takes in both real and fake images and returns probabilities, a number between 0 and 1, with 1 representing a real image and 0 representing fake. The objective function V (D,G) of GAN is as follows:

$$\min_G \max_D E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_{generated}(z)}[1 - \log D(G(z))]$$

where x is a real sample, D(x) represents the probability of discriminating x as a real sample by discriminator networks D, G(z) is a sample generated from noise z by the generator network G, and D (G(z)) indicates the probability of discriminating G(z) as a real sample by discriminator network D. the Generator wants to minimize the V(D, G) whereas the Discriminator wants to maximize the V(D, G). The Discriminator wants to maximize the loss function V(D, G) by correctly classifying real and fake images. The Generator wants to minimize the loss function V(D, G) by generating images that look like real images and tries to fool the Discriminator.

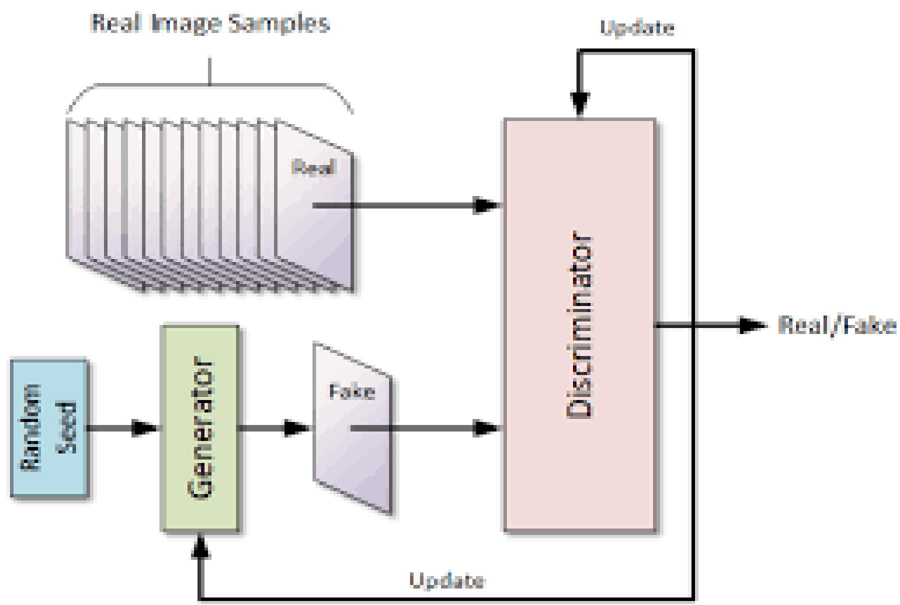

Figure 3.4: Architecture of GAN

3.3.2 Deep Convolutional Generative Adversarial Network(DCGAN)

Deep convolutional Generative Adversarial Network (DCGAN) is considered as the application of GAN extended to the field of CNN. It is used as a data augmentation technique to enhance the size of dataset in order to prevent the network from over-fitting. DCGAN was proposed in 2015. DCGAN made some changes to the structure of convolutional neural network to improve the quality of samples and the speed of convergence. Here, all pooling layers are replaced by strided convolutions (discriminator) and fractional-strided convolutions (generator). A strided convolution can decrease the dimension by jumping multiple pixels between convolutions instead of sliding the kernel one-by-one. Similarly, it can increase dimension by adding empty pixels between the real ones, called fractional-strided convolution. Batch normalization was used on the generator and discriminator networks. Batch Normalization stabilizes learning by normalizing the input. Batchnorm standardizes the input layer to have a zero mean and unit variance. It is typically added after the hidden layer and before the activation layer. This helps to deal with training problems that arise due to poor initialization and helps gradient flow in deeper models. The discriminator network is a convolutional neural network with the whole connection layer removed. In addition to using tanh as the activation function on the output layer, the ReLU activation function is used on the other layers of the generator network. All activation functions using LeakyReLU is used as a binary problem in discriminator network.[9]

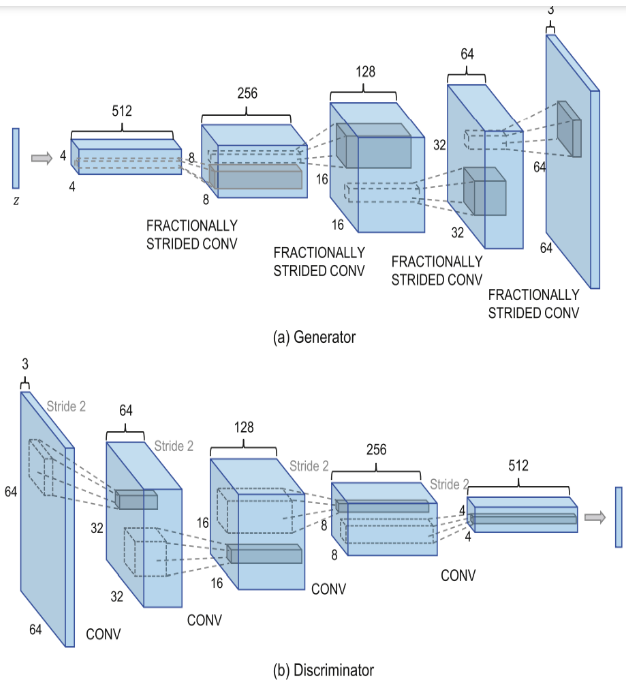

Figure 3.5: Architecture of DCGAN

3.4 DenseNet121 classification Model

In this method a pre-trained DenseNet121 convolutional neural network model using transfer learning is used for rubber leaf disease detection. DenseNet is a convolutional neural network where each layer is connected to all other layers that are deeper in the network, that is, the first layer is connected to the 2nd, 3rd, 4th and so on, the second layer is connected to the 3rd, 4th, 5th and so on. This is done to enable maximum information flow between layers of network. In DenseNet, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers i.e, concatenation is used. Each layer is receiving a “collective knowledge” from all preceding layers. Since each layer receives feature maps from all preceding layers, network can be thinner and compact, i.e. number of channels can be fewer. For 'L' layers, there are L(L+1)/2 direct connections. It hence requires fewer parameters than traditional convolutional neural networks, as there is no need to learn unimportant feature maps. The use of the concatenation operation is not feasible when the size of feature maps changes. However, an essential part of CNNs is the down-sampling of layers which reduces the size of feature-maps through dimensionality reduction to gain higher computation speeds. To enable this, DenseNets are divided into DenseBlocks, where the dimensions of the feature maps remains constant within a block, but the number of filters between them is changed. The layers between the blocks are called Transition Layers which reduce the the number of channels to half of that of the existing channels. DenseNet starts with a basic convolution and pooling layer. Then there is a dense block followed by a transition layer, another dense block followed by a transition layer, another dense block followed by a transition layer, and finally a dense block followed by a classification layer. Unlike existing network architectures, DenseNets can have very narrow layers. Although each layer only produces k output feature-maps, the number of inputs can be quite high, especially for further layers. Thus, a 1x1 convolution layer can be introduced as a bottleneck layer before each 3x3 convolution to improve the efficiency and speed of computations.

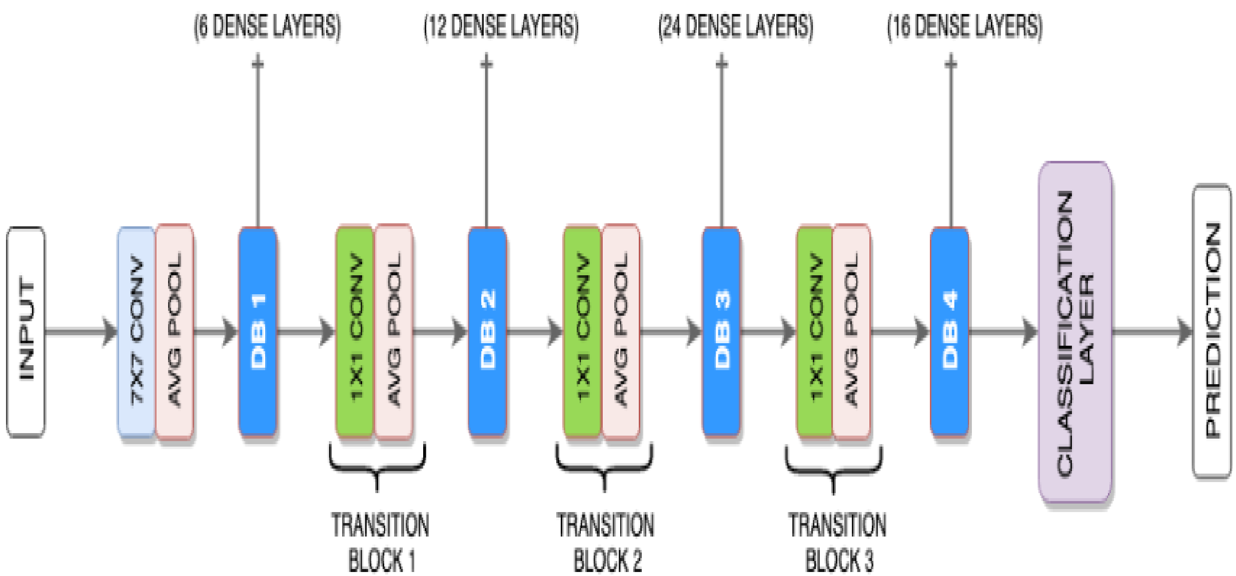

Figure 3.6: Architecture of DensNet121

The DensNet121 model consists of 4 dense blocks that take 224×224 pixels image as input. The first convolution layer consists of 2000 convolutions of size 7×7 with stride 2, which is followed by a 3×3 max pooling layer with stride 2. The pooling layer is followed by three dense blocks with each dense block being followed by a transition layer. The fourth dense block is followed by a classification layer. The dense blocks consist of batch normalization, ReLU and convolutional layers. The transition layer consists of 1×1 convolution layer followed by 2×2 average pooling layer with stride 2.

Layer Name	Output size	Parameter/Filter size
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense block (1)	56×56	[1×1 conv, 3×3 conv $\times$ 6]
Transition layer (1)	56×56 28×28	1×1 conv 2×2 average pool, stride 2
Dense block (2)	28×28	[1×1 conv, 3×3 conv $\times$ 12]
Transition layer (2)	28×28 14×14	1×1 conv 2×2 average pool, stride 2
Dense block (3)	14×14	[1×1 conv, 3×3 conv $\times$ 24]
Transition layer (3)	14×14 7×7	1×1 conv 2×2 average pool, stride 2
Dense block (4)	7×7	[1×1 conv, 3×3 conv $\times$ 16]
Additional convolution layer	7×7	[3×3 conv, 3×3 conv]
Classification layer	1×1	7×7 global average pool 10D fully connected, softmax

Figure 3.7: Summary of DensNet121

3.5 Hardware & Software Requirements

- Operating System : 64-bit Operating System
- Processor : Intel Core i5 9th Gen 2.40GHz
- Graphics card: 6GB NVIDIA GeForce GTX 1650 GPU
- RAM : 8GB
- Supporting Environment : Google Colab
- Supporting Software and Libraries : Python, Tensorflow, Keras, OpenCv.

Chapter 4

Implementation

The proposed model is implemented in Tensorflow framework using Google colab environment. The implementation part involves mainly three stages, a pre-processing stage, data augmentation stage using DCGAN and classification stage using DenseNet121 model. The model is trained with self made dataset of rubber leaves.

4.1 Image Pre-processing using sharpening operation

In image pre-processing stage the collected rubber leaf image dataset which contains two categories diseases and a category of healthy leaves are pre-processed using sharpening operation. The Sharpening operation is performed using unsharp filter. The unsharp mask filter sharpens an image by perform edge enhancement using a smoothing filter. In this sharpening method, first the image is blurred(smoothed) using Gaussian filter of mask 7×7.

Figure 4.1: Input image

Figure 4.2: Blurred image

This blurred image is then subtracted away from the original image which detect the presence of edges, creating the unsharp mask. Then, this mask is added back to the original image to generate sharpened image. Sharpened image is generated using this equation,
$g(x, y) = (k + 1) \times f(x, y)$ - $k \times fb(x, y)$. The value of k=2.

Figure 4.3: Sharpened image

4.2 Data Augmentation

For data augmentation Deep Convolutional Generative Adversarial Network model (DCGAN) is used. DCGAN model is trained with a batch size 16 and image size of 64×64. In DCGAN, the generator network takes random noise as an input and outputs an synthetic images that resembles the training images. The discriminator is a simple binary classifier that tells whether an image is real or fake. Its objective is trying to classify the images correctly. The DCGAN model is trained using 104 images of each class about 2000 epochs to create 60 synthetic images. Since, GAN models requires large amount of data for training, therefore with the available image dataset the model can generate only small amount of augmented images. To make a proper dataset for training the model, about 34 images in addition are created using traditional augmentation method. Therefore, training dataset contains a total of 86 augmented images in each class. For validation a total of 132 images are used which includes 24 GAN generated images.

Figure 4.4: Generated images using DCGAN

4.3 Classification using DenseNet121

DenseNet121 model is implemented using training and validation datasets. The model is trained using 600 images containing 200 images in 3 classes with a batch size of 128 and image size of 224×224. During training, the model is validated using validation dataset containing a total of 150 images with 50 images in 3 classes. For training, the model uses Adam optimizer with learning rate 0.001. The model is trained in 100 epochs. The DenseNet121 model generates the classifier.

Chapter 5

Results & Conclusions

5.1 Performance Analysis of the model

The performance of the model is evaluated using the Confusion matrix. A Confusion matrix is a tabular summary of the number of correct and incorrect predictions made by a classifier. True positive (TP), True negative (TN), False positive (FP), and False negative (FN) are various parameters related to the confusion matrix (FN). With these parameters, it can be used to calculate performance metrics like accuracy, precision, recall, and F1-score in order to assess the effectiveness of a classification model. The model is tested using a total of 30 images containing 10 images in each class.

```
Found 30 images belonging to 3 classes.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:23: UserWarning: `Model.predict_generator` is deprecated and will be removed
Confusion Matrix
[[10  0  0]
 [ 1  8  1]
 [ 2  0  8]]
Classification Report
              precision    recall  f1-score   support

    Circular       0.77      1.00      0.87        10
     Healthy       1.00      0.80      0.89        10
 Traditional       0.89      0.80      0.84        10

    accuracy                           0.87        30
   macro avg       0.89      0.87      0.87        30
weighted avg       0.89      0.87      0.87        30
```

Figure 5.1: Performance Analysis

Some of the results of testing the model with rubber leaf images are shown below:

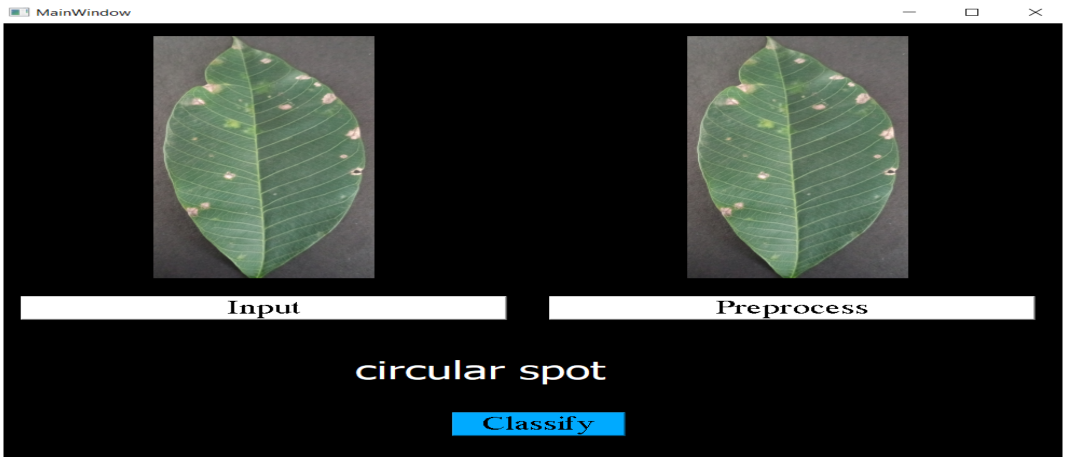

Figure 5.2: Test1

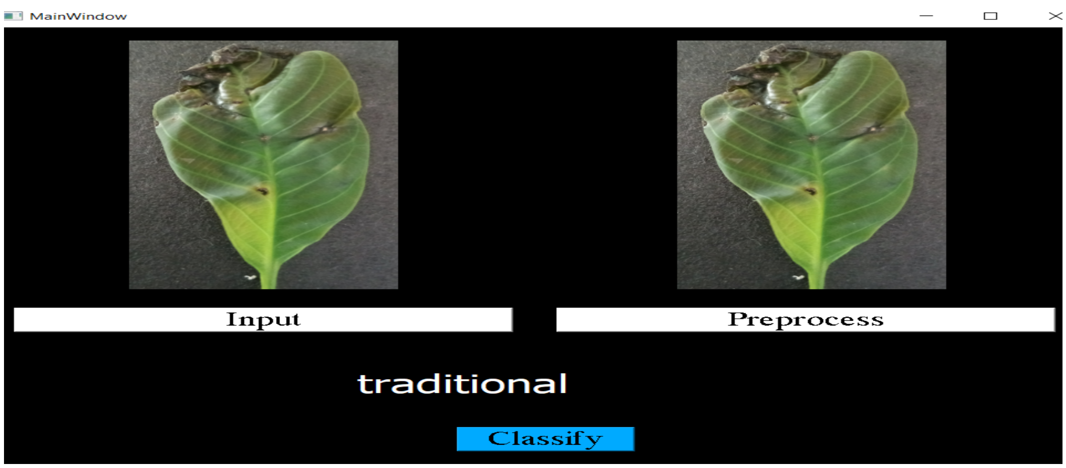

Figure 5.3: Test2

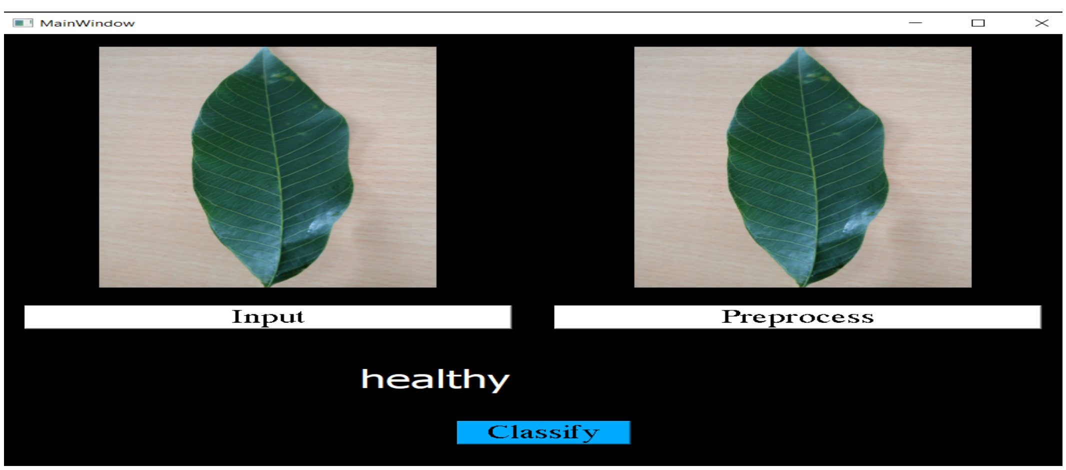

Figure 5.4: Test3

5.2 Conclusion

In this project a Deep learning based model for automatic detection and classification of diseases in rubber leaves is developed. Here, GAN (Generative Adversarial Networks) based model with transfer learning using DenseNet121 CNN is used for the detection and classification of diseases. For data augmentation, a DCGAN(Deep Convolutional Generative Adversarial Network) deep learning model is used. The images generated by DCGAN enlarges the size of the data set, and it also has the characteristics of diversity which makes the model with a good generalization effect. Then the DenseNet121 convolutional neural network model is used for classification which classifies two categories of diseases, Traditional Colletotrichum and New Colletotrichum and a category of healthy rubber leaves. The model achieved an accuracy of 87% after testing.

5.3 Future scope

In future the proposed model can be improved by adding a different CNN classification model for differentiating the rubber leaves from other leaves. In addition, the rubber also have other disease characteristics, that may affect stem which can be specified using this model.

References

[1] Amreen Abbas, Sweta Jain, Mahesh Gour, and SwethaVankudothu. Tomato plant disease detection using trans-fer learning with c-gan synthetic images.Computers andElectronics in Agriculture, 187:106279, June 2021.

[2] Xihai Zhang, Yue Qiao, Fanfeng Meng, Chengguo Fan,and Mingming Zhang. Identification of maize leaf diseasesusing improved deep convolutional neural networks.IeeeAccess, 6:30370–30377, June 2018.

[3] Juncheng Ma, Keming Du, Feixiang Zheng, LingxianZhang, Zhihong Gong, and Zhongfu Sun. A recognitionmethod for cucumber diseases using leaf symptom imagesbased on deep convolutional neural network.Computersand electronics in agriculture, 154:18–24, Novmber 2018.

[4] SUN Xiaoxiao, MU Shaomin, XU Yongyu, CAO Zhihao,and SU Tingting. Image recognition of tea leaf diseasesbased on convolutional neural network. In2018 Inter-national Conference on Security, Pattern Analysis, andCybernetics (SPAC), pages 304–309. IEEE, 2018.

[5] R Karthik, M Hariharan, Sundar Anand, PriyankaMathikshara, Annie Johnson, and R Menaka. Attentionembedded residual cnn for disease detection in tomatoleaves.Applied Soft Computing, 86:105933, 2020.

[6] Suhaili Beeran Kutty, Noor Ezan Abdullah, HadzliHashim, Aida Sulinda Kusim, Tuan Norjihan TuanYaakub, Puteri Nor Ashikin Megat Yunus, Mohd FauziAbd Rahman, et al. Classification of watermelon leaf dis-eases using neural network analysis. In2013 IEEE Busi-ness Engineering and Industrial Applications Colloquium(BEIAC), pages 459–464. IEEE, 2013.

[7] Liu X, Li B, Yang Y, Cai J, Shi T, Zheng X, Huang G. Pathogenic adaptations revealed by comparative genome analyses of two Colletotrichum spp., the causal agent of anthracnose in rubber tree. Frontiers in microbiology. 2020 Jul 21;11:1484.

[8] Mazlan, Safwan, Noraini Md Jafar, Aswad Wahab, Heraa Rajandas, and Dzarifah Zulperi. "Major diseases of rubber (hevea brasiliensis) in malaysia." Pertanika Journal of Scholarly Research Reviews 5, no. 2 , 2019.

[9] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434,2015.

[10] http://prri.da.gov.ph/wp-content/uploads/2019/07/Common-Diseases-of-Rubber-and-their-Management.pdf

[11] https://www.agrigoaexpert.res.in/icar/category/horitculture/plantation_crops/rubber.php