

PROFESSIONAL TRAINING REPORT
At
Sathyabama Institute of Science and
Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Technology Degree in
Information Technology

By
DHANESHWAR.V
(Reg. No.39120022)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

NOVEMBER 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFAIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **Dhaneshwar. V (Reg. No: 39120022)** who carried out the project entitled “**Cotton Disease Prediction using Machine Learning**” under my supervision from June 2021 to November 2021.

Internal Guide

Dr. V. Maria Anu, M.E., Ph.D.

Head of the Department

Dr. R. Subashini, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Dhaneshwar.V** hereby declare that the project report entitled **Cotton Disease Prediction using Machine Learning** done by me under the guidance of **Dr. V. Maria Anu, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE:



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D. Dean**, School of Computing, **Dr. Dr. R. Subhashini, M.E., Ph.D. Heads of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. V. Maria Anu, M.E., Ph.D.** for his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

NPTEL CERTIFICATE:

This certificate is computer generated and can be verified by scanning the QR code given below. This will display the certificate from the NPTEL repository, <https://npTEL.ac.in/noc/>

Roll No: NPTEL21CS62S24481041

To
DHANESHWAR V
8A, GANDHI NAGAR, ANNA STREET, K.K. NAGAR
TAMIL NADU - 620021
PH. NO : 8870327336



Score	Type of Certificate
>=90	Elite+Gold
75-89	Elite+Silver
>=60	Elite
40-59	Successfully Completed
<40	No Certificate

No. of credits recommended by NPTEL: 2
An additional 1 credit may be awarded if the University deems it fit, based on the actual student effort involved.



Elite

NPTEL Online Certification
(Funded by the Ministry of HRD, Govt. of India)



This certificate is awarded to
DHANESHWAR V
for successfully completing the course
Cloud Computing
with a consolidated score of **62** %

Online Assignments	14.83/25	Proctored Exam	46.67/75
--------------------	----------	----------------	----------

Total number of candidates certified in this course: 3189


Prof. G P Raja Sekhar
Dean, Continuing Education
IIT Kharagpur

Aug-Oct 2021
(8 week course)

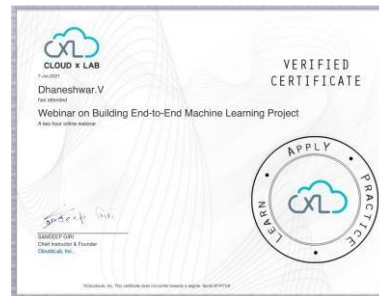

Prof. Debjani Chakraborty
Coordinator, NPTEL
IIT Kharagpur

 Indian Institute of Technology Kharagpur

Roll No: NPTEL21CS62S24481041 To validate and check scores: <https://npTEL.ac.in/noc/>

12

OTHER CERTIFICATES:



ABSTRACT

Cotton is one of the economically significant agricultural products in Ethiopia and India, but it is exposed to different constraints in the leaf area. Mostly, these constraints are identified as diseases and pests that are hard to detect with bare eyes. This study focused to develop a model to boost the detection of cotton leaf disease and pests using the deep learning technique, CNN. To do so, the researchers have used common cotton leaf disease and pests such as bacterial blight, spider mite, and leaf miner. K-fold cross-validation strategy was worn to dataset splitting and boosted generalization of the CNN model. For this research, nearly 2400 specimens (600 images in each class) were accessed for training purposes. This developed model is implemented using python version 3.7.3 and the model is equipped on the deep learning package called Keras, TensorFlow backed, and GCOLAB which are used as the developmental environment. This model achieved an accuracy of 96.4% for identifying classes of leaf disease and pests in cotton plants. This revealed the feasibility of its usage in real-time applications and the potential need for IT-based solutions to support traditional or manual disease and pest's identification

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTRACT LIST OF FIGURES LIST OF ABBREVIATION	7 10 11
1	INTRODUCTION 1.1 Brief Intro 1.3 Cotton Disease Prediction	12-13 12 13
2	AIM AND SCOPE 2.1 Aim 2.2 Problem Statement 2.3 Scope	13-16 13 14 15
3	SYSTEM ANALYSIS AND DESIGN 3.1 General 3.2 User Interface (UI) 3.3 Software's Used 3.4 System Requirement	15-16 15 15 15-16 16
4	PROJECT DISTRIBUTION 4.1 General 4.2 Modules Involved	17-18 17 17-18

5	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS 5.1. Results 5.2. Discussion 5.3. Performance Analysis	20-21 20 20 21
6	SUMMARY AND CONCLUSION 6.1. Summary 6.2. Conclusion	22 22 22
7	APPENDIX A. Screenshots B. Source Code	23-33 23-26 27-33

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
Fig.4.3.1	Workflow	19
Fig.4.3.2	CNN Model	19
Fig A.1	Performance Analysis	21
Fig A.2	Roadmap	23
Fig A.3	Python GUI	23
Fig A.4	Keras applications	24
Fig. A.5	Cotton disease Prediction Using datasets and inception Resnet50	24
Fig. A.6	Cotton disease Prediction Using datasets and inception v3 with keras	25
Fig A.7	Prediction graph	25
Fig A.8	Result Accuracy	26
Fig A.9	Result Accuracy	26

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
ML	Machine Learning
CNN	Convolution Neural network
NN & DL	Neural Networks & Deep Learning

INTRODUCTION

1.1. BRIEF INTRO:

Even though agriculture is the backbone of Ethiopia, so far, no advanced technologies have been explored in the development of automation in agricultural science and also there are high problems in production and quality due to different diseases and pests. In recent times, the sophisticated emerging technology has attracted many researchers in the field of detection and classification of cotton leaf diseases and pests. In Ethiopia, there are several constraints which reduce the yield and quality of the product. Particularly, identification of potential diseases or pests on Ethiopian cotton is based on traditional ways. 'Ere is a wide area of farm suitable for cotton plantation, but only limited research attention is given to cotton crop production. Traditionally, experts detect and identify such plant diseases and pests on bared eyes. Bared eye determination is considered as a loss of low-level accuracy in order to detect any diseases. On high demand, different advanced technologies were aided for structuring the systems to assist nonautomatic recognition of the plant diseases and pests to increase the accuracy for any corrective measures. With the help of advanced technologies, the plant diseases were reduced, thus increasing the productivity which helped to raise the economy via boosting the production. Among different diseases and pests occurred, about 80–90% were on the leaves of cotton. In Ethiopia, it is observed that there might be a fiscal destruction around 16% because of plant syndromes. However, without control measures, it can cause 30%–50% of loss. Cotton diseases and pests are difficult to identify through bared eyes.

1.2 COTTON DISEASE PREDICTION

Now a day's financial impact of agriculture is increase day by day with the economic growth of our country, still, agriculture is one wide sector and that plays a very important role for our county. While keeping track of diseases in plants with the help of specialists can be very costly in the agriculture region. There is a need for a

system that can automatically detect the diseases as it can bring revolution in monitoring large fields of crop and then plant leaves can be taken cure as soon as possible after detection of disease. There are so many diseases that affect cotton and many more crops that affect many filed of agriculture. So those identify this disease and how to recover from it. This objective will satisfy our application which helps for doing the prediction of cotton disease as well as how to overcome it. In this dataset, there are 4 categories like a diseased cotton leaf, diseased cotton plant, fresh cotton leaf, fresh cotton plant. In this project, I have created two transfer learning models like Xception and Resent101 and CNN.

CHAPTER 2

AIM AND SCOPE

2.1 AIM

My main Objective is to discover patterns in your data and then make predictions based on often complex patterns to answer business questions, detect and Analyse trends and help solve problems

2.2 PROBLEM STATEMENT

The cotton plant is susceptible to several disorder (biotic and abiotic constraints) attacks due to temperature fluctuation, diseases, and pests. Indeed, the whole world produced nearly 576kg per hectare of cotton crops, where only 10% of production loss occurred due to different cotton leaf diseases. The United States of America (USA) is a major exporter of cotton in the world and it obtained 5.1 billion US dollars in 2016, but there are well-known native pests which were the reason for the distraction of cotton farms. And, India has 24 percent of cotton land of the world and got 4.6 billions of dollars in 2016, from which generally 18% of cotton crops' production was lost every year due to different diseases that attacked the cotton plants which had its impacts on losing almost nine hundred thousand of Indian rupees. Presently, in Ethiopia, nearly

performance evaluation of GTP-I showed that these diseases and pests are the main constraints of the world standards in cotton quality and quantity of production. This results in the downfall of the economy of both the farmer and the country.

Detecting these diseases with bare eyes increased the complexity of cotton crops productivity which decreased the accuracy in identification precision. Even an expert would fail to assess and diagnose the diseases with their bare eyes, and this inadequate technique leads to more wastage of cotton crops. Due to these mistaken conclusions, most of the time, certain unnecessary pesticides which badly affect healthy cotton are applied. Leaving the farm for even a short time interval without production will affect overall nation GDP. To make an efficient and effective interface system, the human plays an important role. Graph convolutional neural networks, a novel deep learning framework, addressed the issues in order to differentiate the four-class motor imaginary intentions by mutually agreeing through the similarity of electroencephalography electrodes. To find the motor imaginary, four tasks are preferred with the prediction of highest accuracy

2.3 SCOPE

This research study focused on developing an identification model for cotton leaf diseases and pests using deep learning technique called convolutional neural networking. Three common types of disease and pests such as bacterial blight, leaf miner, and spider might have been affecting cotton productivity and quality. Also, the model applied made a supervised learning technique on datasets with four prime feature extraction process and 2400 datasets. The datasets are limited to four different feature descriptors. Taking into consideration the time constraints and reach of the regions that grow cotton, the research focused in the southern part of Ethiopia such as Arba Minch, Shele, and Woyto. MelkaWorer agricultural research center was also proposed as a focus area because it is responsible for cotton farms in SNNPR. Deep learning techniques were used to perform the automatic feature eradication from the different input datasets.

2.3.1 DISADVANTAGES OF CURRENT SYSTEM

- Hard to use the datasets Manually
- These types of problem only solved by digitalization
- Unable handle huge amount samples

2.3.2 ADVANTAGES OVER CURRENT SYSTEM

- Easy to handle and research
- Friendly interface.
- Fast and convenient.
- Easy to update.
- Secure
- Better Understanding

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 GENERAL

There is a need for a system that can automatically detect the diseases as it can bring revolution in monitoring large fields of crop and then plant leaves can be taken cure as soon as possible after detection of disease. There are so many diseases that affect cotton and many more crops that affect many filed of agriculture. So those identify this disease and how to recover from it. This objective will satisfy our application which helps for doing the prediction of cotton disease as well as how to overcome it.

3.2 USER INTERFACE (UI)

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse, and the appearance of a desktop. It is also the way through which a user interacts with an application or a website

3.3 SOFTWARES USED

3.3.1 CODE EDITORS USED

VISUAL STUDIO CODE

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and MacOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.

3.3.2 PROGRAMMING LANGUAGES USED

PYTHON 3

Python is the most used language for Machine Learning (which lives under the umbrella of AI). One of the main reasons Python is so popular within AI development is that it was created as a powerful data analysis tool and has always been popular within the field of big data.

PACKAGES:

- TensorFlow
- Pandas
- NumPy
- Keras
- GUI
- Utilis
- Matplotlib

3.4 SYSTEM REQUIREMENTS

Operating System: Windows / MacOS / Linux

RAM: 1GB (minimum)

Processor: 1GHz processor (minimum)

ROM: 20GB (minimum)

CHAPTER 4

PROJECT DESCRIPTION

4.1 GENERAL

For the Project I have download dataset from KAGGLE. In the Dataset in saved three folder Train, Test, Validation folder. In the dataset there are Four Categories like Diseased Cotton leaf, diseased Cotton plant, Fresh Cotton leaf, Fresh Cotton plant

Hosted Link: <https://github.com/dhaneshwar2310/projectML>

4.2 MODULES INVOLVED

To develop and manage a production-ready model, you must work through the following stages:

- Source and prepare your data.
- Develop your model.
- Train an ML model on your data:
 - Train model
 - Evaluate model accuracy
 - Tune hyperparameters
- Deploy your trained model.
- Send prediction requests to your model:
 - Online prediction
 - Batch prediction
- Monitor the predictions on an ongoing basis.
- Manage your models and model versions.
 -

These stages are iterative. You may need to reevaluate and go back to a previous step at any point in the process.

The rest of this page discusses the stages in detail.

Source and prepare your data

You must have access to a large set of training data that includes the attribute (called a *feature* in ML) that you want to be able to *infer* (predict) based on the other features.

4.3. CNN Models

4.3.1. VGG and ResNet

The VGG technique emerged in 2014 as a prominent deep CNN and has been widely used as the backbone framework in numerous feature recognition tasks [40–53]. Studies on network depth and performance of the VGG structure have indicated that its depth affects the model's performance to a certain extent. The VGG network structure is very regular; several convolutional layers are followed by a pooling layer that reduces the image's height and width. There is certain regularity in filter numbers in the convolutional layer, which doubles from 64 to 128 and then to 256 and 512. In this study, we use the VGG-19, which contains 19 convolutional and fully connected layers. The VGG problem initially inspired the ResNet structure: the problem of degradation with continuously increasing network depths. The ResNet is a modified version of VGG, provided with 50 or 101 layers in common. Using a residual block that transmits information from input to the output directly, the degradation problem is solved, although the numbered layers are largely increased. In practice, the residual block is a combination of 1×1 , 3×3 , and 1×1 convolutional layer. The middle 3×3 convolutional layers first reduces the calculation under a dimension reduction 1×1 convolutional layer. It then restores it under another 1×1 convolutional layer, which maintains the accuracy and reduces the calculation amount. The layers may differ for ResNet, and we use ResNet-101 as a backbone in this study.

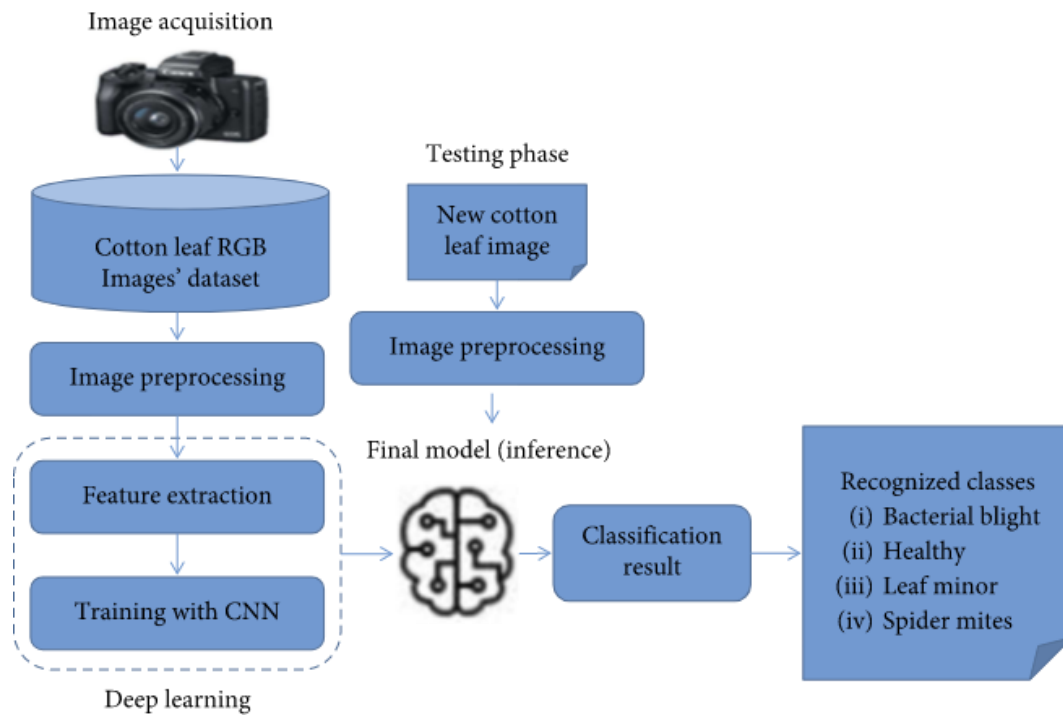


Fig.4.3.1. Workflow

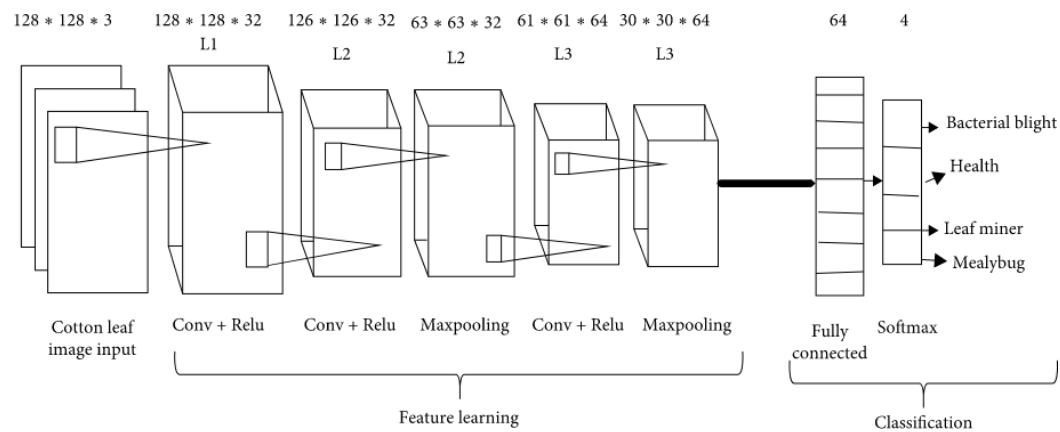


Fig.4.3.2. CNN Model

CHAPTER – 5.

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

5.1. RESULTS:

In Google Colab and Juypiter we can easily use datasets and the implementation of cotton disease prediction and we done keras application inceptionV2 and inceptionResNetV2.

We got the result of accuracy of 0.9745471 and 0.9745471

5.2.DISCUSSION:

To analyze the performance of the model, the last result is achieved using parameters such as K-fold cross-validation using 10 folds. RGB-colored image dataset with augmentation provides 15% best performance for the model. The researchers used the transferred learning CNN model and the grayscale dataset achieved 98.6% accuracy [6]. However, color is the main and most decisive feature in cotton detection and classification; therefore, using a colored dataset takes a long time to train the model to add performance even if it is a complex layer. The number of epochs with 100 iterations and the Adam optimization method is very significant to boost the model performance by 10% and 5.2%, respectively. In the end, this developed CNN model achieves 98% of bacterial blight, 94% healthy, 97.6% of leaf minor, and 100% of spider mite, which are correctly classified. Additionally, the researcher has used different preprocessing techniques for noise removal. The main factors for the misclassification of the result exist between bacterial blight, healthy, and leaf miner. The overall performance of the model, as shown in the confusion matrix, is 96.4% accurate for diagnosis of leaf disease and pests of cotton plants.

5.3 PERFORMANCE ANALYSIS:

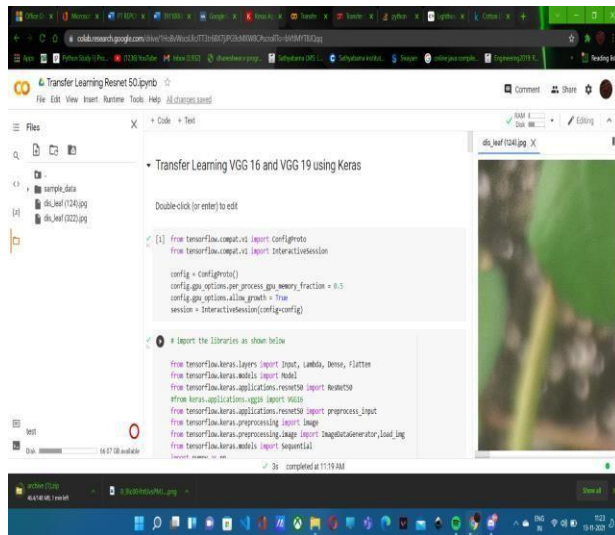


Fig.5.1. Performance Analysis

The overall performance of the cotton leaf disease and pest identification prototype evaluation selected by the evaluator was 80% extremely satisfied option for all questions and 20% of very satisfied.

CHAPTER-6

SUMMARY AND CONCLUSIONS

6.1 SUMMARY

The project I've done is useful for the photo restoration and cotton plant disease prediction and suitable pest prediction which is useful for our society

6.2 CONCLUSION

Machine learning is **quickly growing field in computer science**. It has applications in nearly every other field of study and is already being implemented commercially because machine learning can solve problems too difficult or time consuming for humans to solve.

REFERENCES:

- [1] <https://developer.mozilla.org/en-US/docs/Learn>
- [2] <https://www.w3schools.com/whatis/>

APPENDIX

A. SCREENSHOTS:

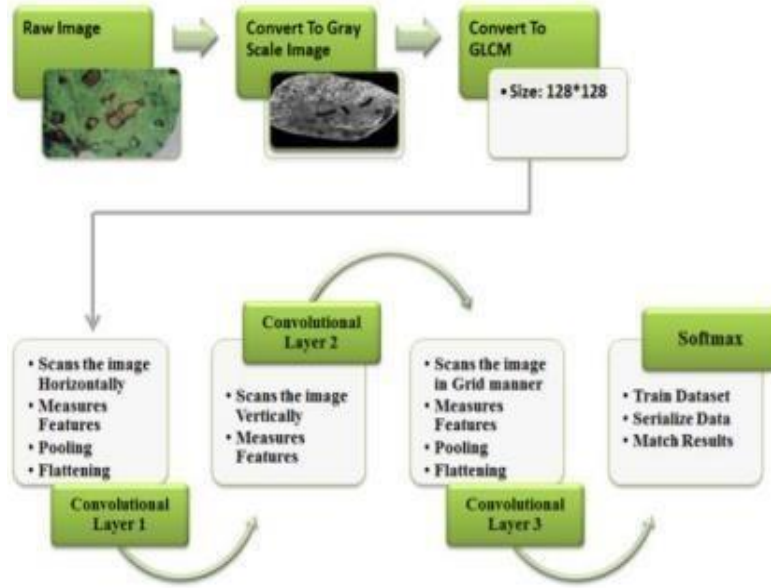


Fig A.1. Roadmap

Backbone Network	AP_{box}	AP_{50}	AP_{75}
<i>ResNet50</i>	41.5	60.1	45.5
<i>ResNet101</i>	43.3	61.7	47.2
<i>ResNeXt101</i>	45.9	64.4	50.2
<i>ResNeXt152</i>	48.3	67.0	52.8
<i>Dual-ResNeXt152 (ours)</i>	50.0	68.8	54.6
<i>Triple-ResNeXt152 (ours)</i>	50.7	69.8	55.5

Fig A.2. keras applications

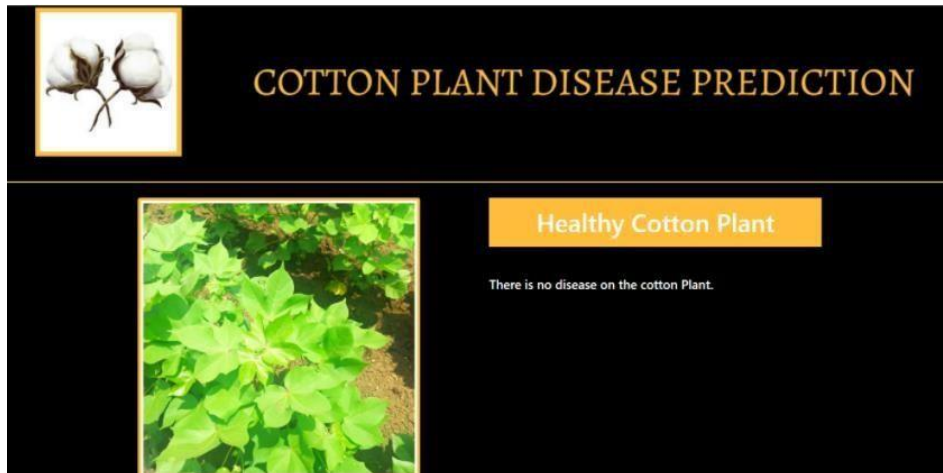


Fig A.3. Python GUI

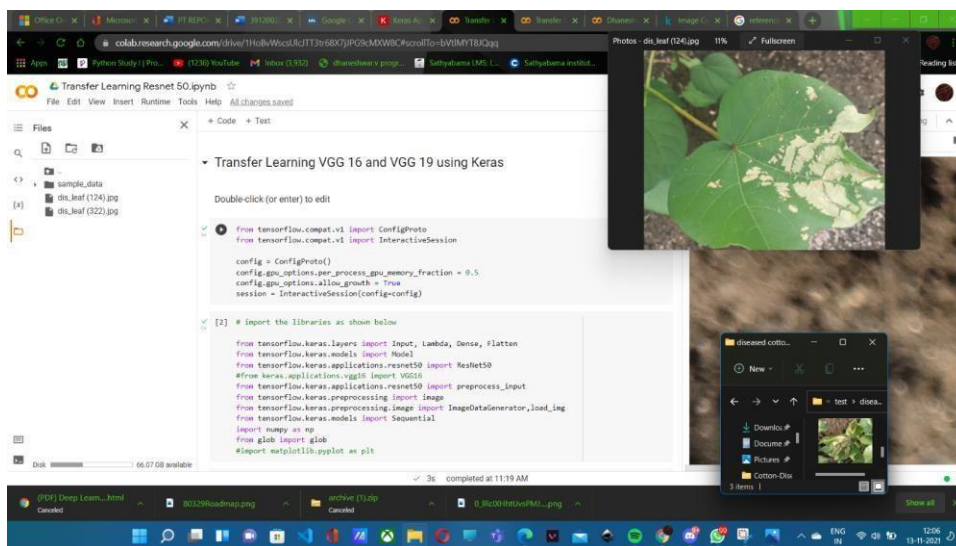


Fig A.4. Cotton disease Prediction Using datasets and inception Resnet50

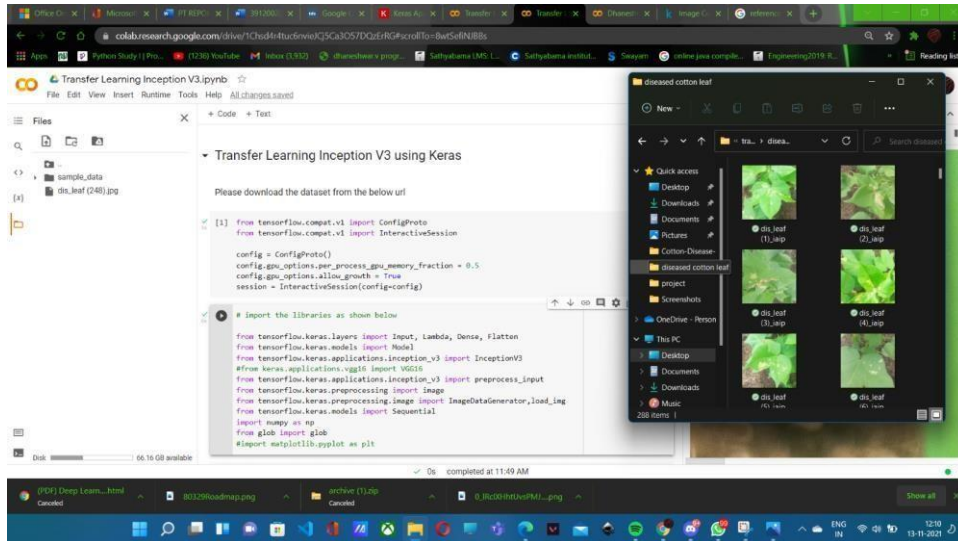


Fig. A.5. Cotton disease Prediction Using datasets and inception v3 with keras

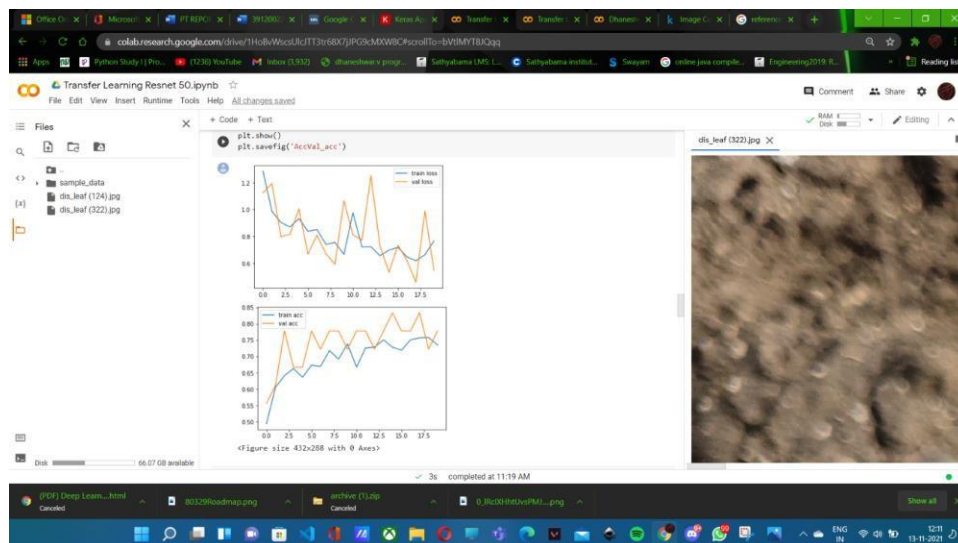


Fig. A.6. Prediction graph

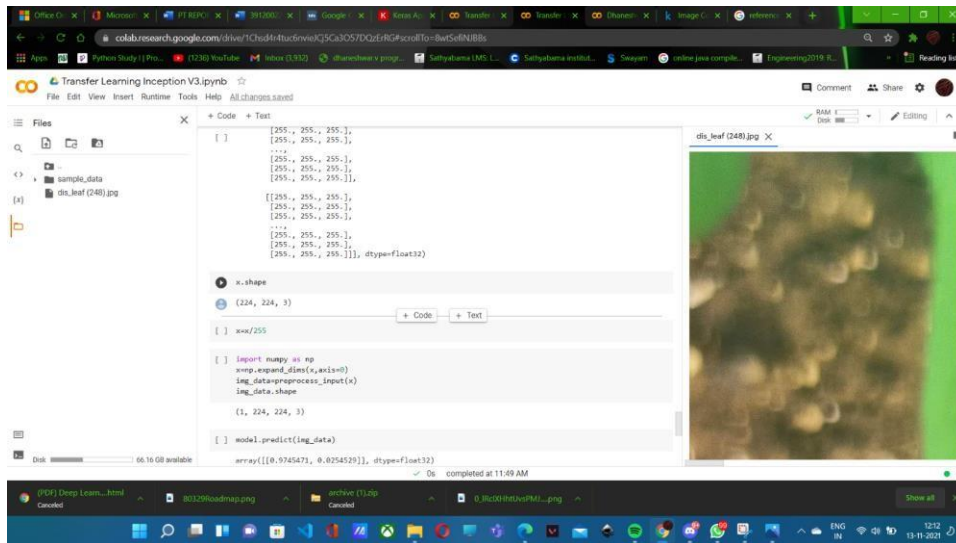


Fig A.7. Result Accuracy

B. SOURCE CODE:

Gcolab

```
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.2
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)

# import the libraries as shown below
s.models import Model
#from tensorflow.keras.applications.resnet152V2 import ResNet152V2
import tensorflow
#from keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.inception_v3 import preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.datasets import mnist
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
```

```

from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
import matplotlib.pyplot as plt

# don't train existing weights
for layer in resnet152V2.layers:
    layer.trainable = False

# useful for getting number of output classes
folders = glob('train/*')

# our layers - you can add more if you want
x = Flatten()(resnet152V2.output)
prediction = Dense(len(folders), activation='softmax')(x)

IMAGE_SIZE = [224, 224]

# create a model object
Model = Model(inputs=resnet152V2.input, outputs=prediction)

train_path = 'Datasets/train'
# view the structure of the model
Model.summary()

valid_path = 'Datasets/test'
Model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)

# Import the Vgg 16 library as shown below and add preprocessing layer to the front of VGG
train_datagen = ImageDataGenerator(rescale = 1./255,
                                    shear_range = 0.2,
                                    zoom_range = 0.2,
                                    horizontal_flip = True)

# Here we will be using imagenet weights
import tensorflow as tf
test_datagen = ImageDataGenerator(rescale = 1./255)

resnet152V2 = tensorflow.keras.applications.ResNet152V2(input_shape=IMAGE

```

```

    )

target_size =
(224, 224),

batch_size =
32,

class_mode =
'categorical')

test_set =
test_datagen.flow_from_directory('test',

import matplotlib.pyplot as plt
# plot the loss
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()
plt.savefig('LossVal_loss')

# plot the accuracy
plt.plot(r.history['accuracy'], label='train acc')
plt.plot(r.history['val_accuracy'], label='val acc')
plt.legend()
plt.show()
plt.savefig('AccVal_acc')

class_mode =
'categorical')
# fit the model
# Run the cell.
It will take
some time to
execute

r = Model.fit(
    training_set,
    validation_data=test_set,
    epochs=20,

from tensorflow.keras.models import load_model
y_pred = model.predict(test_set)
y_pred
import numpy as np
y_pred = np.argmax(y_pred, axis=1)
y_pred

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('model_resnet50.h5')

img_data
validation_step
s=len(test_set) img=image.load_img('Datasets/Test/Coffee/download

```

```

(2).jpg',target
_size=(224,224)
)
x=image.img_to_
array(img)
x

from __future__ import division, print_function
from werkzeug.utils import secure_filename
from flask import Flask, redirect, url_for, request,
render_template
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.resnet50 import
preprocess_input
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
import tensorflow as tf
import tensorflow as tf

a=np.argmax(mod
el.predict(img_
data), axis=1)
a==1

import
tensorflow as
tf
tf.__version__

config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.2
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)

# Keras

# Flask utils
#from gevent.pywsgi import WSGIServer

```

Main.py

```

# -*- coding:
utf-8 -*-
.....

```

Created by
dhanesh

```

model = load_model(model_resnet152V2)

# Define a flask app
app = Flask(__name_)

# Model saved with Keras
model.save()
MODEL_PATH = 'model_resnet152V2.h5'
tf.keras.applications.ResNet152V2(
    include_top=True,

    weights='imagenet',

    input_tensor=None,

    input_shape=None,

    pooling=None,

    classes=1000,

    classifier_activation='softmax'
)

# Load your trained model
def model_predict(img_path, model):
    print(img_path)
    img = image.load_img(img_path, target_size=(224, 224))

    # Preprocessing the image
    x = image.img_to_array(img)
    # x = np.true_divide(x, 255)
    # Scaling
    x = x/255
    x = np.expand_dims(x, axis=0)
    # Be careful how your trained model deals with the input
    # otherwise, it won't make correct prediction!
    # x = preprocess_input(x)

    preds = model.predict(x)
    preds = np.argmax(preds, axis=1)
    if preds == 0:
        preds = "The leaf is diseased cotton leaf"
    elif preds == 1:
        preds = "The leaf is diseased cotton plant"
    elif preds == 2:
        preds = "The leaf is fresh cotton leaf"
    else:
        preds = "The leaf is fresh cotton plant"

    return preds

@app.route('/', methods=['GET'])

```

```

def index():    h)
    # Main page
    return
render_template    # Make prediction
('index.html')    preds = model_predict(file_path, model)
                    result = preds
                    return result

@app.route('/pr    return None
edict',
methods=['GET',
'POST'])

def upload():    if __name__ == '__main__':
                    app.run(port=5001, debug=True)

    if
request.method
== 'POST':

        # Get    Index.html
the file from
post request    {% extends "base.html" %} {% block content %}

        f =
request.files[''
file']    <h2>Image Classifier</h2>

        # Save    <div>
the file to
./uploads    <form id="upload-file" method="post"
                    enctype="multipart/form-data">

                    <label for="imageUpload" class="upload-label">
                        Choose...
                    </label>

                    <input type="file" name="file" id="imageUpload"
accept=".png, .jpg, .jpeg">
                    </form>

basepath =
os.path.dirname
(_file_)

file_path =
os.path.join(

basepath,
'uploads',
secure_filename
(f.filename))

f.save(file_pat    <div class="image-section" style="display:none;">
                    <div class="img-preview">
                        <div id="imagePreview">
                            </div>
                    </div>

```

```

</div>
<div>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
<button
    type="button"
    class="btn btn-
primary btn-lg
" id="btn-
predict">Predic
t!</button>
</div>
</div>
<div
    class="loader"
    style="display:
none;"></div>
<h3
id="result">
<span>
</span>
</h3>
<div>
    <span>
</span>
</div>
Base.html
<html
lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>AI Demo</title>
    <link
        href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.
css" rel="stylesheet">
    <script
        src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"
></script>
    <script
        src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></scrip
t>
    <script
        src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js
"></script>
    <link href="{ { url_for('static', filename='css/main.css')
}} " rel="stylesheet">
</head>
<body>
    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">AI Demo</a>
            <button class="btn btn-outline-secondary my-2 my-sm-
0" type="submit">Help</button>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">{% block
content %}{% endblock %}</div>
    </div>
</body>
<footer>

```

```
<script
src="{{
url_for('static </html>
',
filename='js/ma
in.js') }}"
type="text/java
script"></scrip
t>
</footer>
```