DAYANANDA SAGAR UNIVERSITY

MINOR PROJECT REPORT

ON

Using DL to Determine Cotton Crop Health

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

Ronald Santhosh

Shishira S

Sohan Somaya

Tanmay Sharma

5th SEMESTER, 2021

Under the Guidance of

C V Satayanarayana Reddy
Associate Professor

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate,Bangalore-560068

CERTIFICATE

This is to certify that the minor project report entitled "Using DL to Determine Cotton Crop Health " being submitted by Mr/Ms Ronald Santhosh, Shishira S, Sohan Somaya ,Tanmay Sharma bearing USN ENG19CS0263 ,ENG19CS0296, ENG19CS0314, ENG19CS0333 has satisfactorily completed her Minor Project as prescribed by the University for the 5th sem. B.Tech Program in Computer Science & Engineering during the academic year 2021 – 22 at the School of Engineering, Dayananda Sagar University, Bangalore.

Date:14-02-2022 Signature of the faculty in-charge

Signature of Chairman
Department of Computer Science & Engineering

DECLARATION

We hereby declare that the work presented in this minor project entitled as "Detection of Disease in Cotton Plants Using Machine Learning", has been carried out by us and it has not been submitted for the award of any degree, diploma, or minor project of any other college or university.

Ronald Santosh	ENG19CS0263
Shishira S	ENG19CS0296
Sohan Somaya	ENG19CS0314
Tanmay Sharma	ENG19CS0333

ACKNOWLEDGEMENT

The success and final outcome of this software requirement document required a lot of guidance and assistance from many people and we are extremely privileged to have got this all through the completion of the project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank our mentor, Professors, and the Chairman for providing us an opportunity to do the Software Requirement Document and giving us all support and guidance which made me complete the project duly. We are extremely thankful to all for providing such nice support and guidance

We are especially thankful to our **Dean & Chairman, Dr. A Srinivas**, Department of Computer Science & Engineering who continuously helped us throughout the project, and without his guidance, this project would have been an uphill task.

We are pleased to acknowledge Prof CVSN Reddy, Prof / Associate Prof / Assistant Professor, Department of Computer Science & Engineering for his invaluable guidance, support, motivation, and patience during the course of this mini-project work.

We are thankful for and fortunate enough to get constant encouragement, support, and guidance from all Teaching staff of the Computer Science Engineering department, which helped us in completing our report. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

Table of Contents

1. Introduction

Agriculture is the backbone of any stable economy. It contributes 17% to India's GDP and is the primary occupation of over 70% of India's rural households.
With the new strains and mutations of fungi and bacteria, crop infection can pose a huge risk to the overall economy of a country. Hence, it is vital that it be maintained and not be jeopardized.

This project aims to aid this cause, by helping identify the cause of damage to the crop and provide a suitable remedy. To ensure minimal loss to the cultivated crop, it is crucial to supervise its growth.

Convolutional Neural Network is a class of Deep learning used majorly for image classification, other mainstream tasks such as image segmentation and signal processing. The main aim of our project is to find a solution to the plant diseases detection using the simplest approach while making use of minimal computing resources to achieve better results compared to the traditional models.

1.1 Problem Statement

To design and create a web application that uses a CNN to detect the diseases from a given image of a cotton plant leaf from a user.

2. Literature Survey

1. **Sammy V . Militante, Bobby D. Gerardo, Nanette V . Dionisio, "Plant Leaf Detection and Disease Recognition using Deep Learning" Proceeding of the IEEE Eurasia Conference on IOT**

   This paper proposed a CNN for the classifying the disease types and in this paper the author used 9 different varieties of leaf diseases of tomato, grape, corn, apple and sugarcane.

2. **Ch. Usha Kumari, S. Jeevan Prasad, G. Mounika, "Leaf Disease Detection: Feature Extraction with K-means clustering and Classification with ANN "Proceedings of the 3rd International Conference on Computing Methodologies and Communication (ICCMC)",**

   This paper is based on a system that deploys the methods of K- Means clustering and Artificial Neural Network and performs computation of various features like Contrast, Correlation, Energy, Mean, Standard Deviation and Variance were performed.

3. **Plant Disease Detection and Classification using CNN, Rinu R, Manjula S H**

   This project based on a deep learning approach called CNN is utilized to build 13 different plant leaf disease identification, detection and recognition system.

   This approach utilized a minimum set of layers to identify the diseases of seven classes. The neural network is trained with the Plant Village dataset.

3. Requirement Analysis

Minimum Software Requirement -

Python == 3.7.7

TensorFlow == 2.1.0

Keras == 2.4.3

NumPy == 1.18.5

Flask == 1.1.2

System Requirement -

GPU

8GB RAM,

500MB of storage space

4. Design Methods

4.1 Algorithm

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex.

A ConvNet is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

There are two tasks to be performed for best recognition of plant diseases. The first is to detect objects within an image coming from several classes, which is called object localization. The second is to classify images, each labelled with one of several categories, which is called image classification. The CNN model has seven different layers. Each layer has certain information processed in them. Those seven layers are as follows: Input layer, Output Layer, Convolutional Layer, Fully, Soft-max layer, connected layer, Pooling Layer.

Input layer: It contains data in the form of image. The parameters include height, width, depth and color information of the image (RGB). Input size is fixed to 255 X 255 RGB image.

Convo layer: Convolutional layer is also called as feature extraction layer. This layer extracts the prominent features from the given collection of images using dot products of the image dimensions.

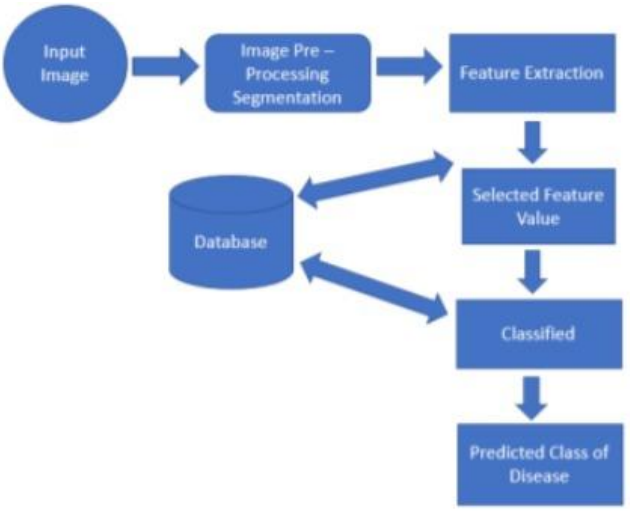
Pooling Layer: The pooling layer helps to reduce the computational power in order to process the data by decreasing (or) reducing the dimensions of the featured matrix obtained by using the dot products.

Fully connected layer: It comprises of loads, neurons and biases. It connects neurons from one convolutional layer to another.

Softmax Layer/ Logistic Layer: Softmax executes multi-classification. Logistic layer executes the binary classification. It determines the probability of the presence of a given object in the image. If the object is present in the image, then the probability is ‘1’otherwise it is ‘0’.

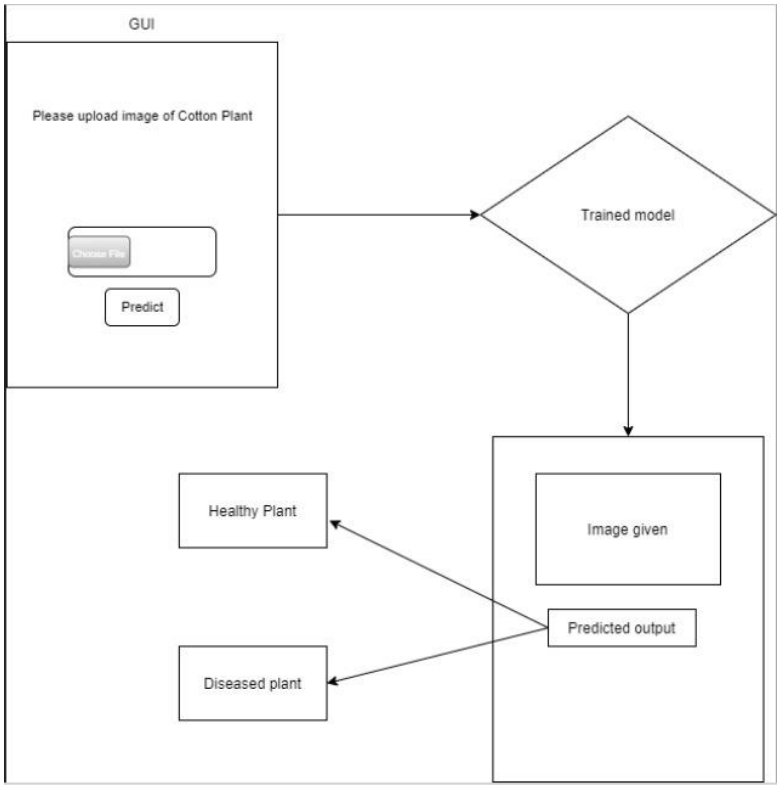
Activation Function- ReLU: It transforms the total weighted input through the node and puts it into the operation, activates the node. Rectified Linear Unit (ReLU) is an activation function used in the neural networks for convolutional operations.

4.2 Architecture Diagram



4.3 Flow chart/ DFD/ UML Diagrams

The below diagram depicts the flow of control from taking the input to the prediction of the desired output.



5. Project Breakdown

GUI using Flask : Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file.

Model : Convolutional neural network (CNN) is a class of artificial neural networks that has become dominant in various computer vision tasks. It is designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers.

6. Implementation

model building and training code:

```
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt
keras.__version__


train_data_path=  "E:/tanmay/PyWorkspace/cotton_pant_disease_prediction_ai/new_dataset/Cotton Disease/train"
validation_data_path= "E:/tanmay/PyWorkspace/cotton_pant_disease_prediction_ai/new_dataset/Cotton Disease/val"



def plotImages(images_arr):
    fig, axes = plt.subplots(1, 5, figsize=(20, 20))
    axes = axes.flatten()
    for img, ax in zip(images_arr, axes):
        ax.imshow(img)
    plt.tight_layout()
    plt.show()



#  generate more images using below parameters
```

```
training_datagen = ImageDataGenerator(rescale=1. / 255,
                                      rotation_range=40,
                                      width_shift_range=0.2,
                                      height_shift_range=0.2,
                                      shear_range=0.2,
                                      zoom_range=0.2,
                                      horizontal_flip=True,
                                      fill_mode='nearest')

# this is a generator that will read pictures found in
# at train_data_path, and indefinitely generate
# batches of augmented image data
training_data = training_datagen.flow_from_directory(train_data_path,  #  target directory
                                                     target_size=(150, 150),  # images  resized to 150x150
                                                     batch_size=32,
                                                      class_mode='binary')  # since we use binary_crossentropy loss, we
need binary labels

training_data.class_indices



# only rescaling
valid_datagen = ImageDataGenerator(rescale=1. / 255)

#  data validation
valid_data = valid_datagen.flow_from_directory(validation_data_path,
                                               target_size=(150, 150),
                                               batch_size=32,
                                               class_mode='binary')

images = [training_data[0][0][0] for i in range(5)]
plotImages(images)
'''
model_path = '/content/drive/My Drive/My ML Project /DL Project/CNN/cotton plant disease
prediction/v3_red_cott_dis.h5'
```

```
checkpoint = ModelCheckpoint(model_path, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
callbacks_list = [checkpoint]
'''
# Building cnn model
cnn_model = keras.models.Sequential([
    keras.layers.Conv2D(filters=32, kernel_size=3, input_shape=[150, 150, 3]),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Conv2D(filters=64, kernel_size=3),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Conv2D(filters=128, kernel_size=3),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Conv2D(filters=256, kernel_size=3),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),

    keras.layers.Dropout(0.5),
    keras.layers.Flatten(),  # neural network building
    keras.layers.Dense(units=128, activation='relu'),  # input layers
    keras.layers.Dropout(0.1),
    keras.layers.Dense(units=256, activation='relu'),
    keras.layers.Dropout(0.25),
    keras.layers.Dense(units=4, activation='softmax')  # output layer
])

# compile cnn model
cnn_model.compile(optimizer=Adam(lr=0.0001), loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# train cnn model
history = cnn_model.fit(training_data,
                        epochs=500,
                        verbose=1,
                        validation_data=valid_data
                        )  # time start 16.06
```

```
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

history.history
```

main program :

```
#Import necessary libraries
from flask import Flask, render_template, request

import numpy as np
import os

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.models import load_model

#load model
model =load_model("model/v3_pred_cott_dis.h5")
```

```
print('@@ Model loaded')


def dis_pred(cott_plant):
  test_image = load_img(cott_plant, target_size = (150, 150)) # load image
  print("@@ Got Image for prediction")

  test_image = img_to_array(test_image)/255 # convert image to np array and normalize
  test_image = np.expand_dims(test_image, axis = 0) # change dimension 3D to 4D

  result = model.predict(test_image).round(3) # predict diseased plant or not
  print('@@ Raw result = ', result)

  pred = np.argmax(result) # get the index of max value

  if pred == 0:
    return "Healthy Cotton Plant", 'healthy_plant_leaf.html' # if index 0 burned leaf
  elif pred == 1:
      return 'Diseased Cotton Plant', 'disease_plant.html' # # if index 1
  elif pred == 2:
      return 'Healthy Cotton Plant', 'healthy_plant.html'  # if index 2  fresh leaf
  else:
    return "Healthy Cotton Plant", 'healthy_plant.html' # if index 3

#------------>>dis_pred<<--end


# Create flask instance
app = Flask(__name__)

# render index.html page
@app.route("/", methods=['GET', 'POST'])
def home():
        return render_template('index.html')
```

```
# get input image from client then predict class and render respective .html page for solution
@app.route("/predict", methods = ['GET','POST'])
def predict():
    if request.method == 'POST':
        file = request.files['image'] # fet input
        filename = file.filename
        print("@@ Input posted = ", filename)

        file_path = os.path.join('static/user uploaded', filename)
        file.save(file_path)

        print("@@ Predicting class.....")
        pred, output_page = dis_pred(cott_plant=file_path)

        return render_template(output_page, pred_output = pred, user_image = file_path)

# For local system & cloud
if __name__ == "__main__":
    app.run(threaded=False,)
```

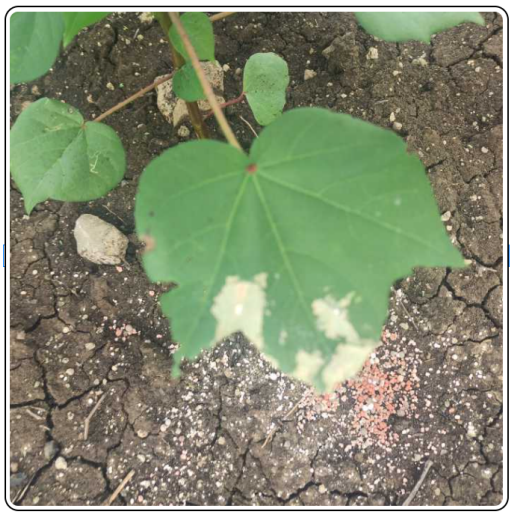
7. Testing

We have three possible outcomes while running the program

1. healthy no problem

There is no disease on the cotton Plant.

2. healthy leaves burnt



There is no disease on the cotton plant.
Although the chemical fertilizer has fallen on the leaves of the tree, the leaves are burnt, but there is no need to worry.

3. unhealthy plant

Diseased Cotton Plant

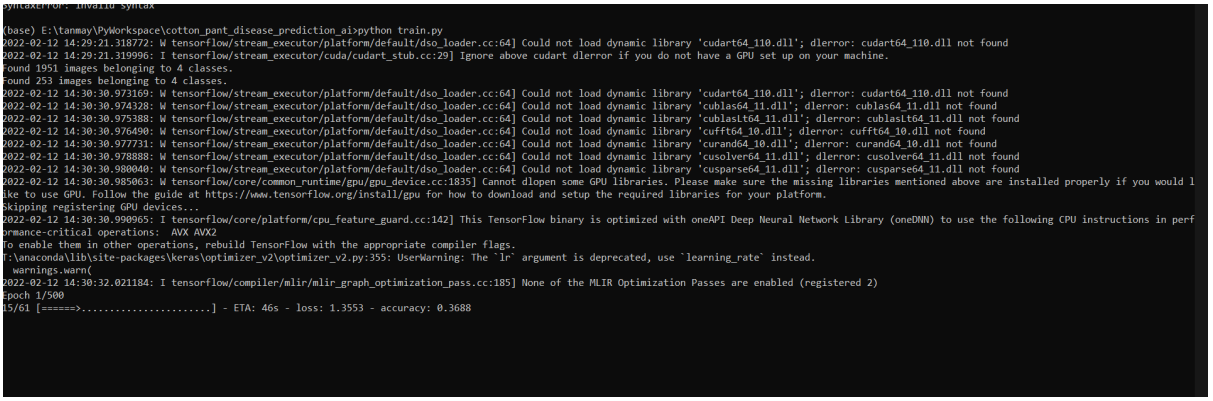
Disease Name :
Attack of Leaf Sucking and Chewing Pests

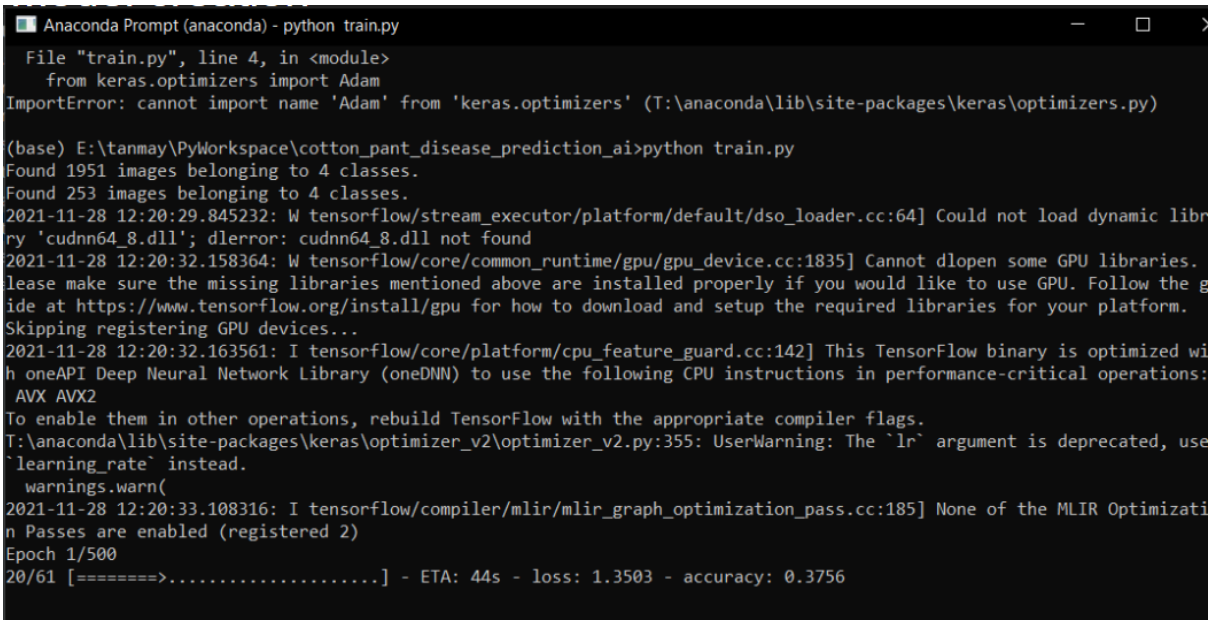
Solution for Disease

Use any one Systemic Insecticide, which contain *Flonicamid 50%/ Thiamethoxam 25% WG / Imidacloprid 17.8 Sl / Acetamiprid 20% SP.*

8. Results/Output Screenshots

- model creation





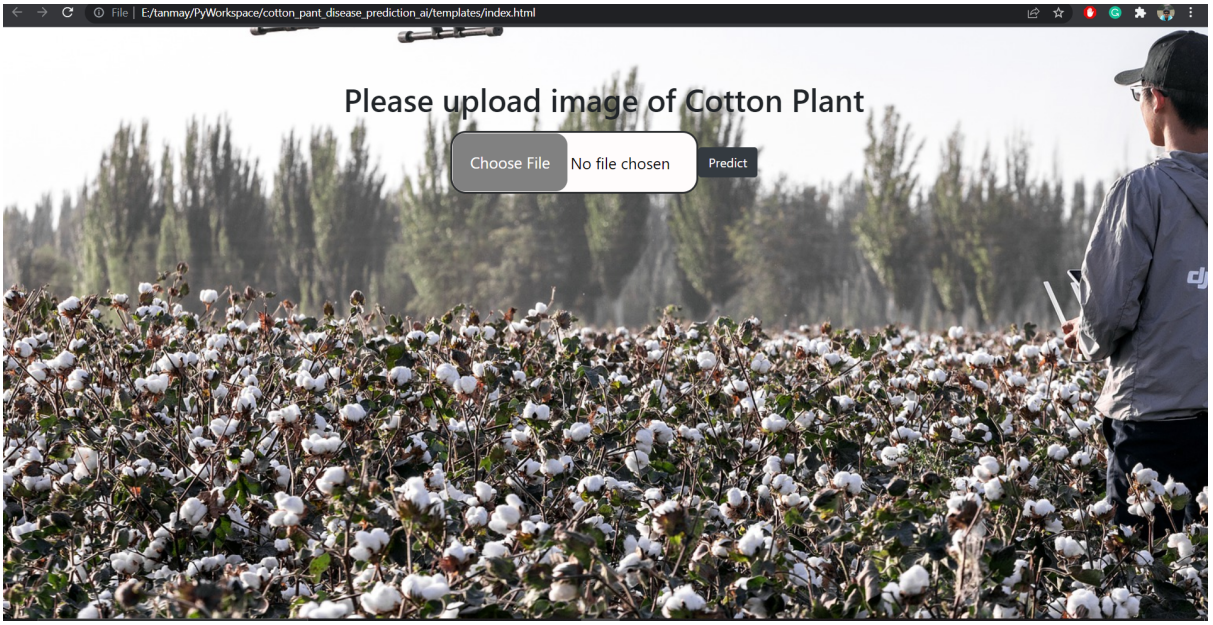
- Server running

```
(base) C:\Users\tanma>e:

(base) E:\>cd E:\tanmay\PyWorkspace\cotton_pant_disease_prediction_ai

(base) E:\tanmay\PyWorkspace\cotton_pant_disease_prediction_ai>python app.py
@@ Model loaded
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
@@ Input posted =  dd (1)_iaip.jpg
@@ Predicting class......
@@ Got Image for prediction
@@ Raw result =  [[0.    0.996 0.    0.004]]
127.0.0.1 - - [12/Feb/2022 15:06:17] "←[37mPOST /predict HTTP/1.1←[0m" 200 -
@@ Input posted =  dd (4)_iaip.jpg
@@ Predicting class......
@@ Got Image for prediction
@@ Raw result =  [[0.    0.915 0.    0.085]]
127.0.0.1 - - [12/Feb/2022 15:06:25] "←[37mPOST /predict HTTP/1.1←[0m" 200 -
127.0.0.1 - - [12/Feb/2022 15:06:25] "←[37mGET /static/user%20uploaded/dd%20(4)_iaip.jpg HTTP/1.1←[0m" 200 -
@@ Input posted =  d (2)_iaip.jpg
@@ Predicting class......
@@ Got Image for prediction
@@ Raw result =  [[0.    0.    0.999 0.   ]]
127.0.0.1 - - [12/Feb/2022 15:06:44] "←[37mPOST /predict HTTP/1.1←[0m" 200 -
```

- Main/Start page



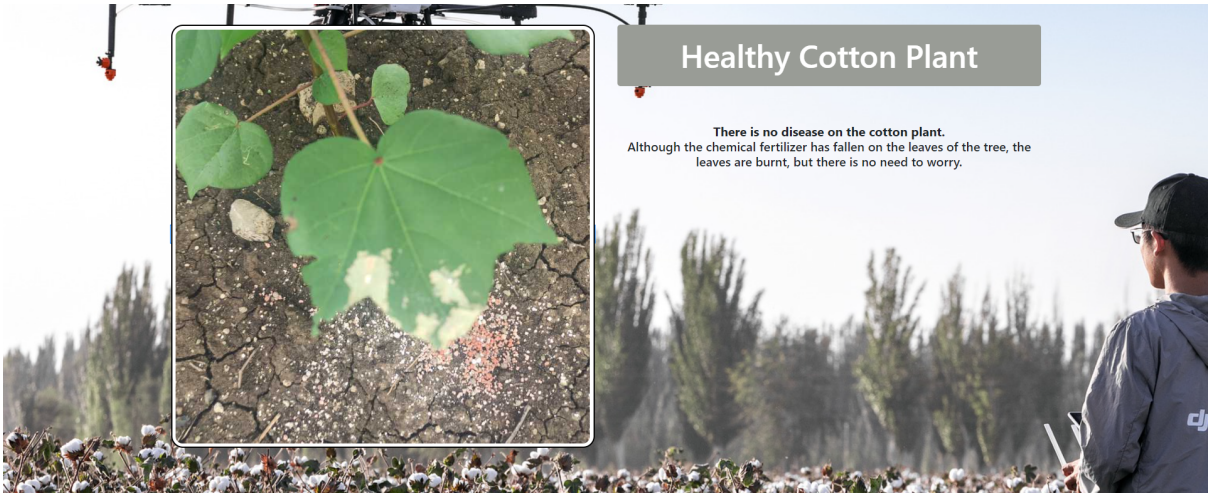
- Demo Healthy page

- Demo Unhealthy page

- Demo Healthy but burnt

- Healthy plant page

- Healthy but burnt plant page



- Unhealthy plant page

9. Conclusion and Future work

Conclusion:

As seen our model predicts if a cotton plant has a disease or not and provides a rudimentary solution as well. Though the accuracy is high we still have a few false negatives. We are hoping by increasing the dataset we reduce the error.

Future work:

- Increase dataset
- increase disease prediction capabilities
- Add in more values to predict
- increase crops
- Integrate the app to raspberry pi/jetson nano to add to a drone

10. References

1. Sammy V . Militante, Bobby D. Gerardo, Nanette V . Dionisio, “Plant Leaf Detection and Disease Recognition using Deep Learning” Proceeding of the IEEE Eurasia Conference on IOT

2. Ch. Usha Kumari, S. Jeevan Prasad, G. Mounika, “Leaf Disease Detection: Feature Extraction with K-means clustering and Classification with ANN “Proceedings of the 3rd International Conference on Computing Methodologies and Communication (ICCMC)”,

3. Plant Disease Detection and Classification using CNN, Rinu R, Manjula S H

4. S. D. Khirade and A. B. Patil. “Plant Disease Detection Using Image Processing”, Proceeding of the International Conference on Computing Communication Control and Automation. Feb. 2015, pp. 768–771.

5. Sharada P Mohanty, David P Hughes, and Marcel Salath, “Using deep learning for image-based plant disease detection”. In: Frontiers in plant science 7 (2016), p. 1419.