

**A Project Based Learning**  
**Report On**  
**Crop Monitoring**  
Submitted Towards The  
Partial Fulfillment of The Requirements of  
**B.Tech Robotics and Automation**  
**Sem VI**  
For the Subject  
**Artificial Intelligence And Neural**  
**Networks**  
**Academic Year: 2022-23**



Bharati Vidyapeeth's  
(Deemed to be University)  
College of Engineering, Pune 411043  
*Ranked 96<sup>th</sup> by NIRF*  
*Accredited with "A" grade by NAAC*

**Submitted by -**

<b>Name of student</b>	<b>Roll No.</b>	<b>PRN No.</b>
Sudarshan Dhage	8	2014111271
Rohit Patil	24	2014111289
Shivani	30	2014111295
Mohini Wable	34	2014111299
Rohit Sonawane	40	2014111306

**Faculty Supervisor**

Prof. Akshay Harale

**Bharati Vidyapeeth's**  
(Deemed to be University)  
**College of Engineering, Pune 411043**  
*Ranked 96<sup>th</sup> by NIRF*  
*Accredited with "A" grade by NAAC*

## CERTIFICATE

This is to certify that the project entitled "**CROP MONITORING**" is a bonafide work carried out by the following students and it is submitted to the Bharati Vidyapeeth Deemed to be University College of Engineering, Pune for the partial fulfillment of the requirement for the of Semester VI Robotics and Automation Bachelor's of Technology Degree for the subject *Artificial Intelligence and neural network* Project Based Learning (PBL) in the of Semester VI Robotics and Automation Engineering .

Student Name	PRN No.	Signature
Sudarshan Dhage.	2014111271	
Rohit Patil	2014111289	
Shivani	2014111295	
Mohini Wable.	2014111299	
Rohit Sonawane	2014111306	

**Prof. Akshay Harale**  
(Faculty Supervisor)

**Dr. K.B.Sutar**  
(Head of Department)

## DECLARATION

We, hereby declare that the project titled“**CROP MONITORING**” being submitted by us towards the partial fulfillment of Bachelor of Technology, is a project-based learning work carried by us is our own work.

Date:

<b>Student Name</b>	<b>PRN No.</b>	<b>Signature</b>
Sudarshan Dhage	2014111271	
Rohit Patil	2014111289	
Shivani	2014111295	
Mohini Wable.	2014111299	
Rohit Sonawane	2014111306	

## ACKNOWLEDGEMENT

It gives us immense pleasure to present this report on “**Crop Monitoring**” carried out at BVPCOE, PUNE in accordance with the prescribed syllabus of university. We express our heartfelt gratitude to those who directly and indirectly contributed towards the completion of this project. We would like to thank **Prof. Akshay Harale** for the valuable guidance and continuous support.

## ABSTRACT

Crop monitoring is a critical task in modern agriculture, as it enables farmers to track the health of their crops and identify potential issues that can impact crop yield and quality. One of the major challenges in crop monitoring is detecting diseases early enough to prevent significant crop losses. Traditionally, crop diseases are identified through visual inspections, which can be time-consuming and often rely on the experience of the observer. However, with recent advancements in technology, there are now more efficient and accurate ways to detect diseases in crops.

In recent years, researchers have explored the use of advanced technologies, such as remote sensing, machine learning, and computer vision, to improve disease detection in crops. These technologies have the potential to significantly improve crop management and reduce crop losses.

One of the most commonly used techniques in crop monitoring is remote sensing. Remote sensing involves collecting data from sensors that are not in direct contact with the object being observed. In agriculture, remote sensing can be used to collect data on crop health, such as temperature, moisture, and nutrient levels. Hyperspectral imaging is another type of remote sensing technology that can detect subtle changes in the reflectance spectrum of plants, which can indicate the presence of disease.

Another technique that has shown promise in crop monitoring is the use of unmanned aerial vehicles (UAVs). UAVs can collect high-resolution images of crops, which can be used to monitor crop health and identify disease outbreaks. These images can be analyzed using machine learning algorithms to identify patterns and classify crops based on disease symptoms.

Machine learning algorithms, such as neural networks and support vector machines, are also being used to analyze data collected from remote sensors and UAVs. These algorithms can learn to identify patterns and classify crops based on disease symptoms. This can significantly improve the accuracy and efficiency of disease detection in crops.

In conclusion, the use of advanced technologies, such as remote sensing, machine learning, and computer vision, holds great promise in improving disease detection in crops and enhancing agricultural productivity. These techniques can enable farmers to detect diseases early, take appropriate action, and reduce crop losses. The continued development and refinement of these technologies will play a critical role in the future of agriculture.

# CONTENTS

	Pg No.
CERTIFICATE	2
DECLARATION	3
ACKNOWLEDGEMENT	4
ABSTRACT	5
CONTENTS	6
INTRODUCTION	7
EXISTING TECHNOLOGY	9
METHODOLOGY	10
DEVELOPMENT OF PROJECT	11
RESULT	40
CONCLUSION	42

## INTRODUCTION

Crop monitoring is the process of collecting and analyzing data related to crop growth, health, and environmental conditions. By introducing crop monitoring into disease detection, students can gain practical experience in identifying early signs of crop diseases and pests, which can help prevent their spread and minimize economic losses. In addition, incorporating crop monitoring into disease detection can provide an interdisciplinary learning experience that can help students develop a holistic understanding of the challenges faced in agriculture. This can include knowledge and skills from multiple disciplines such as agriculture, biology, statistics, and data analysis. Students can develop problem-solving skills and become exposed to innovative approaches to address challenges in agriculture. Overall, the introduction of crop monitoring in disease detection can help students develop a deeper appreciation for the importance of disease prevention and its impact on food production and sustainability.

## PROBLEM STATEMENT

Problem statements of crop monitoring in disease detection are statements that describe the challenges and issues related to the use of crop monitoring techniques for detecting crop diseases and pests. These statements highlight areas where improvements are needed to enhance the effectiveness of crop monitoring and disease detection strategies, leading to better management of agricultural pests and diseases. Identifying and defining problem statements is a crucial step in developing solutions to address these challenges, as it allows for a clear understanding of the issues at hand and helps to guide the development of effective solutions.

### **OBJECTIVE:**

Crop monitoring in disease detection has several objectives that are crucial for ensuring sustainable and profitable agricultural practices. One of the primary objectives is to achieve early detection and diagnosis of crop diseases and pests. By monitoring crops regularly, agriculture professionals can detect and diagnose diseases and pests early, which can help prevent their spread and minimize economic losses for farmers. Early detection also enables farmers to implement appropriate management strategies promptly.

Another important objective of crop monitoring in disease detection is to develop effective management strategies for crop diseases and pests. Monitoring crops can help identify the specific diseases and pests affecting the crops and provide insights into their life cycles, behavior, and population dynamics. Based on this information, agriculture professionals can develop and implement effective management strategies, such as integrated pest management (IPM) techniques, which combine various pest control methods to achieve sustainable and cost-effective pest management.

Optimizing resource use is another objective of crop monitoring in disease detection. By monitoring crop growth and health, farmers can adjust their use of resources such as water, fertilizer, and pesticides based on the specific needs of their crops. This can help optimize the use of resources, reduce environmental impacts, and improve crop yields.

Yield optimization is also a key objective of crop monitoring in disease detection. By identifying and addressing crop diseases and pests early, agriculture professionals can help optimize crop yields. This can have a positive impact on the economy by increasing the quantity and quality of crops produced.

Finally, an objective of crop monitoring in disease detection is to foster innovation and technological development. With the development of new technologies and techniques such as drones and remote sensing, crop monitoring is an area that is rapidly evolving. By staying up-to-date with these innovative approaches and adopting new technologies, agriculture professionals can help address current and future challenges in agriculture.



## EXISTING TECHNOLOGY

Crop monitoring technology is rapidly advancing, and several innovative tools and techniques are now available for detecting crop diseases and pests. Here are some examples of existing technologies for crop monitoring in disease detection:

**Remote sensing:** Remote sensing involves the use of sensors mounted on aircraft, satellites, or drones to capture images of crops. These images are processed to detect changes in vegetation that may indicate the presence of crop diseases or pests. Remote sensing technology can cover large areas quickly, making it ideal for monitoring crops on a large scale.

**Drones:** Drones are unmanned aerial vehicles that can be equipped with sensors and cameras to capture high-resolution images of crops. These images can be processed to identify changes in vegetation that may indicate the presence of crop diseases or pests. Drones are particularly useful for monitoring crops in areas that are difficult to access, such as steep terrain or densely populated areas.

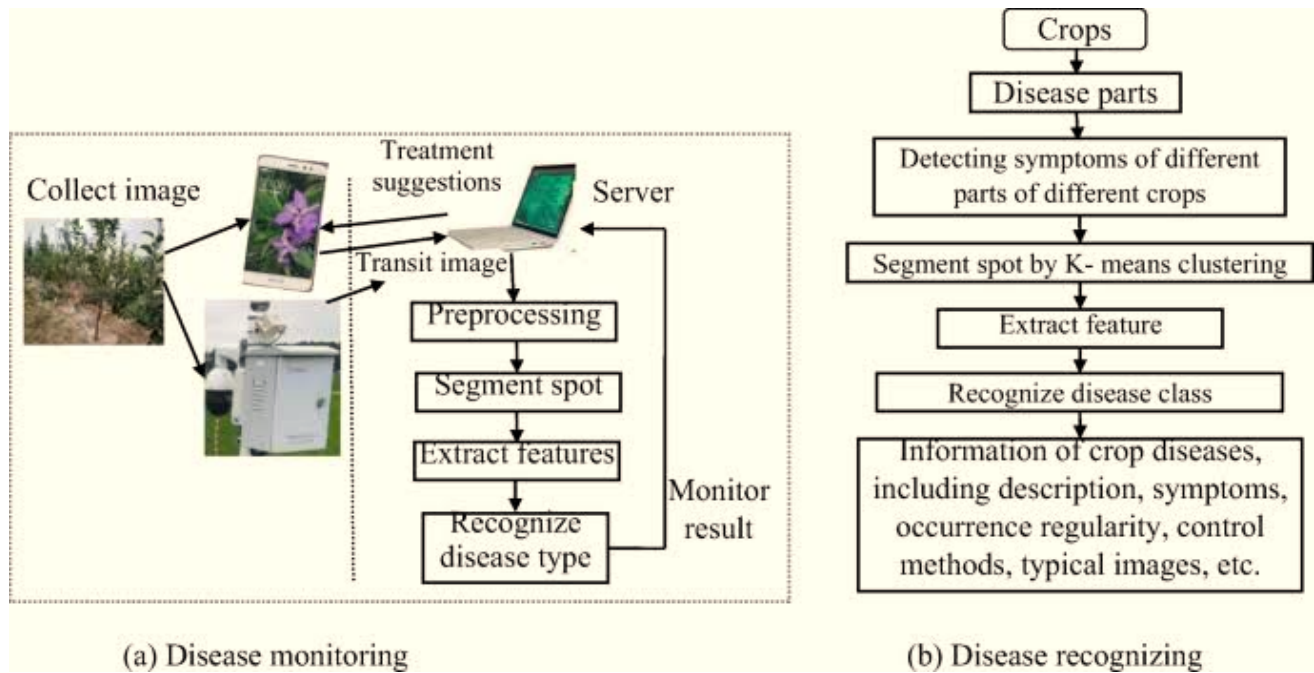
**Smartphone apps:** Several smartphone apps are now available that can help farmers and agriculture professionals identify crop diseases and pests. These apps use image recognition software to analyze pictures of crops and provide information on potential diseases or pests affecting the crop. The apps can also provide recommendations on appropriate management strategies.

**Weather stations:** Weather stations can be used to monitor environmental conditions, such as temperature and humidity, which can influence the development of crop diseases and pests. By tracking these conditions, farmers can implement appropriate management strategies to prevent the spread of diseases and pests.

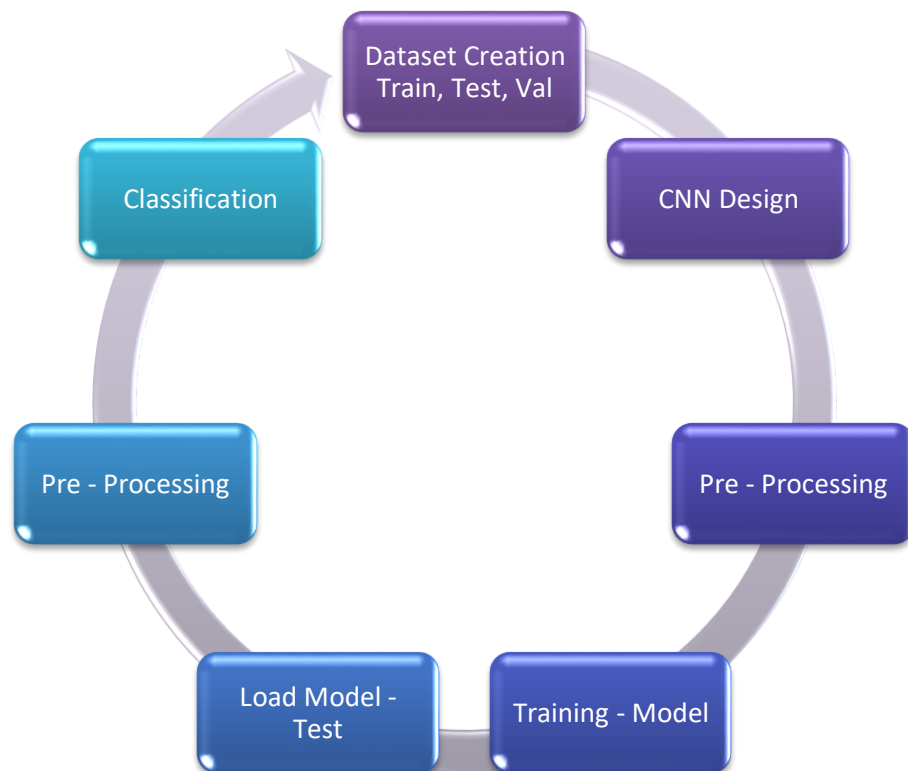
**Data analytics:** Data analytics involves the use of algorithms and statistical models to analyze large datasets. In crop monitoring, data analytics can be used to identify patterns and trends in crop growth and health, which can help identify potential disease or pest issues. The data can also be used to develop predictive models that can forecast disease or pest outbreaks.

In conclusion, there are several existing technologies for crop monitoring in disease detection, including remote sensing, drones, smartphone apps, weather stations, and data analytics. These tools and techniques can help agriculture professionals detect crop diseases and pests early, develop effective management strategies, and optimize resource use, resulting in sustainable and profitable agricultural practices.

## METHODOLOGY



### Process of Crop Monitoring



### Working of Module

## DEVELOPMENT OF PROJECT

### TRAINING MODULE CODE:

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

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

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf
tf.compat.v1.disable_eager_execution()
import matplotlib.pyplot as plt
import numpy as np
import os

#basic cnn
# Initialising the CNN
classifier = Sequential()

# Step 1 - Convolution
classifier.add(Conv2D(32, (3, 3), input_shape = (128, 128, 3), activation = 'relu'))

# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Adding a second convolutional layer
classifier.add(Conv2D(32, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Step 3 - Flattening
classifier.add(Flatten())

# Step 4 - Full connection
classifier.add(Dense(units = 128, activation = 'relu'))
classifier.add(Dense(units = 10, activation = 'sigmoid'))

# Compiling the CNN
```

```

classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])

train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2,
    horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('C:/Users/Sudarshan/Downloads/Plant-Leaf-
    Disease-Prediction-main/Plant-Leaf-Disease-Prediction-main/Dataset/train', # relative path from
    working directoy
        target_size = (128, 128),
        batch_size = 6, class_mode = 'categorical')
valid_set = test_datagen.flow_from_directory('C:/Users/Sudarshan/Downloads/Plant-Leaf-Disease-
    Prediction-main/Plant-Leaf-Disease-Prediction-main/Dataset/val', # relative path from working
    directoy
        target_size = (128, 128),
        batch_size = 3, class_mode = 'categorical')

labels = (training_set.class_indices)
print(labels)

classifier.fit_generator(training_set,
    steps_per_epoch = 20,
    epochs = 50,
    validation_data=valid_set

)

classifier_json=classifier.to_json()
with open("model1.json", "w") as json_file:
    json_file.write(classifier_json)
# serialize weights to HDF5
classifier.save_weights("my_model_weights.h5")
classifier.save("model.h5")
print("Saved model to disk")

```

## LEAF DISEASE DETECTION CODE:

```
from flask import Flask, render_template, request

import numpy as np
import os

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

filepath = "C:/Users/Sudarshan/Downloads/Plant-Leaf-Disease-Prediction-main/Plant-Leaf-Disease-
Prediction-main/model.h5"
model = load_model(filepath)
print(model)

print("Model Loaded Successfully")

def pred_tomato_dieas(tomato_plant):
    test_image = load_img(tomato_plant, target_size = (128, 128)) # 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 dimention 3D to 4D

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

    pred = np.argmax(result, axis=1)
    print(pred)
    if pred==0:
        return "Tomato - Bacteria Spot Disease", 'Tomato-Bacteria Spot.html'

    elif pred==1:
        return "Tomato - Early Blight Disease", 'Tomato-Early_Blight.html'

    elif pred==2:
        return "Tomato - Healthy and Fresh", 'Tomato-Healthy.html'

    elif pred==3:
        return "Tomato - Late Blight Disease", 'Tomato - Late_blight.html'

    elif pred==4:
        return "Tomato - Leaf Mold Disease", 'Tomato - Leaf_Mold.html'

    elif pred==5:
        return "Tomato - Septoria Leaf Spot Disease", 'Tomato - Septoria_leaf_spot.html'

    elif pred==6:
        return "Tomato - Target Spot Disease", 'Tomato - Target_Spot.html'
```

```

elif pred==7:
    return "Tomato - Tomoato Yellow Leaf Curl Virus Disease", 'Tomato -
Tomato_Yellow_Leaf_Curl_Virus.html'
elif pred==8:
    return "Tomato - Tomato Mosaic Virus Disease", 'Tomato - Tomato_mosaic_virus.html'

elif pred==9:
    return "Tomato - Two Spotted Spider Mite Disease", 'Tomato - Two-spotted_spider_mite.html'


# 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('C:/Users/Sudarshan/Downloads/Plant-Leaf-Disease-Prediction-
main/Plant-Leaf-Disease-Prediction-main/static/upload', filename)
        file.save(file_path)

        print("@@ Predicting class.....")
        pred, output_page = pred_tomato_dieas(tomato_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,port=8080)

```

## HOME PAGE FOR LEAF PREDECTION HTML CODE:

```
<html>
<head>
<style>

body {
  background-image: {{url_for('static',filename = 'images/Background.jpg')}};
  background-repeat: no-repeat;
  background-size: auto;
}
* {
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;

}

form {
  display: flex;
  height: 85vh;
  justify-content: center;
  align-items: center;
  margin-top: -150px;
  width: 60%;
  text-align: center;
  margin-left: 300px;

}

.details h2 {
  position: relative;
  top: 100px;
  margin: auto;
  color: rgb(18, 231, 231);
  font-size: 3rem;
}

label:hover {
  transform: scale(1.03);
}

.details h2 {
```

```

/* margin-bottom: 300px; */
position: relative;
top: 100px;
margin: auto;
color: rgb(18, 231, 231);
font-size: 3rem;
}

.details h1 {
  color: white;
  padding: 20px;
  border-radius: 15px;
  background-color: rgb(8, 8, 8);
}

.upload {

  font-size: 20px;
  background-color: rgb(255, 252, 252);
  border-radius: 20px;
  outline: none;
  width: 500px;
  color: rgb(0, 0, 0);
  border: 3px solid rgb(45, 47, 49);
}

::-webkit-file-upload-button {
  color: rgb(255, 252, 252);
  padding: 20px;
  border: 2px solid rgb(15, 176, 229);
  background-color: rgb(15, 176, 229);
  border-radius: 15px;
}

::-webkit-file-upload-button:hover {
  border-radius: 20px;
  border: 2px solid rgb(177, 174, 174);
}

input[type="submit"] {
  position: absolute;
  margin-top: 200px;
  padding: 15px 35px;
  background-color: rgb(31, 185, 190);

```



```

        border-radius: 15px;
        color: black;
        font-size: 1.5rem;
        border: 4px solid rgb(31, 185, 190);
    }
</style>
</head>
<body background="{{ url_for('static',filename = 'images/Background.jpg') }}">
    <h1 style="text-align:center;font-size:4rem">Predict Plant Leaf Disease and Get Cure</h1>
<section>

    <form action="/predict" method="post" enctype="multipart/form-data"
onsubmit="showloading()">

        <input type="file" name="image" class="upload">
            <br>
            <br>
        <input type="submit" value="Predict">

    </form>
</div>
</section>

</body>
</html>

```

## TOMATO - LATE\_BLIGHT DISEASES PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
  integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
  crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          
```

<div class="col-sm">

```
<div>
  <h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"
    class="text-center mb-5 content-h1 rounded">
    {{pred_output}} </h1>
</div>
<div class="details">
```

#### **Treatment :**

###### 

Tomatoes that have early blight require immediate attention before the disease takes over the plants. Thoroughly spray the plant (bottoms of leaves also) with Bonide Liquid Copper Fungicide concentrate or Bonide Tomato & Vegetable. Both of these treatments are organic..

```
</h6>
</div>
</div>
</div>
```

&lt;/div&gt;
&lt;/body&gt;

## TOMATO - LEAF\_MOLD DISEASES PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          
```

```
</span>
</div>
```

```
<div class="col-sm">
```

```
  <div>
    <h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"
      class="text-center mb-5 content-h1 rounded">
      {{pred_output}} </h1>
  </div>
  <div class="details">
```

```
    <h4><b> Treatment : </b> </h4>
```

```
    <h6>
```

Use drip irrigation and avoid watering foliage. Use a stake, strings, or prune the plant to keep it upstanding and increase airflow in and around it. Remove and destroy (burn) all plants debris after the harvest.</br></br>

```
    </h6>
  </div>
</div>
</div>
```

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

```
</html>
```

## TOMATO - SEPTORIA\_LEAF\_SPOT DISEASES PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
  integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
  crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          

</span>

</div>

<div class="col-sm">

<div>

<h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"  
class="text-center mb-5 content-h1 rounded">

{{pred\_output}} </h1>

</div>

<div class="details">

<h4><b> Treatment : </b> </h4>

<h6>

<b>Removing infected leaves: </b> Remove infected leaves immediately, and be sure to wash your hands and pruners thoroughly before working with uninfected plants.

<br>

<b>Consider organic fungicide options:</b> Fungicides containing either copper or potassium bicarbonate will help prevent the spreading of the disease. Begin spraying as soon as the first symptoms appear and follow the label directions for continued management.

<br>

<b>Consider chemical fungicides:</b> While chemical options are not ideal, they may be the only option for controlling advanced infections. One of the least toxic and most effective is chlorothalonil (sold under the names Fungonil and Daconil).

<br>

<br>

</h6>

</div>

</div>

</div>

</div>

</body>

</html>

## TOMATO - TARGET\_SPOT DISEASES PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
  integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
  crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          <img src="{{ url_for('static',filename = 'images/Tomato___Target_Spot.JPG') }}"
```



```
alt="User Image" style="width:500px;height:500px;">
```

```
</span>
</div>
```

```
<div class="col-sm">
```

```
<div>
  <h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"
    class="text-center mb-5 content-h1 rounded">
    {{pred_output}} </h1>
```

```
</div>
```

```
<div class="details">
```

```
<h4><b> Treatment : </b> </h4>
```

```
<h6>
```

Many fungicides are registered to control of target spot on tomatoes. Growers should consult regional disease management guides for recommended products. Products containing chlorothalonil, mancozeb, and copper oxychloride have been shown to provide good control of target spot in research trials

```
<br>
```

```
</h6>
</div>
</div>
</div>
```

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

```
</html>
```

## TOMATO - TOMATO\_MOSAIC\_VIRUS PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
  integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
  crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          

        </span>
```

</div>

<div class="col-sm">

<div>

<h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"  
class="text-center mb-5 content-h1 rounded">  
{ {pred\_output} } </h1>

</div>

<div class="details">

<h4><b> Treatment : </b> </h4>

<h6>

There are no cures for viral diseases such as mosaic once a plant is infected. As a result, every effort should be made to prevent the disease from entering your garden.

<br>

1.Fungicides will NOT treat this viral disease.

<br>

2.Plant resistant varieties when available or purchase transplants from a reputable source.

<br>

3.Do NOT save seed from infected crops.

<br>

4.Spot treat with least-toxic, natural pest control products, such as Safer Soap, Bon-Neem and diatomaceous earth, to reduce the number of disease carrying insects.

<br>

5.Harvest-Guard® row cover will help keep insect pests off vulnerable crops/ transplants and should be installed until bloom.

<br>

6.Remove all perennial weeds, using least-toxic herbicides, within 100 yards of your garden plot.

<br>

7.The virus can be spread through human activity, tools and equipment. Frequently wash your hands and disinfect garden tools, stakes, ties, pots, greenhouse benches, etc. (one part bleach to 4 parts water) to reduce the risk of contamination.

<br>

8.Avoid working in the garden during damp conditions (viruses are easily spread when plants are wet).

<br>

9.Avoid using tobacco around susceptible plants. Cigarettes and other tobacco products may be infected and can spread the virus.

<br>

10.Remove and destroy all infected plants (see Fall Garden Cleanup). Do NOT compost.

<br>

</h6>

</div>

</div>

</div>

</div>

</body>

</html>

## TOMATO - TOMATO\_YELLOW\_LEAF\_CURL\_VIRUS PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
  integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
  crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          
```

```
</span>
```

```
</div>
```

```
<div class="col-sm">
```

```
<div>
```

```
<h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;" class="text-center mb-5 content-h1 rounded">{{pred_output}} </h1>
```

```
</div>
```

```
<div class="details">
```

```
<h4><b> Treatment : </b> </h4>
```

```
<h6>
```

Inspect plants for whitefly infestations two times per week. If whiteflies are beginning to appear, spray with azadirachtin (Neem), pyrethrin or insecticidal soap. For more effective control, it is recommended that at least two of the above insecticides be rotated at each spraying.

```
</h6>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

## TOMATO - TWO-SPOTTED\_SPIDER\_MITE DISEASES PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
  integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
  crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          

</span>

</div>

<div class="col-sm">

<div>

<h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"  
class="text-center mb-5 content-h1 rounded">

{{pred\_output}} </h1>

</div>

<div class="details">

<h4><b> Treatment : </b> </h4>

<h6>

For control, use selective products whenever possible.

Selective products which have worked well in the field include:

bifenazate (Acramite): Group UN, a long residual nerve poison

abamectin (Agri-Mek): Group 6, derived from a soil bacterium

spirotetramat (Movento): Group 23, mainly affects immature stages

spiromesifen (Oberon 2SC): Group 23, mainly affects immature stages

OMRI-listed products include:

insecticidal soap (M-Pede)

neem oil (Trilogy)

soybean oil (Golden Pest Spray Oil)

With most miticides (excluding bifenazate), make 2 applications, approximately 5-7 days apart, to help control immature mites that were in the egg stage and protected during the first application.

Alternate between products after 2 applications to help prevent or delay resistance.

</h6>

</div>

</div>

</div>

</div>

</body>

</html>

## TOMATO-BACTERIA SPOT DISEASES PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
  integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
  crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          <img src="{{ url_for('static',filename = 'images/Tomato___Bacterial_spot.JPG') }}"
```



```
alt="User Image" style="width:500px;height:500px;">
```

```
</span>
```

```
</div>
```

```
<div class="col-sm">
```

```
<div>
```

```
<h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"  
class="text-center mb-5 content-h1 rounded">
```

```
{{pred_output}} </h1>
```

```
</div>
```

```
<div class="details">
```

```
<h4><b> Treatment : </b> </h4>
```

```
<h6>
```

Copper fungicides are the most commonly recommended treatment for bacterial leaf spot. Use copper fungicide as a preventive measure after you've planted your seeds but before you've moved the plants into their permanent homes. You can use copper fungicide spray before or after a rain, but don't treat with copper fungicide while it is raining. If you're seeing signs of bacterial leaf spot, spray with copper fungicide for a seven- to 10-day period, then spray again for one week after plants are moved into the field.

Perform maintenance treatments every 10 days in dry weather and every five to seven days in rainy weather.</br></br>

```
</h6>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

## TOMATO-EARLY\_BLIGHT DISEASES PREDICTION WITH TREATMENT HTML CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
  integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
  crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          

</span>

</div>

<div class="col-sm">

<div>

<h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"  
class="text-center mb-5 content-h1 rounded">

{{pred\_output}} </h1>

</div>

<div class="details">

<h4><b> Treatment : </b> </h4>

<h6>

Tomatoes that have early blight require immediate attention before the disease takes over the plants. Thoroughly spray the plant (bottoms of leaves also) with Bonide Liquid Copper Fungicide concentrate or Bonide Tomato & Vegetable. Both of these treatments are organic..</br></br>

</h6>

</div>

</div>

</div>

</div>

</body>

</html>

## TOMATO-HEALTHY:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/css/bootstrap.min.css"
integrity="sha384-
VCmXjywReHh4PwowAiWNagnWcLhlEJLA5buUprzK8rxFgeH0kww/aWY76TfkUoSX"
crossorigin="anonymous">

  <title>TOMATO LEAF DISEASE PREDICTION</title>

  <style>
    * {
      margin: 0px;
      padding: 0px;
      box-sizing: border-box;
    }

    .border img {
      border-radius: 15px;
      border: 2px solid black;
    }

  </style>
</head>

<body background="{{ url_for('static',filename = 'images/Back.jpg') }}">

  <div>
    <h1 style="text-align:center;font-size:4rem">Predict Tomato Leaf Disease & Get Cure</h1>
  </div>

  <br>
  <br>

  <div class="container my-2">
    <div class="row mb-5">

      <div class="col-sm" style="margin-bottom: 23px;">
        <span class="border border-primary">
          
```

```
</span>
</div>
```

```
<div class="col-sm">
```

```
  <div>
    <h1 style="padding: 15px; background-color: rgb(15, 15, 15); color: white;"
      class="text-center mb-5 content-h1 rounded">
      {{pred_output}} </h1>
  </div>
  <div class="details">
```

```
    <h4><b> There is no disease on the Tomato leaf. </b> </h4>
```

```
  </div>
</div>
</div>
```

```
</div>
</body>

</html>
```

## EXPLANATION:

### Training Module

This code is for training a convolutional neural network (CNN) to predict the type of plant disease based on images of plant leaves. The CNN is built using the Keras library and has four main steps: convolution, pooling, flattening, and fully connected layers.

The first step is convolution, where the input image is filtered through a set of small matrices called kernels, to detect specific features in the image. This is done using the Conv2D layer with 32 filters and a 3x3 kernel size.

The second step is pooling, where the feature maps are downsampled to reduce the number of parameters and computation needed. This is done using the MaxPooling2D layer with a 2x2 pool size.

The third step is flattening, where the output from the pooling layer is reshaped into a 1D array.

The fourth step is fully connected layers, where the flattened output is fed into a series of dense layers that are used to classify the image. The first dense layer has 128 units with a relu activation function, and the second dense layer has 10 units with a sigmoid activation function.

The CNN is then compiled using the adam optimizer, categorical\_crossentropy loss function, and accuracy as a metric for evaluation.

Next, the code uses an ImageDataGenerator object from Keras to generate batches of images for training and validation. The images are normalized by rescaling their pixel values to between 0 and 1 and then augmented by applying random transformations such as shear, zoom, and horizontal flip.

The training set and validation set are then created using the flow\_from\_directory method of the ImageDataGenerator object, which reads the images from the specified directory and converts them into batches of normalized and augmented images.

The CNN is trained using the fit\_generator method, which takes in the training and validation sets and trains the model over 50 epochs.

Finally, the trained model is saved to disk as a JSON file for model architecture and HDF5 file for model weights.

There is also commented-out code for testing the model on a single image, where the image is loaded, resized to 128x128, and then passed through the trained CNN to predict the type of plant disease. The predicted class is then displayed on the image using OpenCV.

### **Leaf Diseases Detection Code:**

The code is a Flask web application that predicts the type of disease present in a tomato plant's leaf by taking an image of the leaf as input. The application uses a pre-trained deep learning model to classify the input image into one of ten classes of tomato plant diseases.

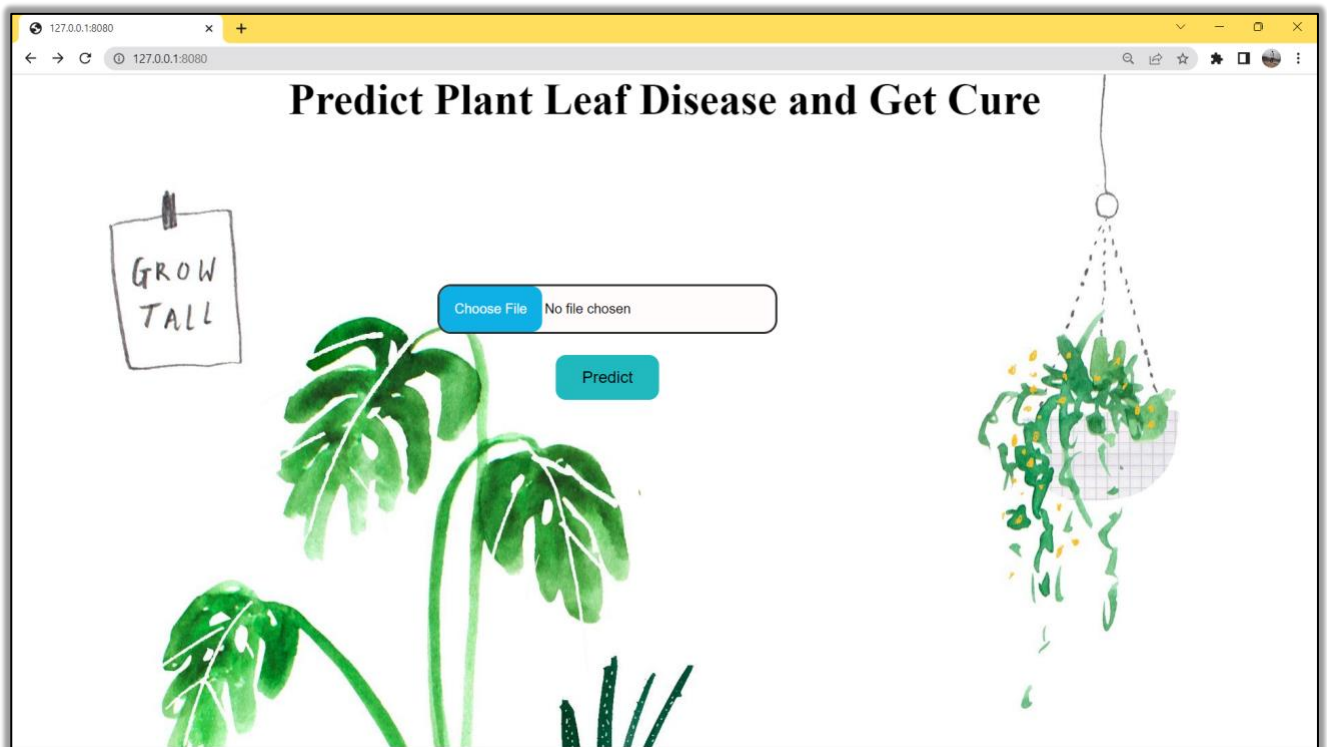
The code first imports the necessary libraries, including Flask for web development, NumPy for numerical computations, and Keras for deep learning. The pre-trained model is loaded from a saved model file using the Keras `load_model()` function.

A function named `pred_tomato_dieas()` takes the path of an image of a tomato leaf as input and returns the predicted class label and the name of the HTML file that displays the diagnosis result. The function first loads the input image using Keras' `load_img()` function and preprocesses it by converting it to a NumPy array and normalizing the pixel values. Then, it uses the loaded model to predict the disease class using the `predict()` function, which returns the probability scores for each class. The function then selects the class with the highest probability score using the `np.argmax()` function and returns the corresponding class label and the name of the HTML file that displays the diagnosis result.

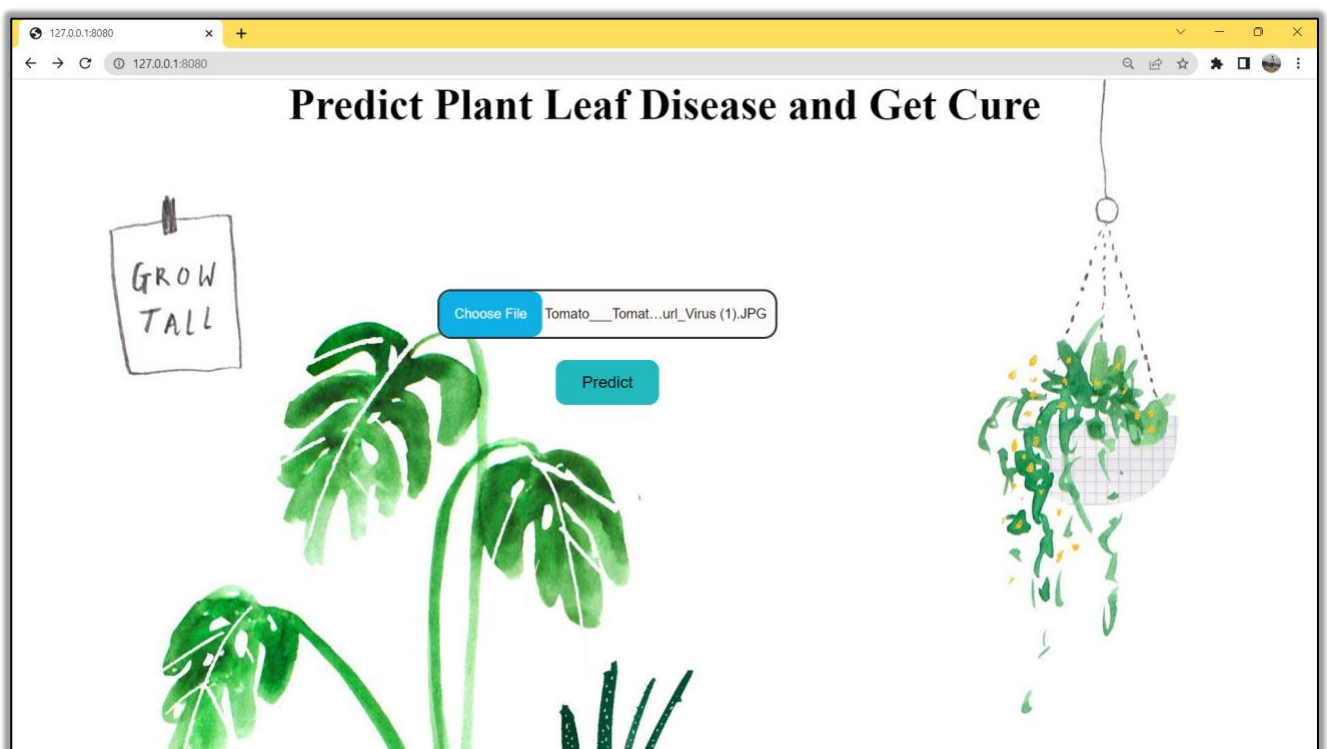
The Flask web application is created using the `Flask()` constructor, and it consists of two routes. The first route, `/`, renders the `index.html` template file, which displays a form for uploading an image. The second route, `/predict`, is triggered when the user submits the image through the form. It receives the uploaded image file and saves it to the local filesystem. Then, it calls the `pred_tomato_dieas()` function to predict the disease class and returns the predicted label and the name of the HTML file that displays the diagnosis result. The HTML file is rendered using Flask's `render_template()` function, which populates the placeholders in the HTML file with the appropriate values.

Finally, the Flask application is run on the local machine or cloud using the `app.run()` function, which specifies the port number and threaded parameter. The application can be accessed in the browser at the specified URL, where users can upload an image and get the diagnosis result for the tomato plant leaf.

## RESULT



Home Page For Leaf Disease Prediction



Choose File To Predict Leaf Disease





### Prediction Of Diseases With Its Treatment

## **EXPLANATION:**

### **User Point Of View**

Crop monitoring is an essential aspect of modern agriculture, and disease detection is a critical component of it. With the help of disease detection techniques, farmers can detect diseases at an early stage and take necessary actions to prevent it from spreading to other crops, which can cause a significant loss in yield.

Disease detection in crops can be done through various methods, including visual inspection, laboratory testing, and remote sensing. However, with the advancement of technology, remote sensing has become a popular choice among farmers for crop monitoring and disease detection.

Remote sensing involves using sensors to collect information about crops from a distance. This information can be used to detect diseases, monitor crop growth, and predict yield. Some of the sensors used for remote sensing include satellites, drones, and ground-based sensors.

In crop monitoring, remote sensing helps farmers detect diseases at an early stage by providing high-resolution images of crops. These images can be analyzed using machine learning techniques, such as convolutional neural networks (CNNs), to detect diseases accurately.

CNNs are trained using a dataset of healthy and diseased crop images. Once trained, they can identify patterns and features in new crop images and predict if a crop is healthy or diseased. With this information, farmers can take necessary actions to prevent the spread of the disease and minimize crop loss.

Overall, crop monitoring in disease detection is an important aspect of modern agriculture. By using remote sensing and machine learning techniques, farmers can detect diseases at an early stage, take necessary actions, and maximize crop yield.

## CONCLUSION

Crop monitoring is a critical practice in agriculture that enables farmers to detect and prevent diseases in their crops. By monitoring their crops regularly, farmers can identify early signs of diseases, take appropriate action, and prevent the spread of the disease, leading to higher yields and better quality crops. With the help of advanced technology, such as remote sensing techniques and digital tools, crop monitoring has become more efficient and accurate than ever before.

One of the primary benefits of crop monitoring is disease detection. Many plant diseases can spread quickly, leading to severe crop losses and economic damage. By monitoring their crops, farmers can detect signs of diseases, such as discoloration, wilting, or abnormal growth, at an early stage. They can then take immediate action, such as removing the infected plant or applying fungicides, to prevent the spread of the disease and minimize crop losses.

Crop monitoring also helps farmers identify other issues that may affect crop health, such as nutrient deficiencies, water stress, or insect infestations. By monitoring soil moisture levels, plant growth rates, and other parameters, farmers can optimize their farming practices and adjust their irrigation and fertilization schedules to improve crop health and yield.

Remote sensing techniques, such as satellite imagery and drones, have revolutionized crop monitoring, enabling farmers to monitor large areas quickly and accurately. These techniques can detect changes in crop health that may not be visible to the naked eye, such as differences in chlorophyll content or canopy temperature. Digital tools, such as mobile apps and web-based platforms, provide farmers with real-time data and analytics, enabling them to make informed decisions about their farming practices.

In conclusion, crop monitoring is a crucial practice in agriculture that enables farmers to detect and prevent diseases in their crops. With the help of advanced technology, such as remote sensing techniques and digital tools, crop monitoring has become more efficient and accurate than ever before. By implementing regular crop monitoring practices, farmers can improve their crop yields, optimize their farming practices, and contribute to the overall food security of their communities and the world.