Plant Disease Fertilizer Recommendation System

Developed By: G.Ruthvik, V.Bhavani, Sangeetha, Vamshi

Smart Bridge – Major Project Report

1. INTRODUCTION

1.1 Overview

Deep learning is a subset of deep learning that involves the use of artificial neural networksto analyze and learn from data. It has been applied in a variety of fields, including agriculture,to develop systems that can analyze large amounts of data and make predictions or recommendations.

In the context of plant disease and fertilizer recommendation systems, deep learning could be used to analyze data on plant growth, soil conditions, weather patterns, and other factors that may affect the likelihood of disease outbreaks or the effectiveness of different fertilizers. This data could include images of plants, sensor readings, and other types of information.

The system could then use this information to make recommendations for the prevention or treatment of plant diseases, or for the use of specific fertilizers. For example, the system might recommend the use of a particular pesticide to prevent a certain type of disease, or suggest the application of a particular fertilizer to improve soil conditions and promote healthy plant growth.

1.2 Purpose

The purpose of a plant disease and fertilizer recommendation system based on deep learning is to use artificial intelligence and machine learning techniques to analyze data on plant growth, soil conditions, weather patterns, and other factors, and make recommendations for the prevention or treatment of plant diseases or the use of specific fertilizers.

Such a system could be used by farmers, gardeners, or other professionals to improve crop yields, reduce losses due to disease, and optimize the use of fertilizers and other inputs. It could also help to reduce the environmental impact of agriculture by minimizing the use of pesticides and other chemicals and improving soil health.

2. LITERATURE SURVEY

2.1 Existing Problem

One existing problem with plant disease and fertilizer recommendation systems based on deep learning is the need for large amounts of high-quality data in order to train and validatethe system. Obtaining and curating such data can be time-consuming and costly, and may require the collaboration of multiple organizations or individuals.

Another issue is the potential for bias in the data used to train the system, which can affect the accuracy and fairness of the recommendations. For example, if the data is disproportionately collected from certain regions or types of crops, the system may not be as effective in other areas or for other crops.

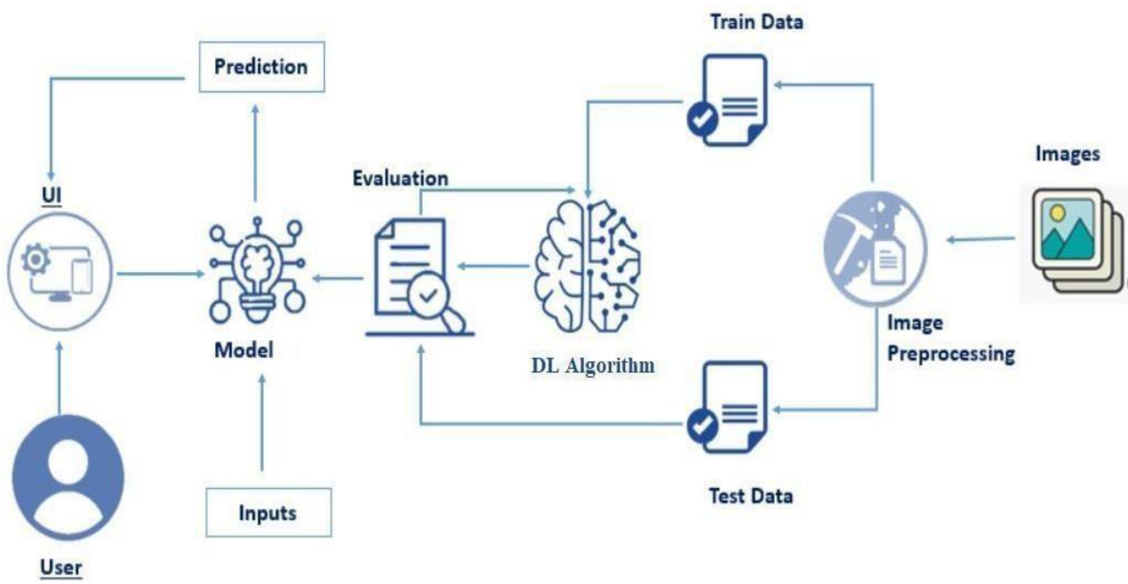
2.2 Proposed Solution

Data collection and curation to improve the accuracy and effectiveness of the system, it is important to collect and curate large amounts of high-quality data from a diverse range of sources. This may involve collaboration with multiple organizations or individuals, as well as efforts to standardize data formats and ensure the data is properly labeled and annotated.

Bias reduction to reduce bias in the data used to train the system, it is important to ensure that the data is representative of the populations and environments for which the system will be used. This may involve efforts to collect data from a wide range of locations, crops, and other factors.

3. THEORITICAL ANALYSIS:

3.1 Block diagram



3.2 HARDWARE AND SOFTWARE REQUIREMENTS IN THE PROJECT:

For running a machine learning model on the system you need a system with minimum of 16 GB RAM in it and you require a good processor for high performance of the model.
In the list of **software requirements** you must have:

- Jupyter Notebook for programming, which can be installed by Anaconda IDE.
- Python packages
- A better software for running the html,js and css files for application building phase e.g.spyder.

4. EXPERIMENTAL INVESTIGATIONS:

An experimental investigation of a plant disease and fertilizer recommendation system based on deep learning could involve several steps:

Data collection and preparation: This would involve collecting and preparing the data that will be used to train and test the system, including images of plants, sensor readings, weather data, and other types of information. The data should be cleaned and pre-processed as needed, and divided into training, validation, and test sets.

Model development: This would involve developing and implementing the deep learning model that will be used to analyze the data and make recommendations. This could involve designing and training a convolutional neural network to recognize plant diseases from images, or using a more general-purpose neural network to analyze other types of data.

Model evaluation: This would involve evaluating the performance of the model on the test data, and comparing its performance to other models or baseline approaches. This could include measures such as accuracy, precision, and recall, as well as more subjective metrics such as the interpretability and explainability of the model's recommendations.

Model deployment: If the model performs well in the evaluation phase, it could be deployed in a real-world setting, such as on a farm or in a garden. This would involve developing an appropriate user interface and providing guidance and support to users as needed.

5. FLOW CHART:-

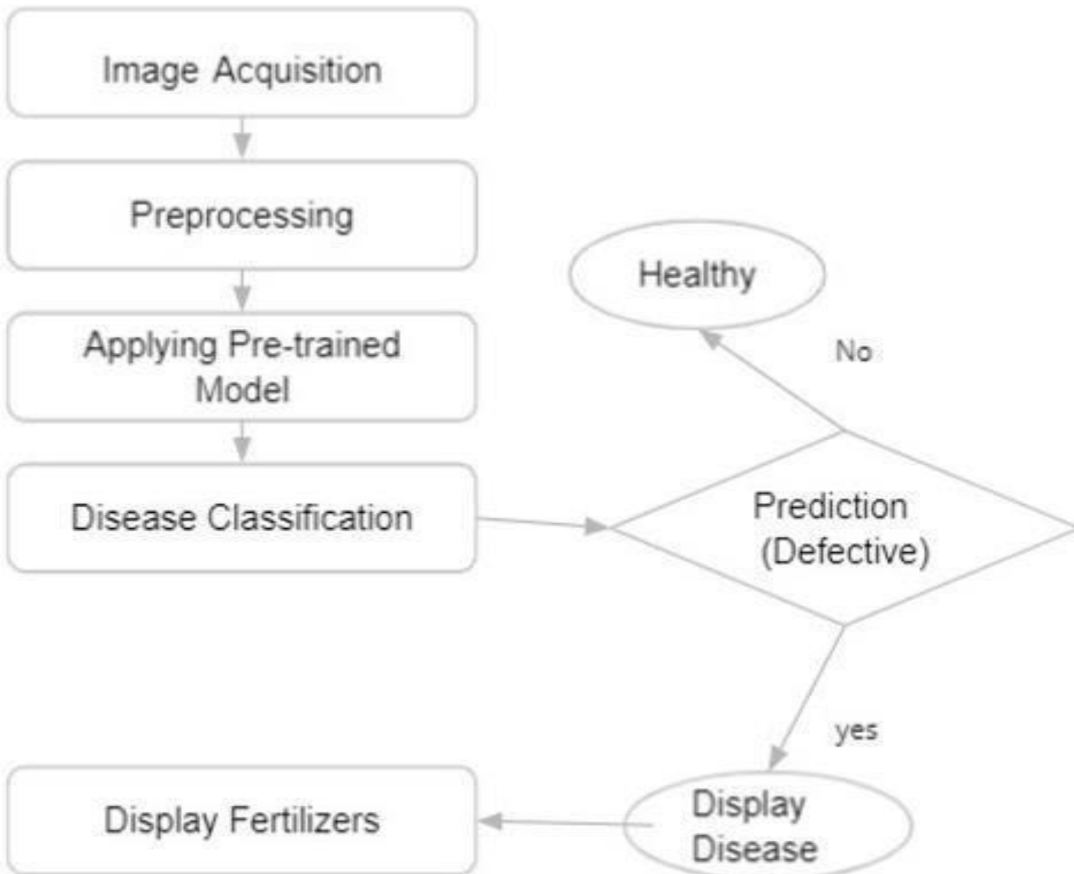
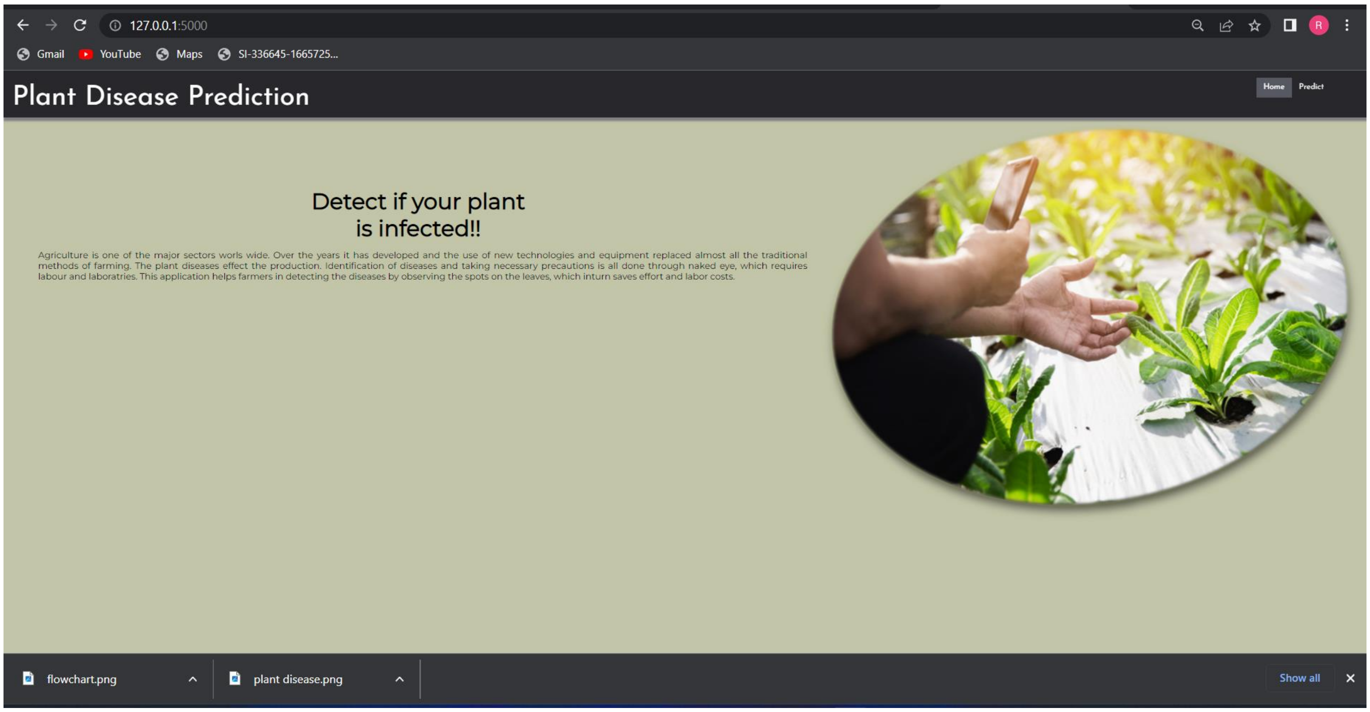
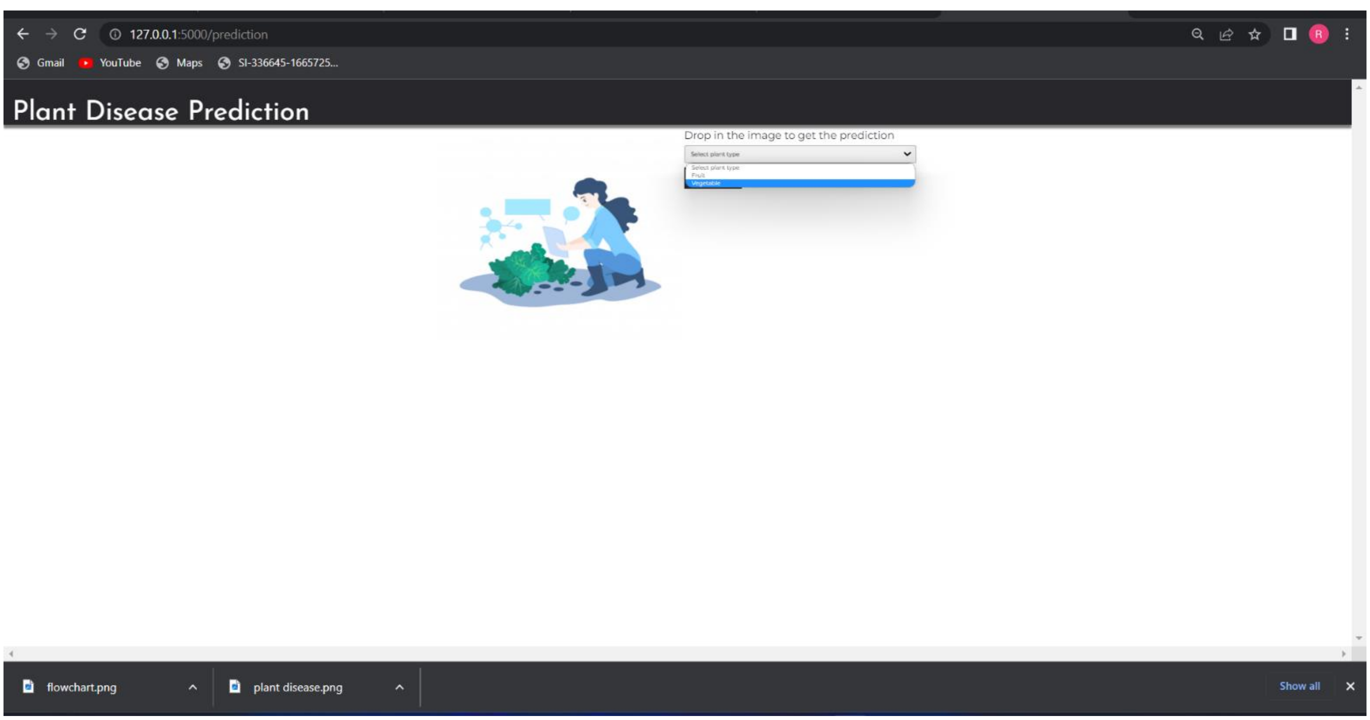


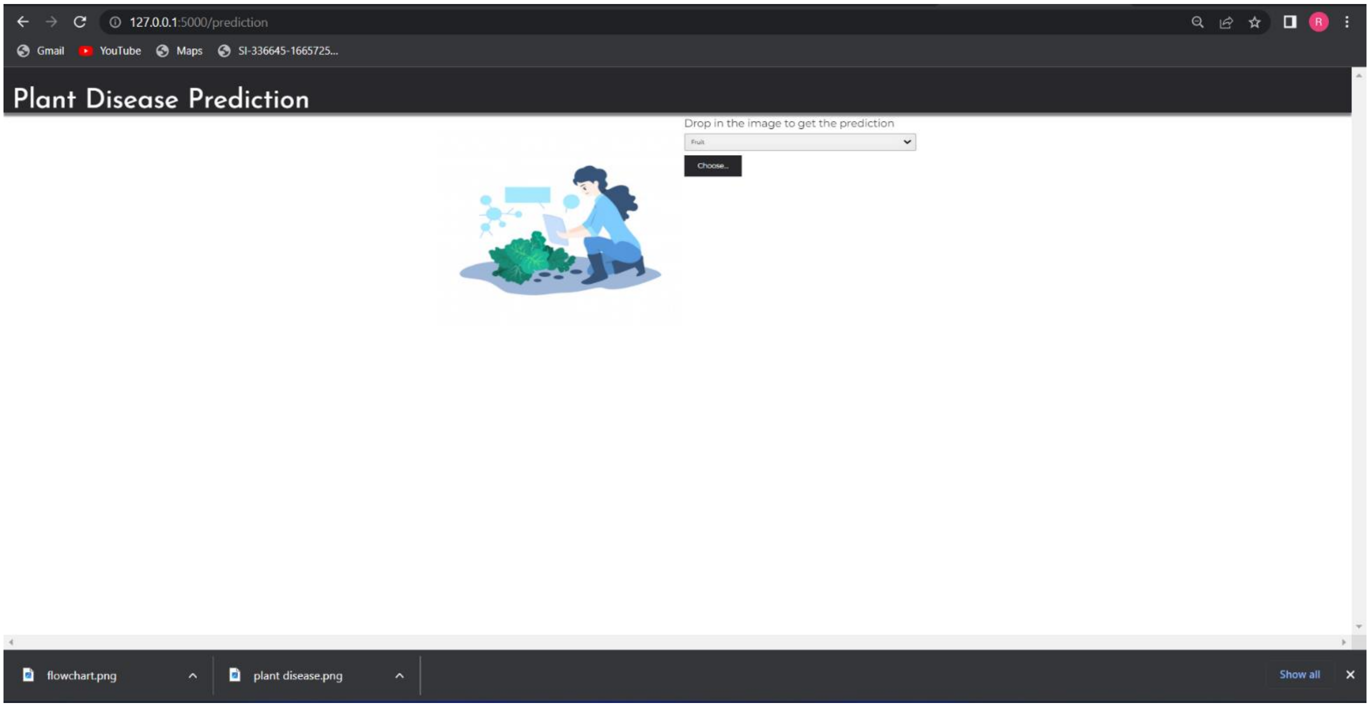
Fig. 2. Flowchart of Training model

6. RESULTS:-

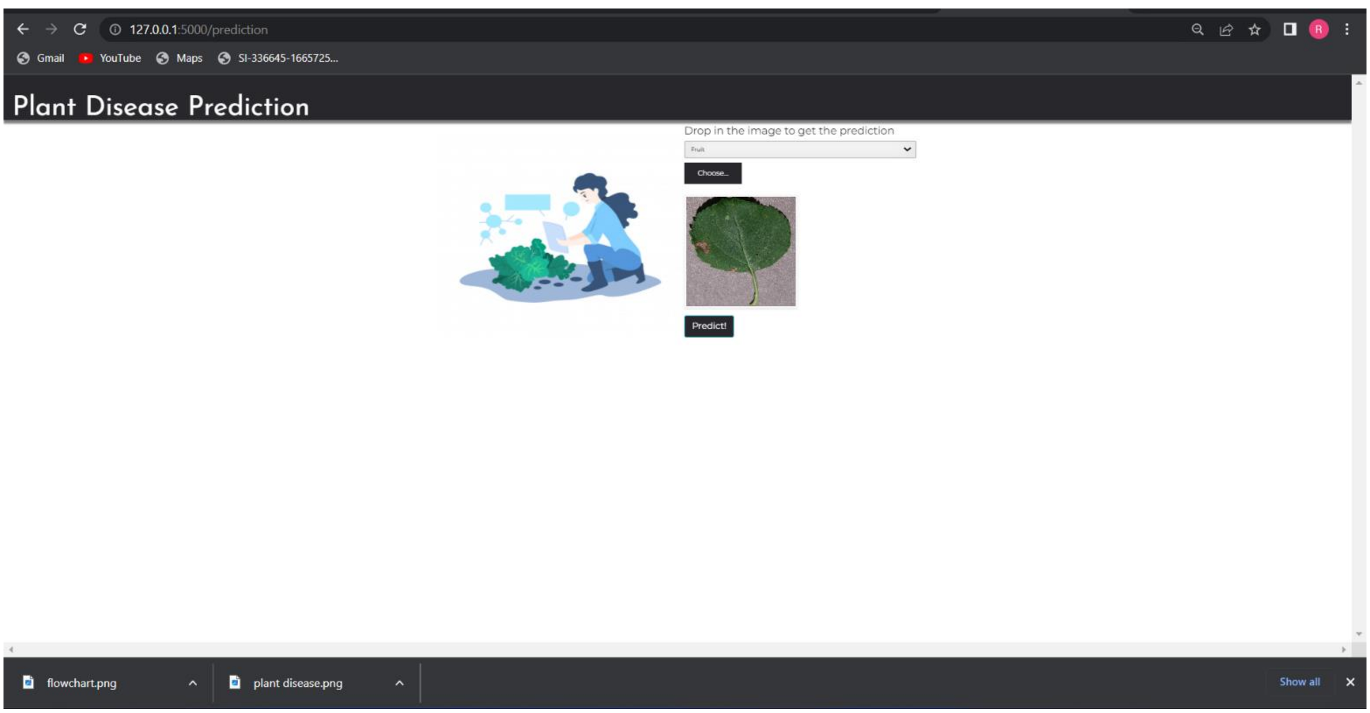
Final output of the project:







127.0.0.1:5000/prediction
Gmail
YouTube
Maps
SI-336645-1665725...
Plant Disease Prediction
Drop in the image to get the prediction
Choose...
flowchart.png
plant disease.png
Show all



127.0.0.1:5000/prediction
Gmail
YouTube
Maps
SI-336645-1665725...
Plant Disease Prediction
Drop in the image to get the prediction
Choose...
Predict!
flowchart.png
plant disease.png
Show all

7. ADVANTAGES & DISADVANTAGES

ADVANTAGES

Accuracy: Deep learning models can achieve high levels of accuracy when trained on large amounts of data, making them well-suited for tasks such as image recognition or data analysis. This can help to improve the accuracy and reliability of the recommendations made by the system.

Efficiency: Deep learning models can process large amounts of data quickly and efficiently, allowing them to make recommendations in real-time or near real-time. This can be particularly useful in situations where timely decisions are critical, such as in the prevention or treatment of plant diseases.

Scalability: Deep learning models can be easily scaled up or down as needed, making them well-suited for use in a variety of contexts, from small gardens to large commercial farms.

Interpretability: With the proper design and implementation, deep learning models can be made more interpretable and explainable, allowing users to understand the basis for the recommendations made by the system.

DISADVANTAGES

Data requirements: Deep learning models require large amounts of high-quality data in order to be trained and validated, which can be time-consuming and costly to obtain.

Bias: There is a risk of bias in the data used to train the system, which can affect the accuracy and fairness of the recommendations made by the system.

Complexity: Deep learning models can be complex and difficult to understand, which may make it challenging to interpret the recommendations made by the system or to debug problems that arise.

Explanation: It may be difficult to explain the basis for the recommendations made by the system to users, which could limit trust and adoption.

Dependency: The system may become dependent on the data used to train it, and may not generalize well to new or unseen data. This could limit its usefulness in a rapidly changing environment.

8. APPLICATIONS:-

There are many potential applications for deep learning in the development of plant disease and fertilizer recommendation systems. Here are a few examples:

Image recognition: A deep learning system could be trained to recognize plant diseases by analyzing images of infected plants. The system could then provide recommendations for the appropriate treatment, such as the use of a particular pesticide or fertilizer.

Data analysis: A deep learning system could be trained to analyze data on plant growth, soil conditions, and weather patterns to predict the likelihood of disease outbreaks and recommend preventive measures such as the use of specific fertilizers.

Natural language processing: A deep learning system could be trained to understand and respond to questions about plant diseases and fertilization in natural language, allowing farmers or gardeners to get personalized recommendations through a chatbot or other interface.

Predictive maintenance: A deep learning system could be used to analyze data on the health and performance of crops and soil, and provide recommendations for maintenance and repair tasks such as irrigation, fertilization, and pest control.

Overall, deep learning has many potential applications in the development of plant disease and fertilizer recommendation systems, and can be used to improve the accuracy, efficiency, and effectiveness of these systems.

9. CONCLUSION-

In conclusion, deep learning techniques offer many potential advantages for the development of plant disease and fertilizer recommendation systems, including accuracy, efficiency, scalability, and interpretability. However, there are also potential disadvantages to consider, including the need for large amounts of high-quality data, the risk of bias, the complexity of the models, the difficulty of explaining the recommendations, and the potential for dependency on the data used to train the system.

To maximize the benefits of deep learning in plant disease and fertilizer recommendation systems, it is important to carefully consider these and other challenges, and to develop and deploy the systems in a way that addresses these issues. This may involve efforts to collect and curate high-quality data, to reduce bias in the data, to increase the interpretability and explainability of the models, and to provide clear guidance and support to users. By addressing these challenges, it should be possible to develop and deploy plant disease and fertilizer recommendation systems based on deep learning that are accurate, efficient, and effective.

10. FUTURE SCOPE:-

There is significant potential for the future development and use of deep learning in plant disease and fertilizer recommendation systems. Some possible areas of focus for future work could include:

Improved data collection and curation: As deep learning models become more sophisticated and data-driven, there will be a need for better methods of collecting and curating data from a wide range of sources. This could involve efforts to standardize data formats and ensure the data is properly labeled and annotated, as well as to develop new types of sensors and other data-gathering technologies.

Bias reduction: As deep learning models become more widely used, there will be a need to address concerns about bias in the data and the recommendations made by the models. This could involve efforts to collect data from a more diverse range of sources and to develop techniques for reducing bias in the data and the models.

Interpretability and explainability: To increase the adoption and trust of deep learning models, there will be a need to develop techniques for making the models more interpretable and explainable, and for providing context and background information to users.

Integration with other technologies: Deep learning models for plant disease and fertilizer recommendation could be integrated with other technologies, such as precision agriculture, to improve their effectiveness and efficiency.

Overall, the future scope of deep learning in plant disease and fertilizer recommendation systems is wide and varied, and will likely involve ongoing research and development efforts in a variety of areas.

APPENDIX :-

app.py

```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request, render_template, redirect, url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app = Flask(__name__)
global sess

global graph
graph=tf.compat.v1.get_default_graph()



model = load_model("fruit.h5")
model1=load_model("vegetable.h5")


@app.route('/')
def home():
    return render_template('home.html')


@app.route('/prediction')
def prediction():
    return render_template('predict.html')

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

def predict():
    if request.method == 'POST':

        f = request.files['image']



        basepath = os.path.dirname('__file__')
        file_path = os.path.join(
            basepath, 'Dataset', secure_filename(f.filename))
        f.save(file_path)
        filepath = os.path.join(basepath,'uploads',f.filename)
        print("upload folder is ", filepath)
        f.save(filepath)
```

```
    img = image.load_img(file_path, target_size=(128, 128))

    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    plant=request.form['plant']
    print(plant)

    if(plant=="vegetable"):
      preds = model.predict(x)
      preds = np.argmax(preds)
      print(preds)
      df=pd.read_excel('precautions - veg.xlsx')
      print(df.iloc[preds]['caution'])
    else:
      preds = model1.predict(x)
      preds = np.argmax(preds)

      df=pd.read_excel('precautions - fruits.xlsx')
      print(df.iloc[preds]['caution'])


    return df.iloc[preds]['caution']

if __name__ == "__main__":
  app.run(debug=False)
```

main.js

```
$(document).ready(function () {
  // Init
  $('.image-section').hide();
  $('.loader').hide();
  $('#result').hide();

  // Upload Preview
  function readURL(input) {
    if (input.files && input.files[0]) {
      var reader = new FileReader();
      reader.onload = function (e) {
        $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
        $('#imagePreview').hide();
        $('#imagePreview').fadeIn(650);
      }
      reader.readAsDataURL(input.files[0]);
    }
  }
  $("#imageUpload").change(function () {
    $('.image-section').show();
```

```
    $('#btn-predict').show();
    $('#result').text('');
    $('#result').hide();
    readURL(this);
});

// Predict
$('#btn-predict').click(function () {
    var form_data = new FormData($('#upload-file')[0]);

    // Show loading animation
    $(this).hide();
    $('.loader').show();

    // Make prediction by calling api /predict
    $.ajax({
        type: 'POST',
        url: '/predict',
        data: form_data,
        contentType: false,
        cache: false,
        processData: false,
        async: true,
        success: function (data) {
            // Get and display the result
            $('.loader').hide();
            $('#result').fadeIn(600);
            $('#result').text('Prediction: '+data);
            console.log('Success!');
        },
    });
```

final.css

```css
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #28272c;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #C2C5A8;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #28272c; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }}
```