PLANT DISEASE FERTILIZER RECOMMENDATION SYSTEM

A UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

VENUVANKA BHAVANI	**19UK1A0514**
GOLLAPELLY RUTHVIK	**19UK1A0505**
DARAM SANGEETHA	**19UK1A0503**
MACHARLA VAMSHI SAI	**19UK1A0510**

Under the esteemed guidance of

Mrs. T.SRAVANTHI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal – 506005

2019 – 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005
2019 – 2023

CERTIFICATE OF COMPLETION
UG PROJECT PHASE-1

This is to certify that the UG Project Phase-1entitled **"PLANT DISEASE FERTILIZER RECOMMENDATION SYSTEM"** is being submitted by **V.BHAVANI (H.NO:19UK1A0514), G.RUTHVIK (H.NO:19UK1A0505),D.SANGEETHA (H.NO:19UK1A0503),M.VAMSHI SAI (H.NO:19UK1A0510)** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2022-23**, is a record of work carried out by them under the guidance and supervision.

Project Guide	**Head of the Department**
Mrs. T. SRAVANTHI	**DR. R. NAVEEN KUMAR**
(Assistant Professor)	(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carryout this UG Project Phase-I in the institute.

We express our heartful thanks to **Dr. R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-I.

We express our heartful thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-I and for their support in completing the UG Project Phase-I.

We express our heartful thanks to the guide, **Mrs. T.SRAVANTHI,** Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion Of this UG Project Phase-I.

Finally, We express our sincere thanks and gratitude to my family members, friends for theirencouragement and outpouring their knowledge and experience throughout the thesis.

V. BHAVANI	**(19UK1A0514)**
G.RUTHVIK	**(19UK1A0505)**
D.SANGEETHA	**(19UK1A0503)**
M.VAMSHI SAI	**(19UK1A0510)**

‘

ABSTRACT

The Plant Disease Fertilizer Recommendation System (PDFRS) is an intelligent agricultural tool designed to assist farmers and agronomists in making informed decisions regarding the application of fertilizers for the management of plant diseases. The system leverages advanced technologies such as machine learning and data analytics to analyze plant disease symptoms, assess nutrient deficiencies, and provide tailored fertilizer recommendations.

The PDFRS utilizes a comprehensive database of plant diseases, their symptoms, and corresponding nutrient deficiencies. By inputting the observed symptoms of a diseased plant, the system employs machine learning algorithms to identify potential diseases and their associated nutrient imbalances. Additionally, it considers environmental factors such as soil type, climate conditions, and crop type to provide more accurate recommendations.

TABLE OF CONTENT'S

LIST OF FIGURES

1. INTRODUCTION

1.1 OVER VIEW

Deep learning is a subset of deep learning that involves the use of artificial neural networks to 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.

Overall, deep learning techniques can be a powerful tool for developing plant disease and fertilizer recommendation systems that are accurate, efficient, and easy to use.

1.2 PURPOSE

The purpose of a plant disease and fertilizer recommendation system based on deep learning isto 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. PROBLEM STATEMENT

A plant disease prediction deep learning problem involves using data about plants and their environment to predict the likelihood that a plant will develop a specific disease. This type of problem can be approached using supervised learning, in which a machine learning model is trained on a labeled dataset that includes examples of plants with and without the disease in question. The model can then be used to make predictions about the likelihood that a new plant will develop the disease, based on its characteristics and the conditions it is grown in.

There are a number of factors that can influence the development of plant diseases, including the plant's genetics, its environment (e.g., temperature, humidity, sunlight), and the presence of pathogens (e.g., bacteria, viruses, fungi). To build a machine learning model for plant disease prediction, it is important to gather data on these factors as well as on the presence or absence of the disease in question. This data can be used to train and evaluate the model, with the goal of achieving high accuracy in predicting the likelihood of disease development.

3. LITERATURE SURVEY

3.1 EXISTING PROBLEM:

One existing problem with plant disease and fertilizer recommendation systems based on deeplearning 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.

Finally, there may be concerns about the interpretability and explain ability of the recommendations made by the system, as well as the potential for errors or unintended consequences if the system is not used properly.

3.2 PROPOSED SOLUTION:

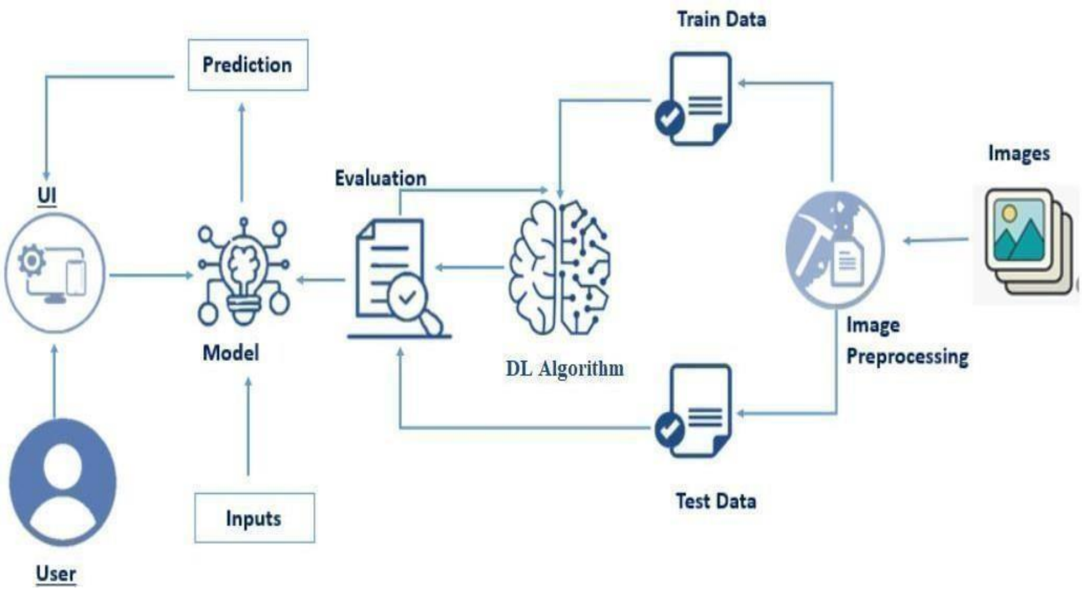
Data collection and curation to improve the accuracy and effectiveness of the system, it is important to collect and curate large amounts of high-quality data from a diverse range of sources. This may involve collaboration with multiple organizations or individuals, as well as efforts to standardize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

Interpretability and explain ability to increase the interpretability and explain ability of the system, it may be helpful to develop techniques for visualizing and explaining the decision-making process of the system, as well as to provide context and background information to users.

4. THEORITICAL ANALYSIS

4.1 BLOCK DIAGRAM



4.2 HARDWARE / SOFTWARE DESIGNING

The following is the Hardware required to complete this project:

- Internet connection to download and activate
- Administration access to install and run Anaconda Navigator
- Minimum 10GB free disk space
- Windows 8.1 or 10 (64-bit or 32-bit version) OR Cloud: Get started free, *Cloud account required.

Minimum System Requirements To run Office Excel 2013, your computer needs to meet the following minimum hardware requirements:

- 500 megahertz (MHz)
- 256 megabytes (MB) RAM
- 1.5 gigabytes (GB) available space
- 1024x768 or higher resolution monitor

The following are the software s required for the project:

- Jupyter Notebook
- Spyder

5. 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, weatherdata, 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 involvedesigning 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 explain ability of the model's recommendations.

Model deployment: If the model performs well in the evaluation phase, it could be deployedin 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.

Overall, the experimental investigation of a plant disease and fertilizer recommendation system based on deep learning would involve a combination of data analysis, machine learning techniques, and user testing in order to develop and evaluate the system.

6. DESIGN

6.1 FLOWCHART

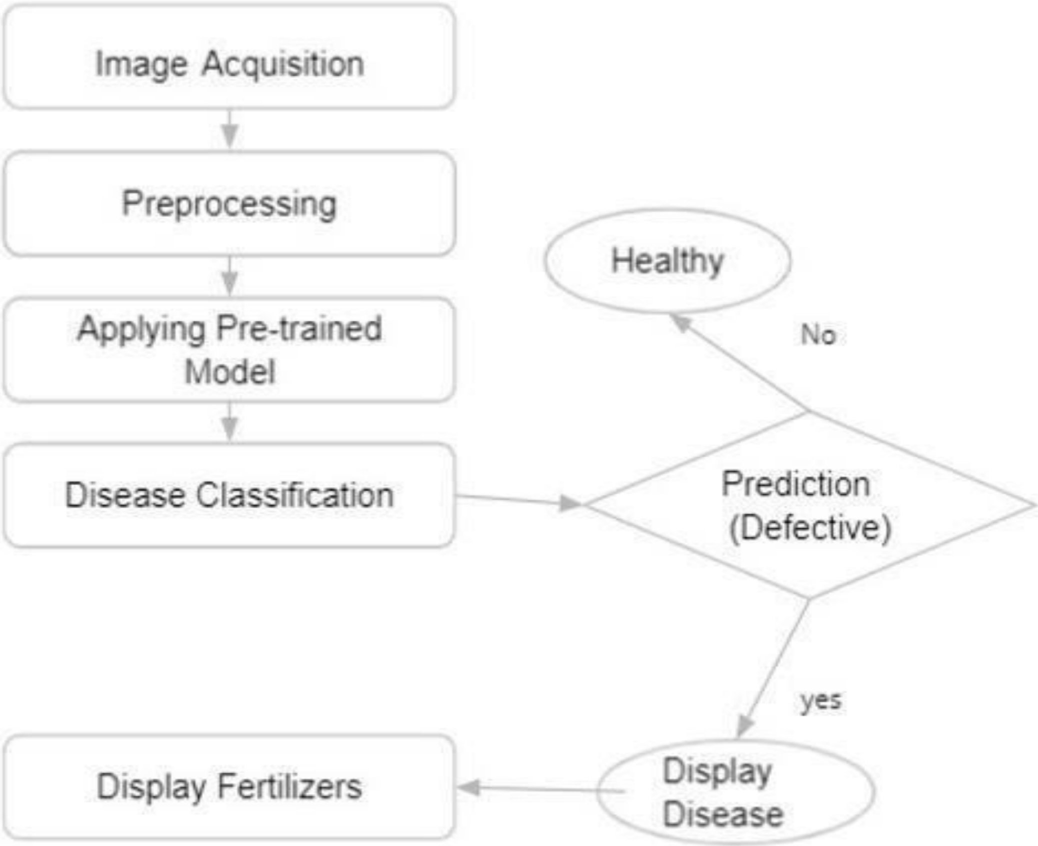


Fig. 2. Flowchart of Training model

7. CONCLUSION

In UG Project Phase-1, we have worked on problem statement, literature survey and also done the experiment analysis which are required for the project to move forward. In experimental analysis we have discussed about the machine learning concepts and models and explained the algorithm to be used in the project. We also discussed about the flowcharts, use case diagrams, decision tree and sequence diagrams which are used in the project. Based on the experimental analysis we have designed the model for the project. Entire designing part is involved in UG Project Phase-1

8. FUTURE SCOPE

UG Project Phase-2 is the extension of UG Project Phase-1. UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusions will be retrieved in thephase. We will also work on the applications, advantages, and disadvantages of the project in this phase. Future scope of the project will be also discussed in the UG Project Phase-2.

PLANT DISEASE FERTILIZER RECOMMENDATION SYSTEM

A UG PROJECT PHASE-2 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

VENUVANKA BHAVANI	**19UK1A0514**
GOLLAPELLY RUTHVIK	**19UK1A0505**
DARAM SANGEETHA	**19UK1A0503**
MACHARLA VAMSHI SAI	**19UK1A0510**

Under the esteemed guidance of

Mrs. T. SRAVANTHI

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTUH, Hyderabad)

Bollikunta, Warangal – 506005

2019– 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005
2019 – 2023

CERTIFICATE OF COMPLETION
UG PROJECT PHASE-2

This is to certify that the UG Project Phase-2 entitled **"PLANT DISEASE FERTILIZER RECOMMENDATION SYSTEM"** is being submitted by **V. BHAVANI (H.NO:19UK1A0514),G.RUTHVIK(H.NO:19UK1A0505),D.SANGEETHA(H.NO:19UK1A0503), M.VAMSHI SAI(19UK1A0510)** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2022-23**, is a record of work carried out by them under the guidance and supervision.

Project Guide	**Head of the Department**
Mrs. T.SRAVANTHI	**DR. R. NAVEEN KUMAR**
(Assistant Professor)	(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respectto our beloved **Dr. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-2 in the institute.

We express our heartful thanks to **Dr. R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and therebygiving us freedom to carry out the UG Project Phase-2.

We express our heartful thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-2 and for their support in completing the UG Project Phase-2.

We express our heartful thanks to the guide, **Mrs. T. SRAVANTHI,** Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this UG Project Phase-2.

Finally, We express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

V. BHAVANI	**(19UK1A0514)**
G.RUTHVIK	**(19UK1A0505)**
D.SANGEETHA	**(19UK1A0503)**
M.VAMSHI SAI	**(19UK1A0510)**

TABLE OF CONTENT'S

LIST OF FIGURES PG NO.

1. INTRODUCTION

In this project, two datasets name fruit dataset and vegetable dataset are collected. The collected datasets are trained and tested with deep learning neural network named Convolutional Neural Networks (CNN). First, the fruit dataset is trained and then tested with CNN. It has 6 classes and all the classes are trained and tested. Second, the vegetable dataset is trained and tested. The software used for training and testing of datasets is Python. All the Python codes are first written in Jupyter notebook supplied along with Anaconda Python and then the codes are tested in IBM cloud. Finally a web based framework is designed with help Flask a Python library. There are 2 html files are created in templates folder along with their associated files in static folder. The Python program 'app.py' used to interface with these two webpages is written in Spyder-Anaconda python and tested.

UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusions are retrieved in this phase. We will also work on the applications, advantages, and disadvantagesof the project in this phase. Future scope of the project will be also discussed in the UG Project Phase-2.

2. CODE SNIPPETS

2.1 MODEL CODE :

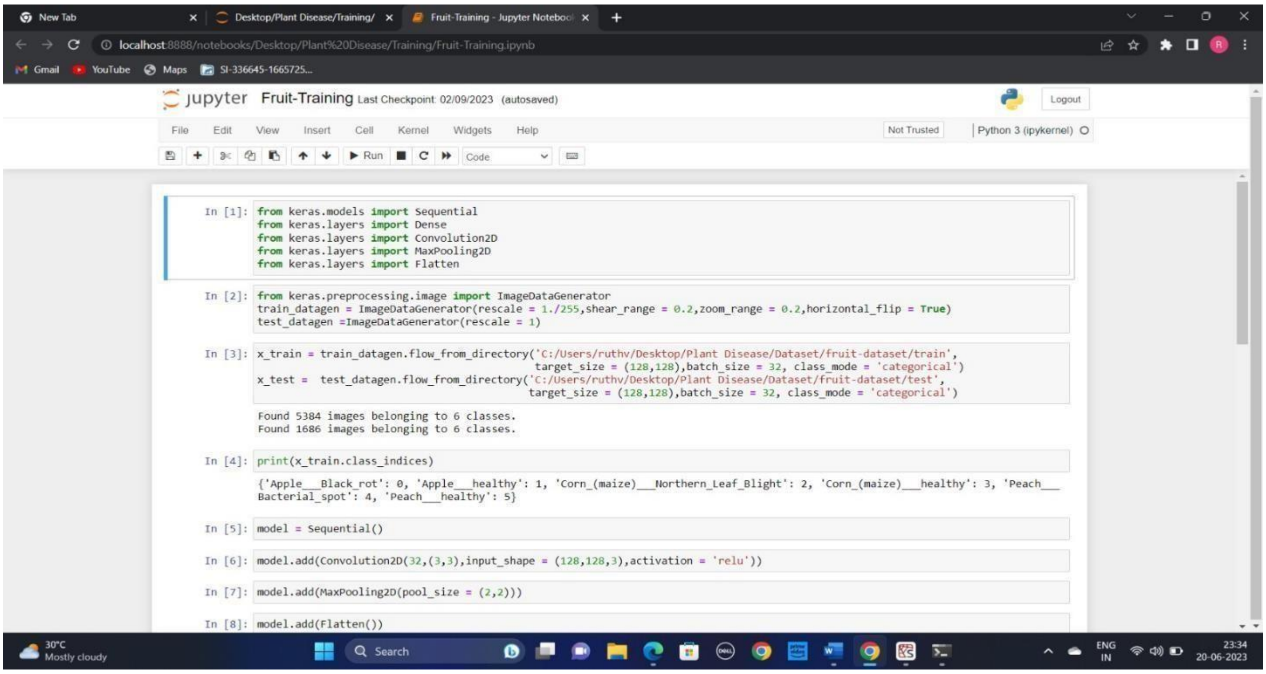


Figure 1: Import the libraries

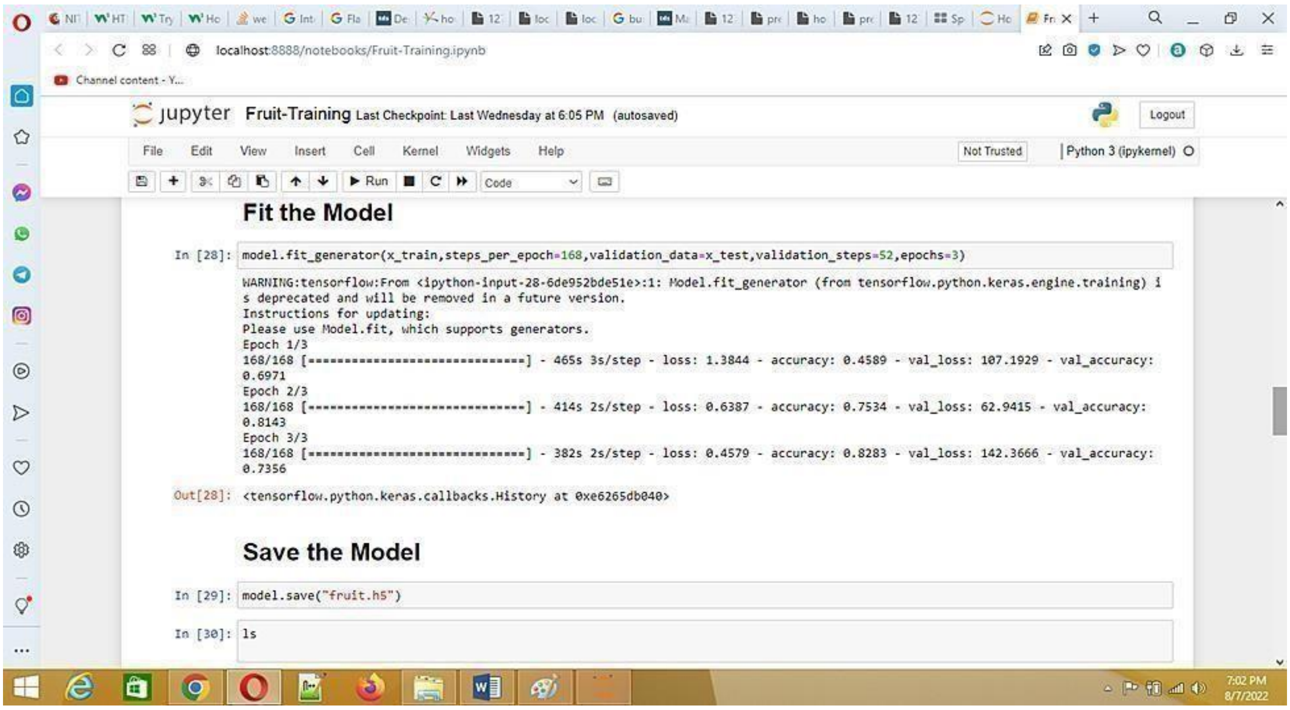


Figure 2 : Fit a model for Fruit dataset

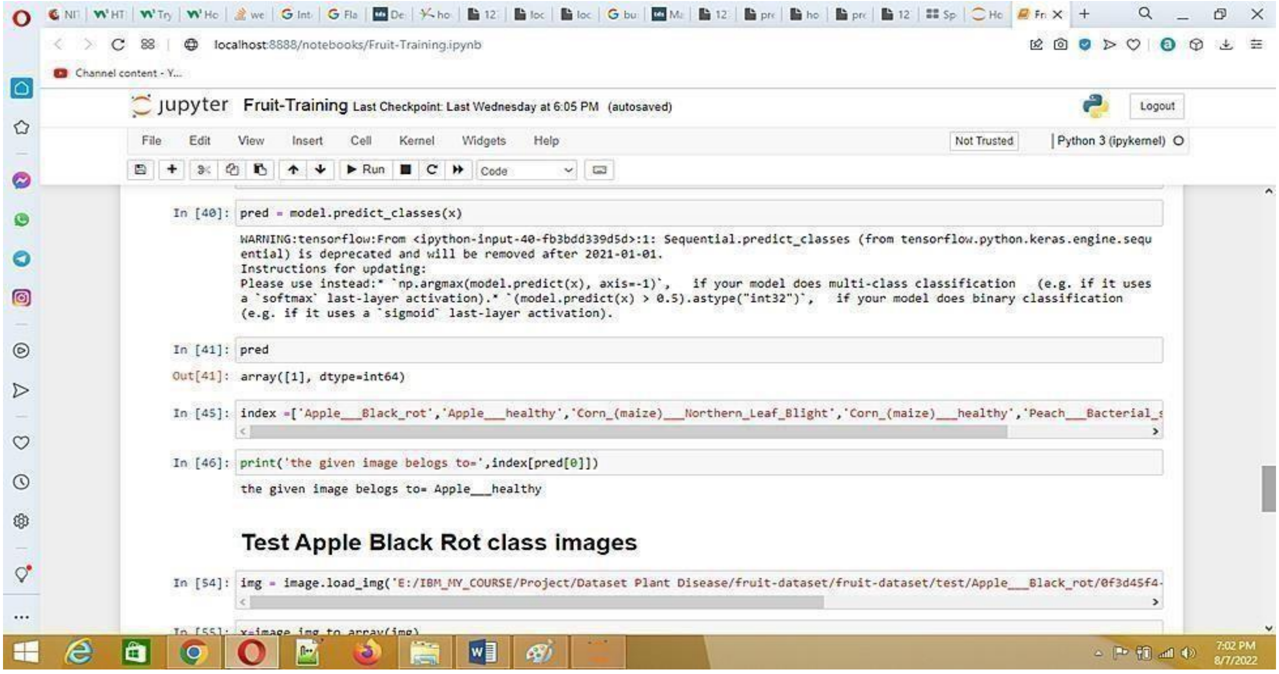


Figure 3:**Test the fruit dataset**

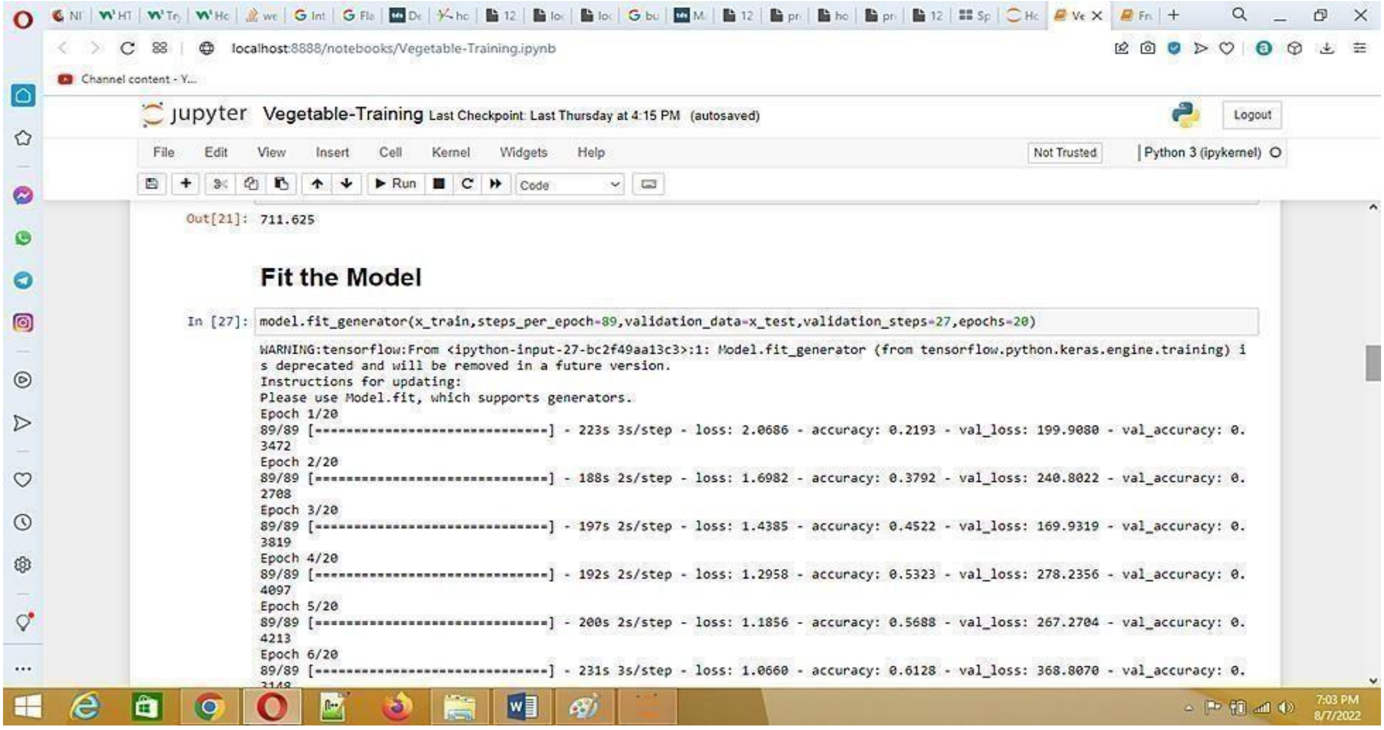


Figure 4: **Train the Vegetable dataset**

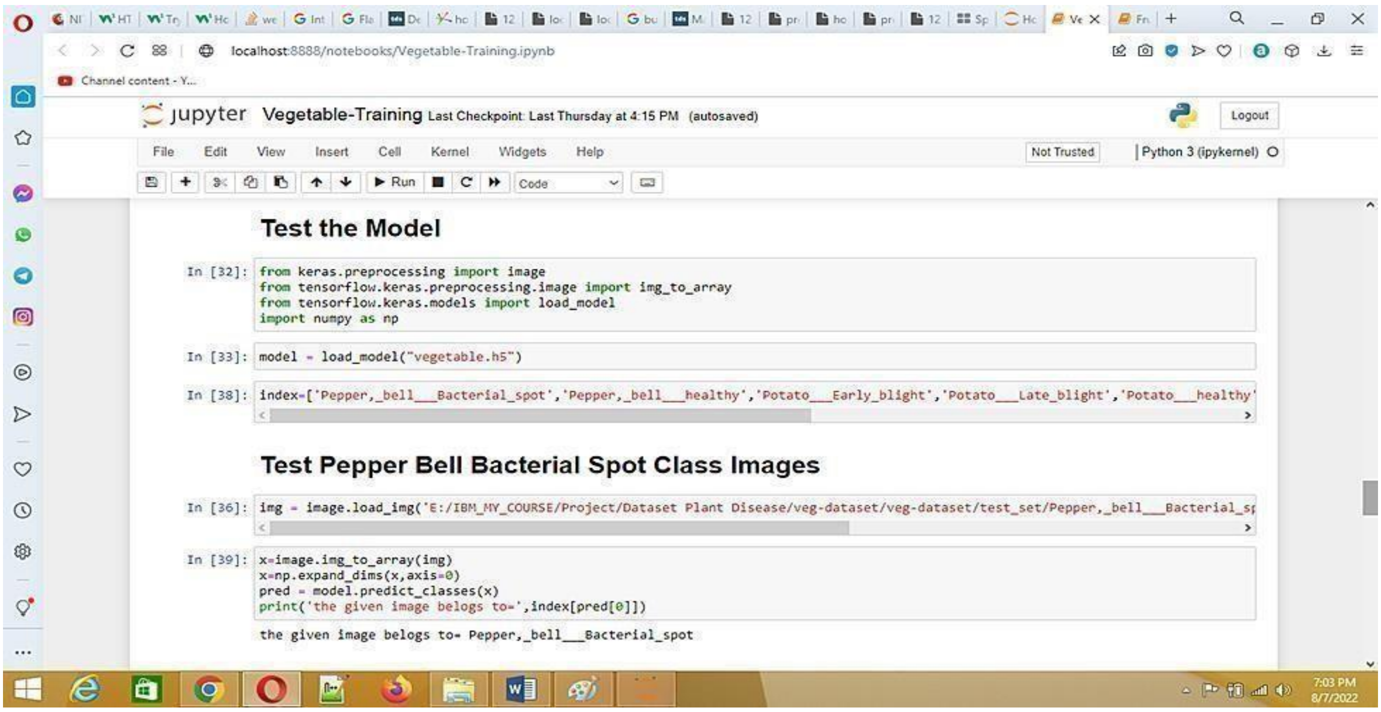


Figure 5: **Test vegetable dataset**

2.2 HTML CODE AND PYTHON CODE

1. app.py code:

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

@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['image']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        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_classes(x)
            print(preds)
            df=pd.read_excel('precautions - veg.xlsx')
            print(df.iloc[preds[0]]['caution'])
        else:
            preds = model1.predict_classes(x)

            df=pd.read_excel('precautions - fruits.xlsx')
            print(df.iloc[preds[0]]['caution'])
```


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

@app.route('/predict',methods=['POST'])
def predict():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['image']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        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_classes(x)
            print(preds)
            df=pd.read_excel('precautions - veg.xlsx')
            print(df.iloc[preds[0]]['caution'])
        else:
            preds = model1.predict_classes(x)

            df=pd.read_excel('precautions - fruits.xlsx')
            print(df.iloc[preds[0]]['caution'])
```

```
        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)
```

Figure 6 : .python code used for rendering all the HTML pages

2. home.html

```
s\ruthv\OneDrive\Desktop\Plant Disease\Flask\templates\home.html
app.py ×  final.css ×  main.js ×  home.html ×  predict.html ×

<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
<style>
.header {
             top:0;
             margin:0px;
             left: 0px;
             right: 0px;
             position: fixed;
             background-color: #28272c;
             color: white;
             box-shadow: 0px 8px 4px grey;
             overflow: hidden;
             padding-left:20px;
             font-family: 'Josefin Sans';
             font-size: 2vw;
             width: 100%;
             height:8%;
             text-align: center;
         }
         .topnav {
  overflow: hidden;
  background-color: #333;
}

.topnav-right a {
```

app.py × final.css × main.js × home.html × predict.html ×

```css
    float: left;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    font-size: 18px;
  }

  .topnav-right a:hover {
    background-color: #ddd;
    color: black;
  }

  .topnav-right a.active {
    background-color: #565961;
    color: white;
  }

  .topnav-right {
    float: right;
    padding-right:100px;
  }

  body {

    background-color:#ffffff;
    background-repeat: no-repeat;
    background-size:cover;
    background-position: 0px 0px;
    }
    .button {
    background-color: #28272c;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 16px;
    border-radius: 12px;
```

app.py × final.css × main.js × home.html × predict.html ×

```css
    border-radius: 12px;
  }
  .button:hover {
    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);
  }
  form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

  input[type=text], input[type=password] {
    width: 100%;
    padding: 12px 20px;
    display: inline-block;
    margin-bottom:18px;
    border: 1px solid #ccc;
    box-sizing: border-box;
  }

  button {
    background-color: #28272c;
    color: white;
    padding: 14px 20px;
    margin-bottom:8px;
    border: none;
    cursor: pointer;
    width: 15%;
    border-radius:4px;
  }

  button:hover {
    opacity: 0.8;
  }

  .cancelbtn {
    width: auto;
    padding: 10px 18px;
    background-color: #f44336;
  }

  .imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;
```

```
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
</style>
</head>

<body style="font-family:'Times New Roman', Times, serif;background-color:#C2C5A8;">

<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Plant Disease Prediction</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a href="{{ url_for('prediction')}}">Predict</a>
  </div>
</div>

<div style="background-color:#ffffff;">
<div style="width:60%;float:left;">
<div style="font-size:50px;font-family:Montserrat;padding-left:20px;text-align:center;padding-top:10%;">
<b>Detect if your plant<br> is infected!!</b></div><br>
<div style="font-size:20px;font-family:Montserrat;padding-left:70px;padding-right:30px;text-align:justify;">Agriculture is one of the major sectors worls wide. Over the years
</div>
</div>
<div style="width:40%;float:right;"><br><br>
<img src="{{url_for('static',filename='images/12456.png')}}" style="max-height:100%;max-width:100%;">

</div>
</div>
```

```
<b>Detect if your plant<br> is infected!!</b></div><br>
<div style="font-size:20px;font-family:Montserrat;padding-left:70px;padding-right:30px;text-align:justify;">Agriculture is one of the major sectors worls wide. Over the years
</div>
</div>
<div style="width:40%;float:right;"><br><br>
<img src="{{url_for('static',filename='images/12456.png')}}" style="max-height:100%;max-width:100%;">

</div>
</div>

<div class="home">

<br>

</div>

<script>
var slideIndex = 0;
showSlides();

function showSlides() {
  var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
</body>
</html>
```

Figure 7: home.html page is the code for home page of our Web Applications

3. predict.html

app.py ✕ final.css ✕ main.js ✕ home.html ✕ predict.html ✕

```html
<!DOCTYPE html>
<html >

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<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"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
<link href="{{ url_for('static', filename='css/final.css') }}" rel="stylesheet">
<style>
.header {
            top:0;
            margin:0px;
            left: 0px;
            right: 0px;
            position: fixed;
            background-color: #28272c;
            color: white;
            box-shadow: 0px 8px 4px grey;
            overflow: hidden;
            padding-left:20px;
            font-family: 'Josefin Sans';
            font-size: 2vw;
            width: 100%;
            height:8%;
            text-align: center;
        }
        .topnav {
  overflow: hidden;
  background-color: #333;
```

app.py ✕ final.css ✕ main.js ✕ home.html ✕ predict.html ✕

```css
  background-color: #333;
}

.topnav-right a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

.topnav-right a:hover {
  background-color: #ddd;
  color: black;
}

.topnav-right a.active {
  background-color: #565961;
  color: white;
}

.topnav-right {
  float: right;
  padding-right:100px;
}

.login{
margin-top:-70px;
}
body {

  background-color:#ffffff;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
.login{
    margin-top:100px;
}
```

```
\ruthv\OneDrive\Desktop\Plant Disease\Flask\templates\predict.html
pp.py  final.css  main.js  home.html  predict.html
}

.container {
  margin-top:40px;
  padding: 16px;
}
select {
    width: 100%;
    margin-bottom: 10px;
    background: rgba(255,255,255,255);
    border: none;
    outline: none;
    padding: 10px;
    font-size: 13px;
    color: #000000;
    text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
    border: 1px solid rgba(0,0,0,0.3);
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px rgba(255,255,255,0.2);
    -webkit-transition: box-shadow .5s ease;
    -moz-transition: box-shadow .5s ease;
    -o-transition: box-shadow .5s ease;
    -ms-transition: box-shadow .5s ease;
    transition: box-shadow .5s ease;
}


</style>
</head>

<body style="font-family:Montserrat;overflow:scroll;">

<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Plant Disease Prediction</div>
  <div class="topnav-right" style="padding-top:0.5%;">


  </div>
</div>
<div class="container">
```

```
hv\OneDrive\Desktop\Plant Disease\Flask\templates\predict.html
py  final.css  main.js  home.html  predict.html
<div class="container">
        <div id="content" style="margin-top:2em">
        <div class="container">
          <div class="row">
            <div class="col-sm-6 bd" >

              <br>
                <img src="{{url_for('static',filename='images/789.jpg')}}" style="height:450px;width:550px"class="img-rounded" alt="Gesture">
            </div>
            <div class="col-sm-6">
                <div>
                    <h4>Drop in the image to get the prediction  </h4>
            <form action = "" id="upload-file" method="post" enctype="multipart/form-data">
                <select name="plant">

                      <option value="select" selected>Select plant type</option>
                      <option value="fruit">Fruit</option>
                      <option value="vegetable">Vegetable</option>
        </select><br>
                <label for="imageUpload" class="upload-label" style="background: #28272c;">
                    Choose...
                </label>
                <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
            </form>


            <div class="image-section" style="display:none;">
                <div class="img-preview">
                    <div id="imagePreview">
                    </div>
                </div>
                <div>
                    <button type="button" class="btn btn-info btn-lg " id="btn-predict" style="background: #28272c;">Predict!</button>
                </div>
            </div>

            <div class="loader" style="display:none;"></div>

            <h3>
                <span id="result" style="font-size:17px; "> </span>
```

```html
                        <option value="fruit">Fruit</option>
                        <option value="vegetable">Vegetable</option>
          </select><br>
                  <label for="imageUpload" class="upload-label" style="background: #28272c;">
                      Choose...
                  </label>
                  <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
              </form>


              <div class="image-section" style="display:none;">
                  <div class="img-preview">
                      <div id="imagePreview">
                      </div>
                  </div>
                  <div>
                      <button type="button" class="btn btn-info btn-lg " id="btn-predict" style="background: #28272c;">Predict!</button>
                  </div>
              </div>

              <div class="loader" style="display:none;"></div>

              <h3>
                  <span id="result" style="font-size:17px; "> </span>
              </h3>

          </div>
              </div>

            </div>
          </div>
          </div>
      </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>
</html>
```

Figure 8: predict.html is the page which displays that the plant not having disease

3.CONCLUSION

Figure 9: Home page(which gives introduction to disease prediction)

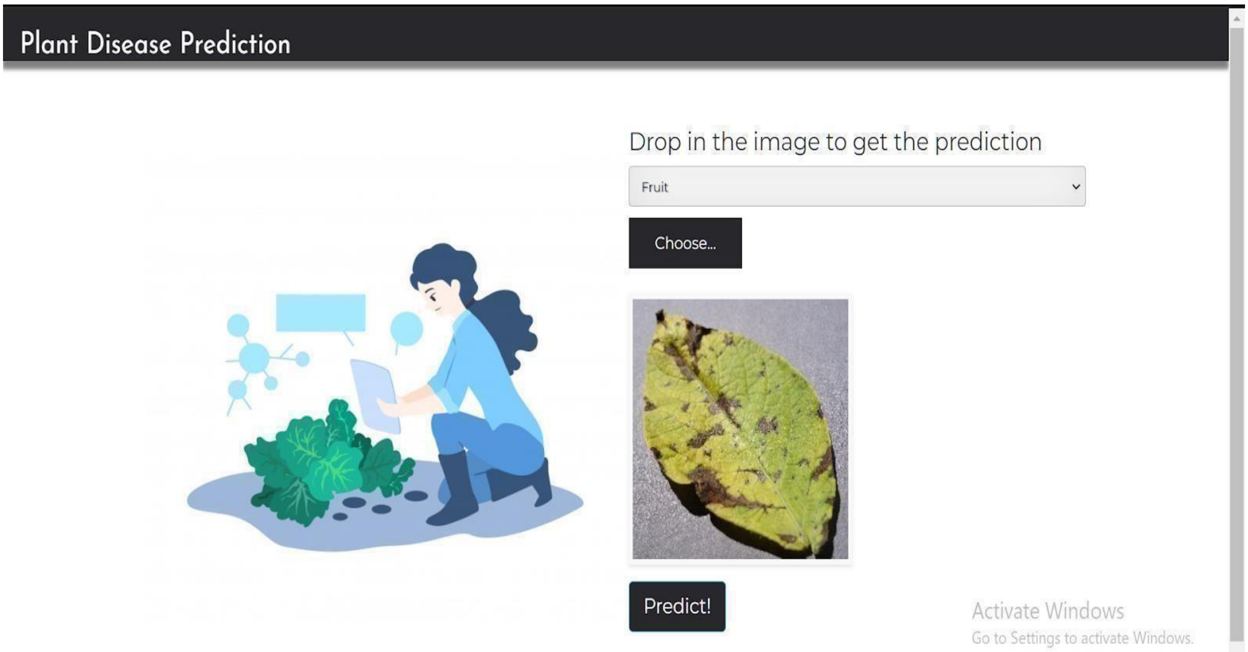


Figure 10: Input page(Which inputs from users)

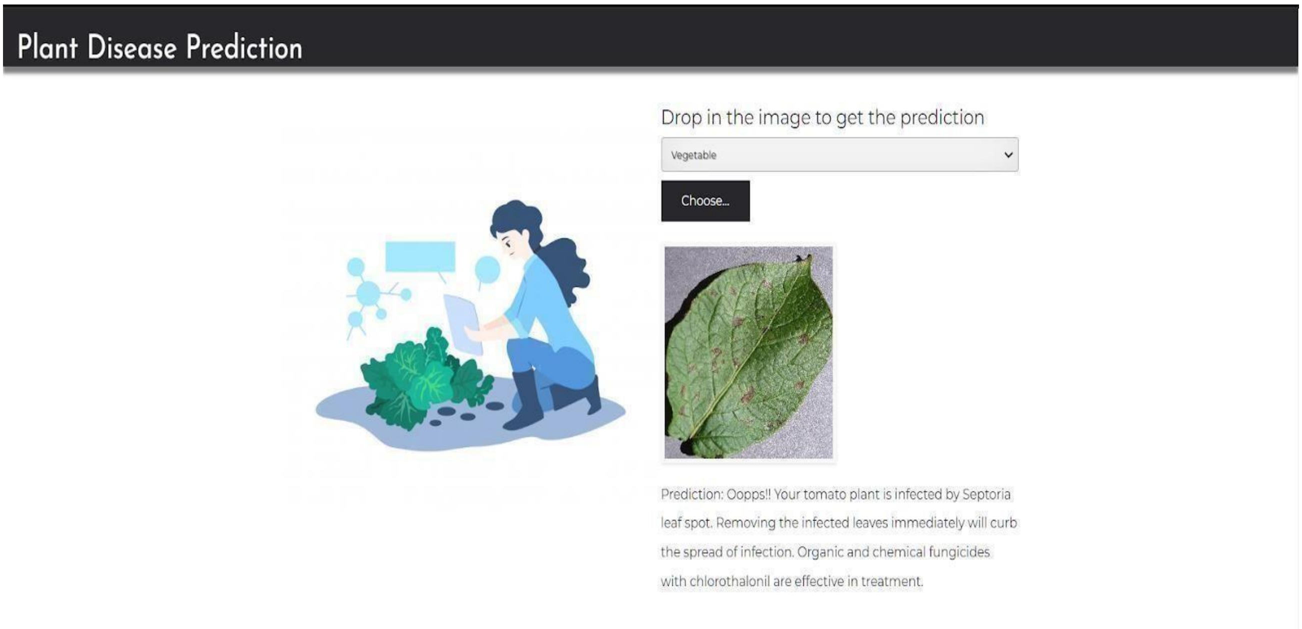


Figure 11: Out page(displays that the leaf is infected)

4.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 respondto questions about plant diseases and fertilization in natural language, allowing farmers orgardeners 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.

5.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 particularlyuseful 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.

6.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.

7.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 explain ability: 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.

8.BIBILOGRAPHY

[1]. P.Pandi Selvi,P. Poornima, "Soil Based Fertilizer Recommendation System for Crop Disease Prediction System", International Journal of Engineering Trends and Applications(IJETA)– Volume 8 Issue 2,Mar-Apr 2021

[2]. R Indumathi.N .Saagari.; V Thejuswini.;R Swarnareka.," Leaf Disease Detection and Fertilizer Suggestion ", IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 29-30 March 2019, DOI:10.1109/ICSCAN.2019.8878781.

[3]. R Indumathi.; N Saagari.; V Thejuswini.; R Swarnareka.," Leaf Disease Detection and Fertilizer Suggestion", IEEE International Conference on System, Computation, 9 Automation and Networking (ICSCAN), 29-30 March2019, DOI: 10.1109/ICSCAN.2019.8878781.

[4]. P. Pandi Selvi, P. Poornima, "Soil Based Fertilizer Recommendation System for Crop Disease Prediction System", International Journal of Engineering Trends and Applications (IJETA) – Volume 8 Issue 2, Mar-Apr 2021 .

[5]. H Shiva reddy, Ganesh hedge, Prof. DR Chinnaya3, "IoT based Leaf Disease Detection and Fertilizer Recommendation", International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 11, Nov 2019, e-ISSN: 2395- 005

9.HELP FILE

PROJECT EXECUTION:

STEP-1: Go to **Start,** search and launch **ANACONDA NAVIGATOR.**

STEP-2: After launching of **ANACONDA NAVIGATOR,** launch **JUPYTER NOTEBOOK.**

STEP-3: Open **"Major project code" IPYNB file.**

STEP-4: Then run all the cells.

STEP-5: All the **data preprocessing**, **training and testing**, **model building**, **accuracy** of the model can be showcased.

STEP-6: And a pickle file will be generated.

STEP-7: Create a Folder named **FLASK** on the **DESKTOP.** Extract the pickle file into this Flask Folder.

STEP-8: Extract all the html files (home.html, index.html, chance.html, nochance.html) and python file(app.py) into the **FLASK Folder.**

STEP-9: Then go back to **ANACONDA NAVIGATOR** and the launch the **SPYDER.**

STEP-10: After launching Spyder, give the path of **FLASK FOLDER** which you have created on the DESKTOP.

STEP-11: Open all the app.py and html files present in the Flask Folder.

STEP-12: After running of the app.py, open **ANACONDA PROMPT** and follow the below steps:

cd File Path→click enter

python app.py→click enter (We could see running of files).

STEP-13: Then open **BROWSER,** at the URL area type

http://127.0.0.1:5000/**.**

STEP-14: Home page of the project will be displayed.

STEP-15: Click on **—Go to Predict".** Directly it will be navigated to index page.

STEP-16:A index page will be displayed where the user needs to give the inputs and then click on **—Predict"**. Output will be generated whether the plant is infected or not.