FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

ANNIE SUJITH

Contents

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to IBM for providing me this opportunity under the academic initiative to enhance my knowledge in Artificial Intelligence. Also, I am very grateful to all the mentors of SmartInternz especially to Mr. Shivam Shivhare and Mr. Hemant Kumar for explaining all the concepts very clearly and thoroughly and for their patience in clearing all our doubts. They are always readily available and eager to help and guide every participant.

1 INTRODUCTION

1.1 Overview

Agriculture is recognized as the most significant profession in India. India is the second-largest producer of agricultural goods in the world. Here, farmers prepare a vast variety of agricultural goods and crops. The construction of agricultural goods is influenced by a variety of elements, including weather, soil quality, various diseases, and others. Although the current methods for disease prediction in plants only depend on visual inspection, these techniques will need more personnel, well-equipped laboratories, and a range of tools [1]. The agricultural system and crops are subject to influence from a wide range of factors. These variables can include the soil, the climate, and the relief of the terrain [3]. Prosperous agriculture depends on healthy soil since it is the original and major source of the nutrients needed to grow crops. The foods we consume include the nutrients found in the soil. The finest soils are used to cultivate the tastiest and most plentiful food staples [4].

1.2 Purpose

Incorrect disease prediction in plants or crops may also result in random pesticide use in the field, which can eventually result in pathogen resistance, lowering the resistance of agricultural goods. When predicting illness, the leaf of the infected crop is examined. It seems to be very difficult to predict sickness in plants or crops using photos from mobile cameras [2]. Convolutional neural network (CNN)-based models have increased the effectiveness of categorization, and we are using image processing in conjunction with CNN to predict crop sickness. Thus, this work suggested a system that can identify plant illnesses using CNN and recommend a cure for them.

2 LITERATURE SURVEY

2.1 Existing problem

The standard quality and quantity of agricultural output are severely harmed by the widespread occurrence of crop diseases and by the inability of the soil to support plant growth. Therefore, it is important to identify crop diseases as soon as possible by coming up with or using a quick, creative method, and a crop recommendation system will be helpful to farmers. In the existing system, the approach used to anticipate plant diseases is visual examination, however, these procedures need knowledgeable staff, well-equipped labs, and a variety of equipment. Examining an infected crop's leaf may help forecast sickness. In some of the existing automated systems, images from mobile cameras are used, but it seems to be exceedingly difficult to anticipate illness in plants or crops by using these methods [2].

2.2 Proposed solution

As categorization has become more successful thanks to convolutional neural network (CNN) based models, in our work we are combining image processing and CNN to predict crop sickness. As a result, this application uses CNN to recognize plant diseases and also suggest a treatment. First, the training and testing dataset for illness prediction is gathered. The classifier is then trained using training data. The next phase is to acquire a test database made up of several plant diseases that will be used to evaluate the proposed module's precision and degree of confidence. The output will then be identified with the highest accuracy possible.

3 THEORETICAL ANALYSIS

3.1 Architecture diagram

The module extracts the leaf characteristics initially when we provide a fresh input picture. The CNN model is then applied. The features are then compared to the training dataset. The leaf characteristics are retrieved independently after it runs through a dense CNN. The module will then forecast if a disease is present on the plant leaf or not. It displays the output from one of the preset and trained classes that are listed. The result will then be shown as text.

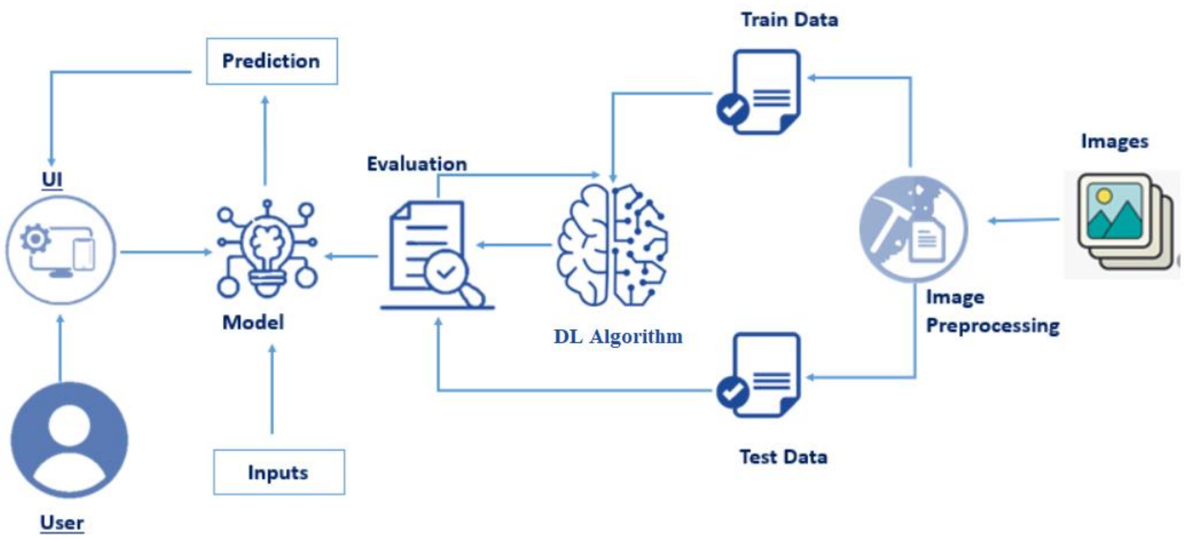

Figure 1 Architecture Diagram [5]

3.2 System Requirements

The following software and packages are needed for building this application.

Anaconda Navigator: Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning-related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder.

For building Deep learning models we require the following packages:

Tensor flow: TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers can easily build and deploy ML powered applications.

Keras: Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features: Consistent, simple and extensible API.

Minimal structure - easy to achieve the result without any frills.

It supports multiple platforms and backends.

It is a user-friendly framework that runs on both CPU and GPU.

It has high scalability of computation.

Flask: Web framework used for building Web applications.

4 EXPERIMENTAL INVESTIGATIONS

For diagnosing correctly plant diseases by using an application, an image processing system was used to develop and automate the identification and classification of the diseases by studying the respective leaves of plants. The dataset is pre-processed like Image reshaping, resizing and conversion to an array form. Similar processing is additionally done on the test image. A dataset consisting of different plant leaf diseases is obtained, out of which an image is often used as a test image for the software. The training dataset is employed to coach the model (CNN) so that it can identify the test image and therefore the disease. CNN has different layers that are Dense, Activation, Flatten, Convolution2D, and maxpooling2d. After the model is trained successfully, the software can identify the disease if the plant species are contained within the dataset.

5 FLOWCHART

We have built a web application where the Farmers interact with the user interface to upload images of a diseased leaf. Our model analyses the image of the leaf to identify the disease and suggests the farmer with fertilizers to be used for curing the disease.

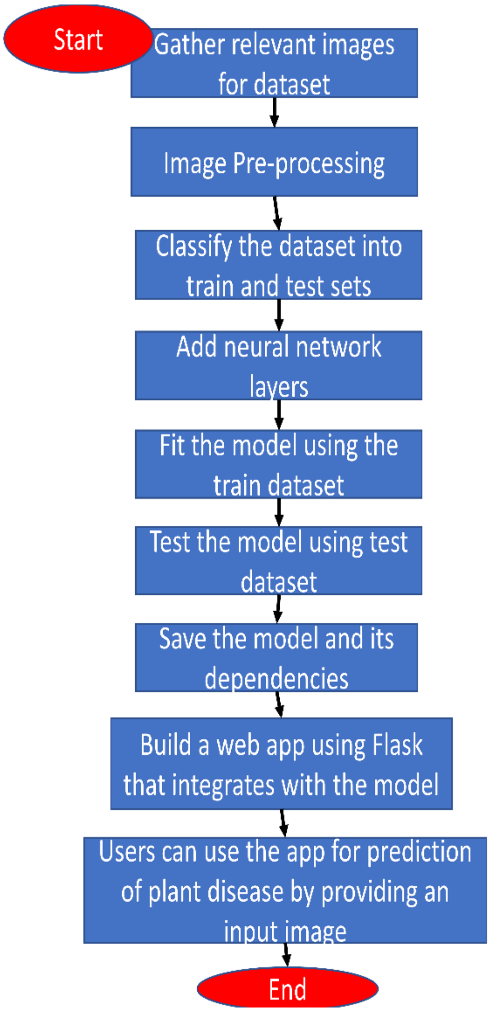

Figure 2 Flow Chart

6 RESULT

In the built model an overall accuracy of 92% was achieved. The trained model was tested on each class individually. This work explores the approach of using a deep learning method to detect plant diseases from leaf images.

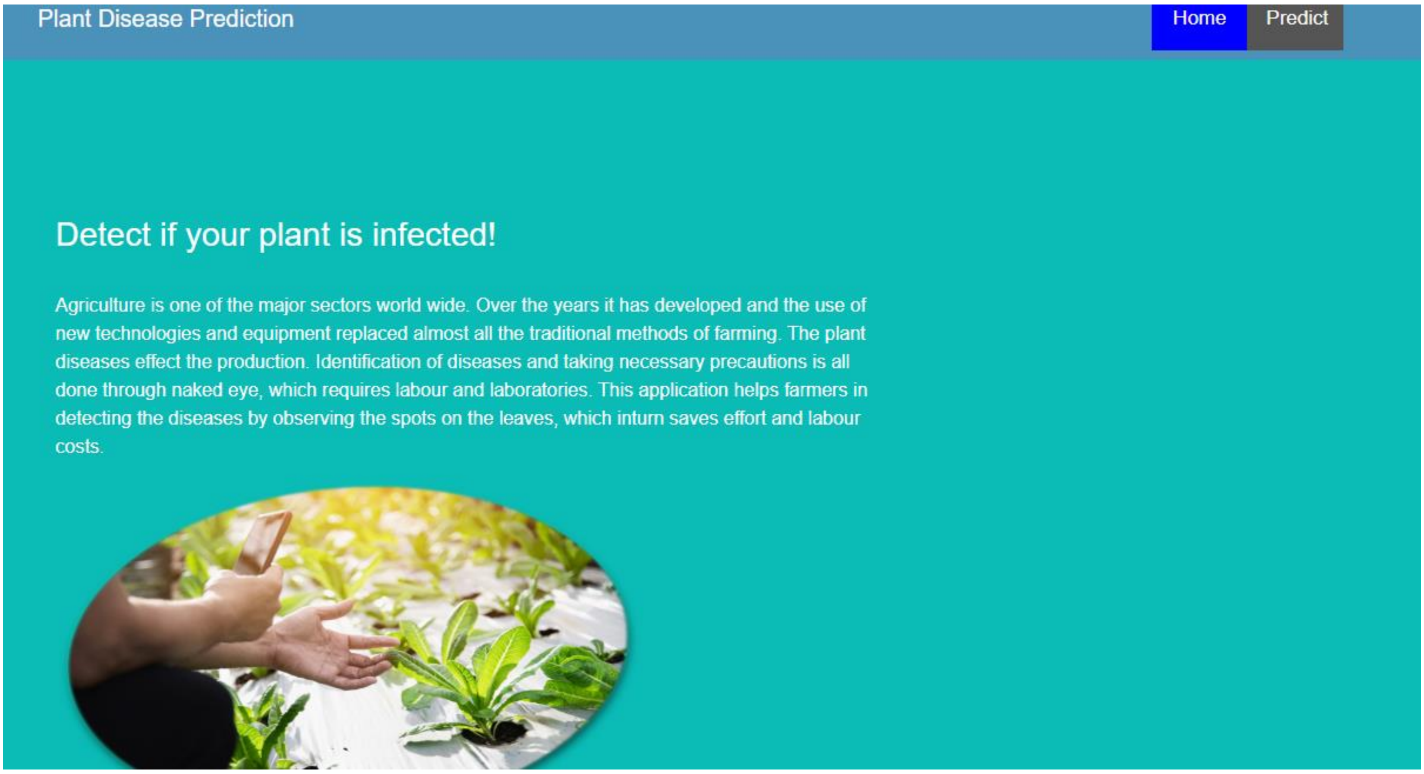

Figure 3 Home page

Figure 3 shows the Home Page of the Plant disease detection and Fertilizer recommendation system.

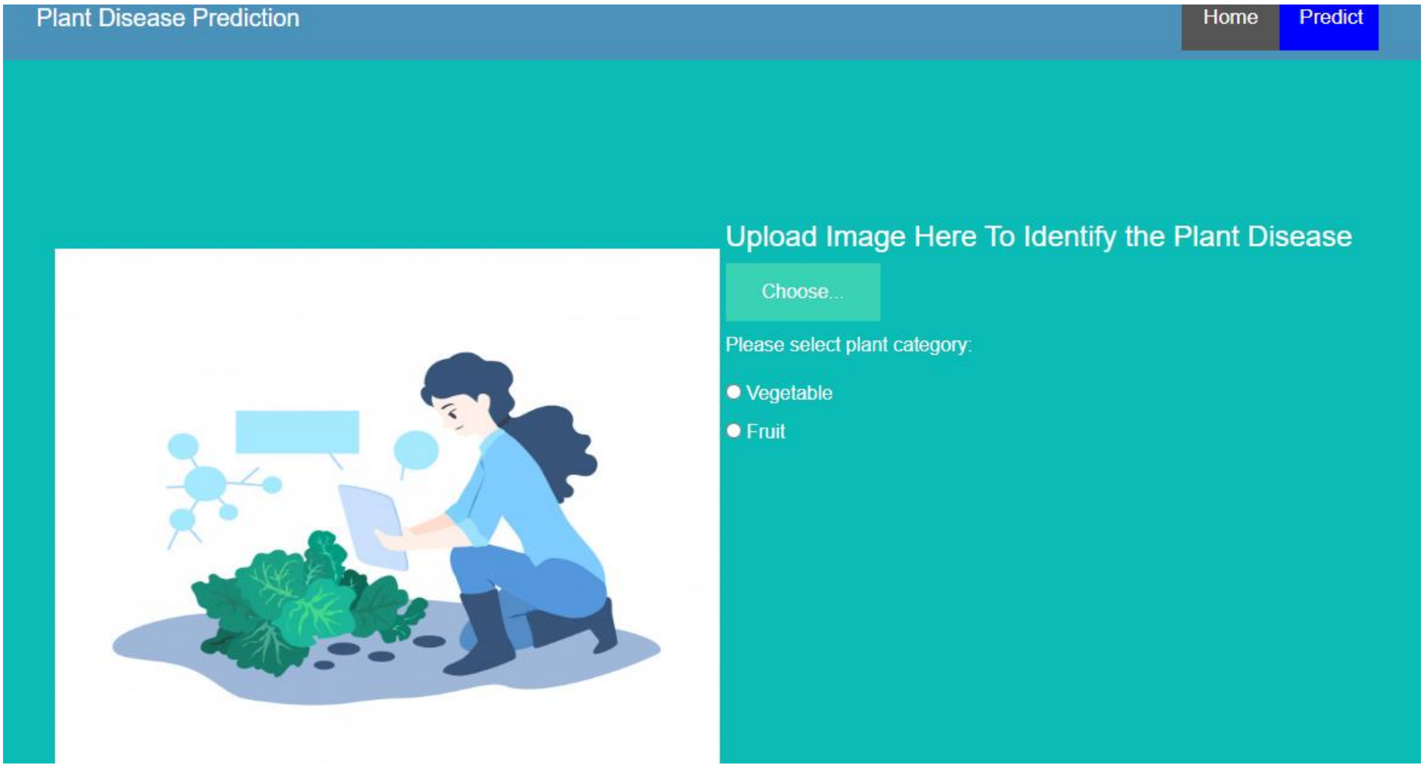

Figure 4 Web page for Plant disease prediction

Figure 4 shows the user interface where farmers can upload images of leaves for disease prediction and fertilizer recommendation.

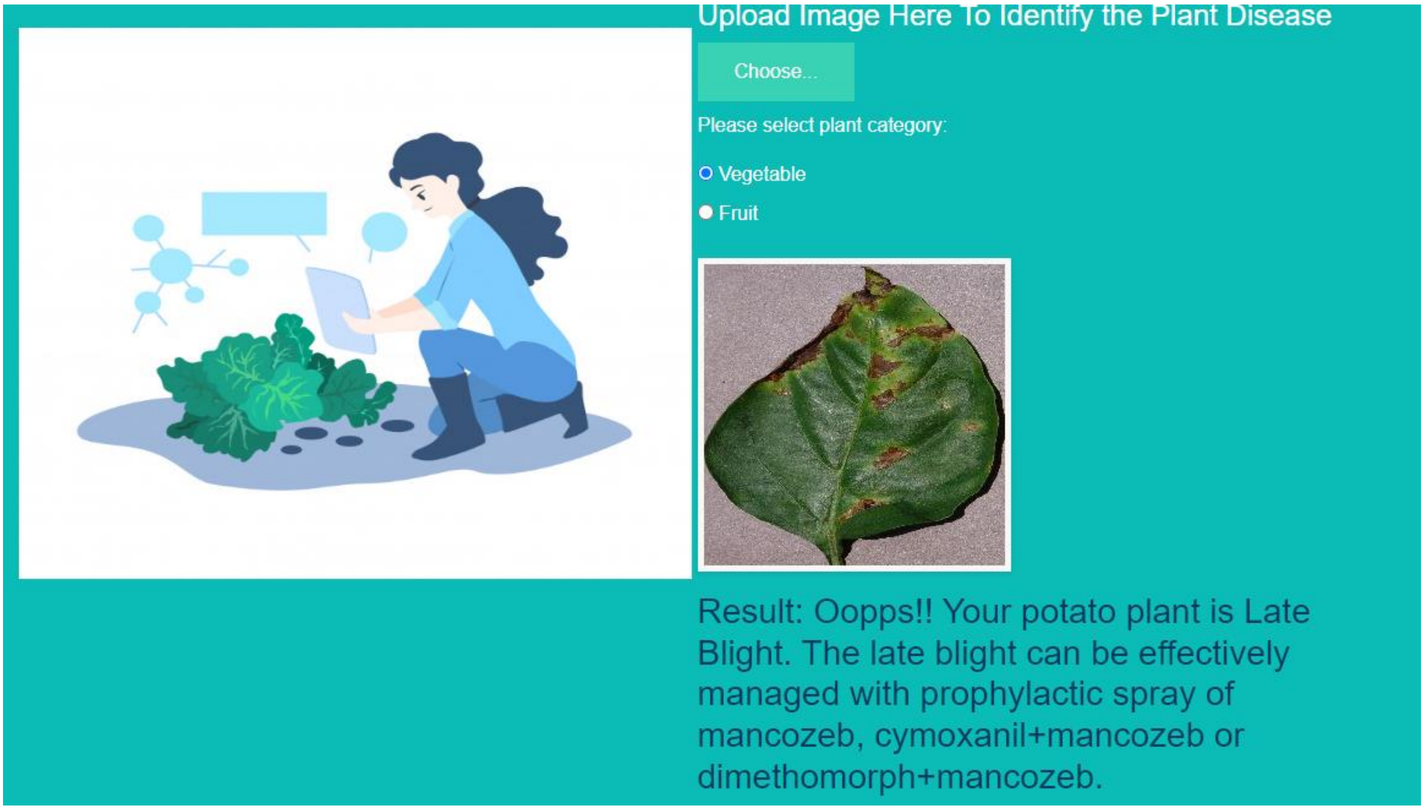

Figure 5 Web page displaying the result for a vegetable plant

For testing purposes, a leaf image of a potato plant infected with late blight was given as input to the software. As seen in Figure 5 the software has correctly identified the disease and also provided fertilizer recommendations for the cure of the vegetable plant.

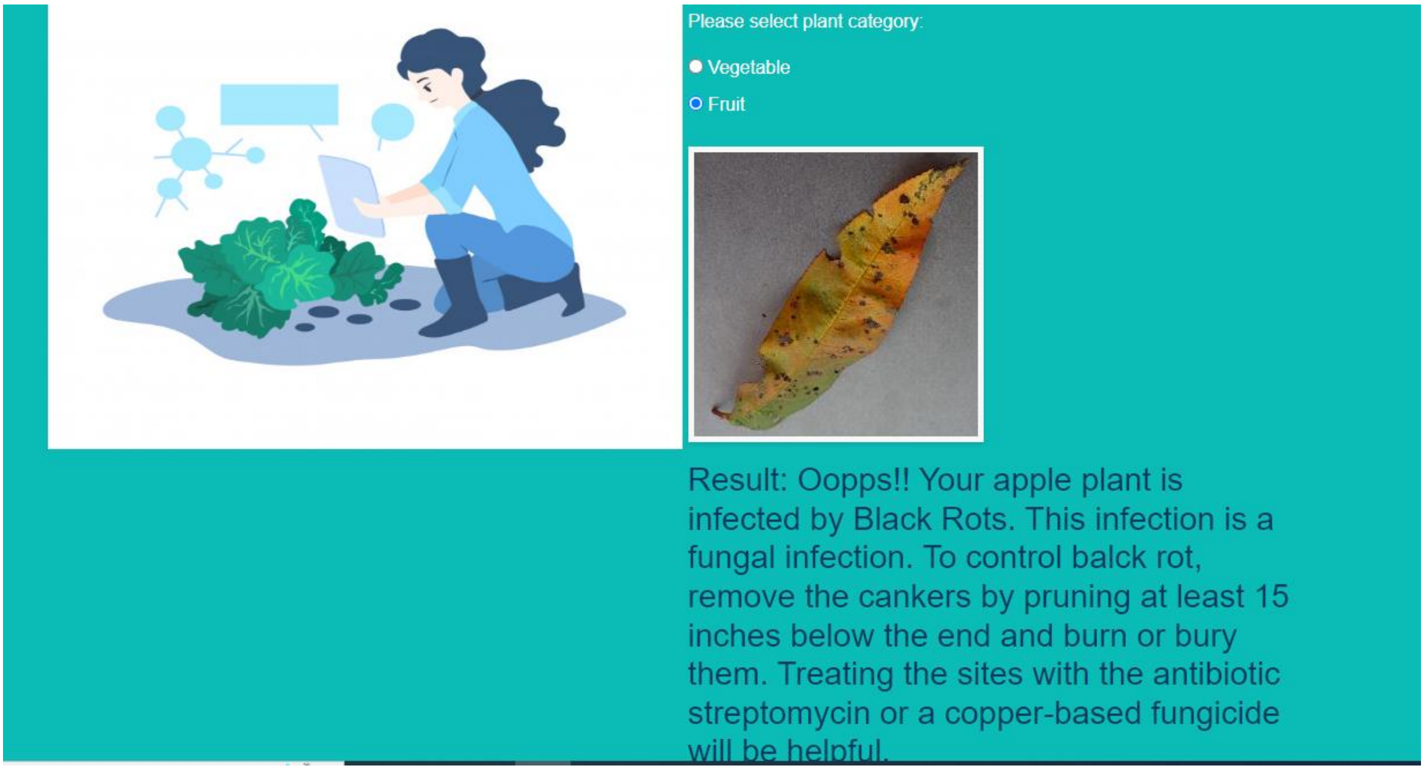

Figure 6 Web page displaying the result for a fruit plant

For testing purposes, a leaf image of an apple plant infected by black rot was given as input to the software. As seen in Figure 6 the software has correctly identified the disease and also provided fertilizer recommendations for the cure of the fruit plant.

7 ADVANTAGES & DISADVANTAGES

Plant diseases have long been a major problem in agriculture since they lower crop quality and hence productivity. Plant diseases can cause anything from minor symptoms to severe damage to entire fields of planted crops. This results in significant financial costs and has a significant negative impact on the agricultural economy, especially in developing nations where the economy is heavily dependent on a single crop or a small number of crops. The advantage of the system is that it can identify plant illnesses using CNN and recommend a cure for them. The software uses the latest technology of deep learning to optimize the decision-making process. In the traditional methods, the approach used to anticipate plant diseases is visual examination, however, these procedures need knowledgeable staff, well-equipped labs, and a variety of equipment. The suggested method provides more accurate predictions than traditional methods.

Several tests were conducted to analyze the impact of training in a controlled environment, however, the software should be tested by using it in real-life situations to accurately identify plant diseases in a complex background and in various conditions. A disadvantage of the model is that it is currently unable to detect multiple diseases in a single leaf. The increased quantity of data needed by DL approaches is necessary to provide improved outcomes in the identification of plant disease. Images taken under various circumstances should be included in a comprehensive dataset as much as possible. This is a disadvantage since the datasets that are presently accessible often have insufficient numbers of images, which are required for making high-quality decisions.

8 APPLICATIONS

Remarkable results of using Deep learning approach in various domains such as automatic game playing, natural language processing, autonomous vehicles etc has encouraged researchers to also apply it in the field of agriculture. Plant diseases can cause severe damage to entire fields of planted crops. This results in significant financial losses and has a significant negative impact on the agricultural economy of nations. Hence using it in the field of agriculture is critical and more research is needed in this field. This developed software can be used by farmers doing farming on both large scale and small scale. It can also be of help to users who adopt vegetable cultivation on private terraces and balconies as well.

9 CONCLUSION

Plant diseases have long been a major issue in agriculture. Making the best judgments possible based on the outcomes of Deep Learning techniques has made early disease identification and the minimizing of losses possible. Recent developments in DL provide solutions with very precise findings, and the technology that is now accessible permits quick processing. The decision-making process, nevertheless, may be enhanced. The models that are now available perform poorly when evaluated under actual circumstances. Taking a snapshot of the plant leaf and giving as input into a smartphone application provides a low-cost solution for identifying plant illnesses.

10 FUTURE SCOPE

The trained model had an accuracy of 92 percent. In future enhancements, exploiting other data sources, such as location, climate, and plant age, might improve accuracy. Future research should concentrate on identifying disease stages and locations throughout the plant. It can include the detection of multiple diseases in a single leaf. The system can be installed on drones for aerial surveillances of crop fields by the agricultural department.

REFERENCES

[1] International Journal of Computer Science and Information Technologies, vol. 5, no. 3, pp. 4638-4645, 2014.

[2] I. Sutskever Krizhevsky and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *In Advances in neural information processing systems*, pp. 1097-1105, 2012.

[3] Artificial Neural Networks, Humana Press, 2015.

[4] Preetam Tamsekar, Nilesh Deshmukh, Parag Bhalchandra, Govind Kulkarni, Kailas Hambarde and Shaikh Husen, Comparative Analysis of Supervised Machine Learning Algorithms for GIS-Based Crop Selection Prediction Model, Journal of Springer Nature Singapore Pte Ltd, pp. 310-312, 2019.

[5] SmartInternz Project information

APPENDIX

A. Source Code

Flask file (app.py)

```
import numpy as np

import os

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

from flask import Flask,render_template,request, redirect, url_for

import requests

import pandas as pd

import tensorflow as tf

from werkzeug.utils import secure_filename

from tensorflow.python.keras.backend import set_session


app=Flask(__name__)

model=load_model("fruit.h5")

model1=load_model("vegetable.h5")

@app.route('/')

def home():

    return render_template("home.html")

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

def predict():

    if request.method=='POST':
```

```
f=request.files['image']

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=np.argmax(model.predict(x),axis=1)

    index=['Pepper Bell Bacterial spot','Pepper bell healthy','Potato Early blight','Potato Late blight','Potato
healthy','Tomato Bacterial spot','Tomato Late blight','Tomato Leaf Mold','Tomato Septoria leaf spot']

    print(index[preds[0]])

    df=pd.read_excel('precautions - veg.xlsx')

    print(df.iloc[preds[0]]['caution'])

else:

    preds=np.argmax(model1.predict(x),axis=1)

    index=['Apple Black rot','Apple healthy','Corn (maize) Northern Leaf Blight','Corn (maize)
healthy','Peach Bacterial spot',

 'Peach healthy']

    print(index[preds[0]])

    df=pd.read_excel('precautions - fruits.xlsx')

    print(df.iloc[preds[0]]['caution'])
```

```
        #text="The prediction is : " +str(index[preds[0]])
        #return text
        return df.iloc[preds[0]]['caution']
if __name__=='__main__':
    app.run(debug=False)
```

HTML File (home.html)

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Plant Disease Prediction</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
        <style>
        .bg-dark {
                background-color:     #4B92BB!important;
        }
        #result {
                color: #0a1c4ed1;
        }
```

```
        body
{
    background-image: url("https://wallpaperaccess.com/full/327148.jpg");
    background-size: cover;
}
    body, html {
      height: 100%;
      margin: 0;
      font-family: Arial;
    }
    .tablink {
      background-color: #555;
      color: white;
      float: left;
      border: none;
      outline: none;
      cursor: pointer;
      padding: 14px 16px;
      font-size: 17px;
      width: 50%;
    }
    .tablink:hover {
      background-color: #777;
```

```
}
.tabcontent {
  color: white;
  display: none;
  padding: 100px 20px;
  height: 100%;
}
</style>
        <script>
        function openPage(pageName, elmnt, color) {
      var i, tabcontent, tablinks;
      tabcontent = document.getElementsByClassName("tabcontent");
      for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
      }
      // Remove the background color of all tablinks/buttons
      tablinks = document.getElementsByClassName("tablink");
      for (i = 0; i < tablinks.length; i++) {
        tablinks[i].style.backgroundColor = "";
      }
      document.getElementById(pageName).style.display = "block";
      elmnt.style.backgroundColor = color;
    }
```

```
        document.getElementById("defaultOpen").click();

            </script>

</head>

<body>

  <nav class="navbar navbar-dark bg-dark">

        <div class="container">

            <a class="navbar-brand" href="#">Plant Disease Prediction</a>

            <div class="tab">

                <button class="tablink" onclick="openPage('Home', this, 'blue')" id="defaultOpen">Home</button>

                <button class="tablink" onclick="openPage('Predict', this, 'blue')">Predict</button>

            </div>

        </div>

    </nav>

    <div id="Home" class="tabcontent">

     <div class="container">

         <div id="content" style="margin-top:2em">

                        <div class="container">

                          <div class="row">

                                    <div class="col-sm-8 bd">

                                      <h3>Detect if your plant is infected! </h3>

                                      <br>

                                      <p>Agriculture is one of the major sectors world wide. Over the years it has developed and the use of new technologies and equipment replaced almost all the traditional methods of farming. The plant diseases effect the production. Identification of diseases and taking necessary precautions is all done through
```

naked eye, which requires labour and laboratories. This application helps farmers in detecting the diseases by observing the spots on the leaves, which inturn saves effort and labour costs.</p>

```
<img src="{{url_for('static', filename='img/12456.png')}}" style="width:500px;height:300px;">

</div>

</div>

</div>

</div>

</div>

</div>



<div id="Predict" class="tabcontent">

<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='img/789.jpg')}}" style="height:450px;width:550px"class="img-rounded" alt="Gesture">

</div>

<div class="col-sm-6">

<div>

<h4>Upload Image Here To Identify the Plant Disease</h4>
```

```
                    <form action = "http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-data">
                        <label for="imageUpload" class="upload-label">
                            Choose...
                        </label>
                        <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">

                        <p>Please select plant category:</p>
        <input type="radio" id="vegetable" name="plant" value="vegetable">
        <label for="vegetable">Vegetable</label><br>
        <input type="radio" id="fruit" name="plant" value="fruit">
        <label for="fruit">Fruit</label><br>
                        </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">Predict!</button>
                        </div>
                    </div>
```

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


                        <h3>

                                <span id="result"> </span>

                        </h3>


                </div>

                        </div>

                         </div>

                </div>

                </div>

    </div>

</div>

</body>

<footer>

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

</footer>

</html>
```