PROJECT REPORT

Fertilizers Recommendation for Disease Prediction

Team ID: PNT2022TMID46043

Team Leader: Madhumitha A-814719106036

Team Member: Rangeetha S-814719106046

Team Member: Rohan S 814719106049

Team Member: Shivani S-814719106054

INTRODUCTION:

Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

Project Overview

An Automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases changes in cultivation method and inadequate plant protection techniques and suggest all the precautions that can be taken for those diseases.

Purpose

- **To Detect and recognize the plant diseases and to recommend fertilizer, it is necessary to identify the diseases and to recommend to get different and useful features needed for the purpose of analyzing later.**

- **To provide symptoms in identifying the disease at its earliest. Hence the authors proposed and implemented new fertilizers Recommendation System for Crop Disease Prediction.**

LITREATURE SURVEY

Existing Problem

- **Adequate mineral nutrition is central to crop production. However, it can also exert considerable Influence on disease development. Fertilizer application can increase or decrease development of diseases caused by different pathogens,**
  and the mechanisms responsible are complex, including effects of nutrients on plant growth, plant resistance mechanisms and direct effects on the pathogen.The effects of mineral nutrition on plant disease and the mechanisms responsible for those effects have been dealt with comprehensively elsewhere. In India, around 40% of land is kept and grown using reliable irrigation technologies, while the rest relies on the monsoon environment for water. Irrigation decreases reliance on the monsoon, increases food security, and boosts agricultural production.

- **Most research articles use humidity, moisture, and temperature sensors near the plant's root, with an external device handling all of the data provided by the sensors and transmitting it directly to an Android application. It was created to**
  measure the approximate values of temperature, humidity and moisture sensors that were programmed into a microcontroller to manage the amount of water.

Empathy Map Canvas :

Ideation & Brainstorming :

BLOCK DIAGRAM:

Figure.3.1. Block Diagram of the project

The block diagram of the entire project is shown in Fig.3.1. First step is the image dataset collection followed by image preprocessing. The third step is the training of image datasets with initializing different hyper parameters. Then build the model and save the model file with h5 format. The final stage is the testing of existing or new datasets using the trained model.

HARDWARE/SOFTWARE DESIGNING:

arranged in different batch sizes. Then train set and test set are formed from the collected datasets. In order to do the above steps in Python, the following Python libraries must be imported before starting the process:

- NumPy
- TensorFlow
- Keras
- Matplotlib (optional for data visualization) **The software used for training and testing the dataset is Python. The Jupyter notebook (Notebook of IBM cloud also) is used for python programming. The neural network used for training and testing the model is Convolutional Neural Network (CNN).**

The CNN has following layers:

- **Convolutional layer (32x32 kernal (3x3))**
- **Max-pool layer (kernel(2x2))**
- **Flatten layer**
- **Dense layer (different layers with different size)**

- **Drop out layer (optional)**

- **Final output dense layer(size 6x1 for fruit dataset and 9x1 for Vegetable dataset)**

In the preprocessing step, images are normalized to 1 and then resized to 128x128. The images are

The following activation functions used in the CNN training:

- **RELU at the end of convolution layer and Max Pool layer**

- **SoftMax at the end of output dense layer**

- **For testing the dataset argmax is used, its an optional**

EXPERIMENTAL INVESTIGATIONS

Analysis made while working on the solution. The batch sizes are varied and tested. For different batch sizes, the CNN gives different accuracies. The batch size determines the number of iterations per epoch. Another important hyper parameter is the number of epochs. This determines accuracy and it has high influence on accuracy compared to other hyper parameters. The accuracy can be varied from 80% to 90% in vegetable dataset and 95% to 98% in the case of fruit dataset by increasing the number of epochs. The size of test dataset and train dataset also has very high influence on accuracies. The accuracy can be increased by using more number of images in train dataset. The computational time for model building is increased when the size of the train dataset increased and also number of epochs increased. The batch size of train dataset and test dataset also play a vital role in computational time. The Neural Network complexity is increased when more number of convolutional layers increased. If the number of layers increased, better accuracy result will obtain. At the same increasing the number of layers in CNN leads to more training time and also requires more time to build a model. The model .h5 size depends on the size of train datasets. But the memory requirement depends on the size of train dataset and CNN architecture complexity.

PROJECT FLOW:

A digital camera or similar devices are used to take images of different types, and then those are used to identify the affected area in leaves. Then different types of image- processing techniques are applied to them, the process those images, to get different and useful features needed for the purpose of analyzing later-Plant leaf disease identification is especially needed to predict both the quality and quantity of the First segmentation step primarily based on a mild polygonal leaf model is first achieved and later used to guide the evolution of an energetic contour.

Combining global shape descriptors given by the polygonal model with local curvature based features, the leaves are then classified overleaf datasets. In this research work introduce a method designed to deal with the obstacies raised by such complex images, for simple and plant leaves. A first segmentation step based on graph-cut approach is first performed and later used to guide the evolution of leaf boundaries, and implement classification algorithm to classify the diseases and recommend the fertilizers to affected

Figure.1 Proposed Architecture

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

PROJECT FLOW :

5.FLOWCHARTS

Proposed Solution :

- **The idea of the proposed solution uses Deep learning and Machine algorithm to classify leaves and identify the diseases and siggest the fertilizers. The deep learning process includes the MobileNetV2 and VGG19 training Models.**

- **Based on the leaf disease detected , the model recommendation for fertilizersfor the prevention. The farmers and researchers are the end users get benefied by the system.**

- **More accurate in others. The system is more robust corporating more image data sets with wider variations. This system also estimates the probability of infected plant.**
- **Plant growth can be enhanced. Ensure plants are getting supplied with every nutrient they need also and multiple cross in grow in every yields for every season. It also helps people's nutritional needs.**

Problem Solution Fit

- **This Learn and Build phase has proven to be the most important, parallel phase that successful startups follow. It contains the very first activity that startups should follow if they have an idea: Find prospective customers to talk to. Usually, this idea is already translated to a software product, which should always be a Minimum Viable Product (MVP) a version of the product that requires the least amount of development time with a minimum amount of effort.**

- **An MVP is based on requirements desired by potential customers, but to obtain these requirements, the startup should talk as early as possible with those customers. The startup then requires to prioritise the 'must haves', which arethe minimum necessary requirements for the MVP. Once the MVP is ready for customer feedback, the second most important activity is performed by the startup foreveryone.**

REQUIREMENT ANALYSIS :

Functional Requirements

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Specific characteristics	It identifies the diseases especially rice bran diseases
FR-4	Functions	The proposed methods uses the SVM to classify tree leaves, identify the diseases and suggest the fertilizer.
FR-5	Fault tolerance	This study enables a possible prediction of crop yield from the historic data collected and offers a suggestion to farmers.
FR-6	Analyze	It helps us to classify the data based on the diseases, and data extracted from the classifier is used to predict soil and crop.

Non Functional Requirements

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Crop and fertilizer recommendation system help the farmer to identify the diseases.
NFR-2	**Security**	The proposed method combines two major aspects in farming , pest identification and insecticide recommendation.
NFR-3	**Reliability**	It is easy use so that health issues can be avoided.
NFR-4	**Performance**	Precision fertilizer and precision crops is mostly used. They used to predict the crop in artificial intelligence.
NFR-5	**Availability**	reduces the losses as ammonia , nitrate leaching, apply the right rate, apply accurately.
NFR-6	**Scalability**	If the soil is not replenised with nutrients through fertilizing ,crop yields will deteriorate over time.

PYTHON CODE:

```
from flask
import Flask,
render_template,
request, Markup
import numpy as np
import pandas as pd
from utils.disease import disease_dic
from utils.fertilizer import fertilizer_dic
import requests
import config
import pickle
import io
import torch
from torchvision import transforms
from PIL import Image
from utils.model import ResNet9
import os
disease_classes = ['Apple___Apple_scab',
 'Apple___Black_rot',
 'Apple___Cedar_apple_rust',
 'Apple___healthy',
 'Blueberry___healthy',
 'Cherry_(including_sour)___Powdery_mildew',
 'Cherry_(including_sour)___healthy',
'Corn_(maize)___Cercospora_leaf_spot
Gray_leaf_spot',
 'Corn_(maize)__Common_rust',
 'Corn_(maize)___Northern_Leaf_Blight',
 'Corn_(maize)___healthy',
 'Grape___Black_rot',
 'Grape__Esca(Black_Measles)',
 'Grape__Leaf_blight(Isariopsis_Leaf_Spot)',
 'Grape___healthy',
 'Orange__Haunglongbing(Citrus_greening)',
 'Peach___Bacterial_spot',
 'Peach___healthy',
 'Pepper,bell__Bacterial_spot',
 'Pepper,bell__healthy',
 'Potato___Early_blight',
 'Potato___Late_blight',
 'Potato___healthy',
 'Raspberry___healthy',
 'Soybean___healthy',
 'Squash___Powdery_mildew',
 'Strawberry___Leaf_scorch',
```

```
 'Strawberry___healthy',
 'Tomato___Bacterial_spot',
 'Tomato___Early_blight',
 'Tomato___Late_blight',
 'Tomato___Leaf_Mold',
 'Tomato___Septoria_leaf_spot',
 'Tomato___Spider_mites Two-spotted_spider_mite',
 'Tomato___Target_Spot',
 'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
 'Tomato___Tomato_mosaic_virus',
 'Tomato___healthy']
disease_model_path = 'models/plant_disease_model.pth'
disease_model = ResNet9(3, len(disease_classes))
disease_model.load_state_dict(torch.load(
disease_model_path, map_location=torch.device('cpu')))
disease_model.eval()
crop_recommendation_model_path = 'models/RandomForest.pkl'
crop_recommendation_model = pickle.load(
open(crop_recommendation_model_path, 'rb'))
def weather_fetch(city_name):
 api_key = config.weather_api_key
 base_url = "http://api.openweathermap.org/data/2.5/weather?"
 complete_url = base_url + "appid=" + api_key + "&q=" + city_name
 response = requests.get(complete_url)
 x = response.json()
 if x["cod"] != "404":
 y = x["main"]
 temperature = round((y["temp"] - 273.15), 2)
 return temperature
 else:
 return None
def predict_image(img, model=disease_model):
 transform = transforms.Compose([
 transforms.Resize(256),


 print('Image to gray')
 x=np.expand_dims(x,axis=0)
 plant=request.form['plant']
 if(plant=='vegetable'):
 model=load_model("vegitable.h5")
 y=np.argmax(model.predict(x),axis=1)
 df=pd.read_excel('precautions_veg.xlsx')
 if(plant=='fruit'):
 model=load_model('fruit.h5')
 y=np.argmax(model.predict(x),axis=1)
```

```
 df=pd.read_excel('precautions_fruits.xlsx')
 return df.iloc[y[0]]['caution']
if _name_=='_main_':
 temp.run(debug=False
```

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>home page</title>
 <style>
 body{
 margin: 0;
 padding: 0;
 }
 .container{
 padding: 30px 70px 30px 70px;


 left: 20px;
 right:20px;
 background-color:rgb(163, 192, 120);
 font-size: 20pt;
 font-family: 'Times New Roman';

 }
 .card{
 font: optional;
 display: flex;

 }
 #h1{
 font-size: 50pt;
 }
 .menu{

 background-color:black;


 }
```

```
 #abc{
 color: white;
 }

 </style>
</head>
<body><div class="menu">
 <ul>
<li id="abc"> plant Disease
<a href="firstpage.html" id="abc">
home</a>
<a>
href="predict.html" id="abc">predict</a></li></ul>

 <div class="container" >

 <h1 id="h1"><center><b> Detect if your plant is infected!!
</b></center></h1>
 <div class="card" >
 <p >
</p>
 <img src="img.jpg" height="300" width="300">
 </div>
 </div>
 </div>
</body>
</html>
```

TEST THE FRUIT DATASET:


```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
```

},
"cells": [
{
"cell_type": "markdown",
"source": [
"# Team ID : PNT2022TMID38633\n",
"# Testing the Fruit dataset"
],
"metadata": {
"id": "axIa6Hz2Nq_S"
}
},
{
"cell_type": "code",
"source": [
"from keras.preprocessing import image\n",
"from tensorflow.keras.preprocessing.image import img_to_array\n",
"from tensorflow.keras.preprocessing import image\n",
"from tensorflow.keras.models import load_model\n",
"import numpy as nps"
],
"metadata": {
"id": "MdEVyGfTAbl-"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"model=load_model(r'/content/fruit.h5')"
],
"metadata": {
"id": "m_Xs4zCZAc0G"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"import numpy as np\n",
"from tensorflow.keras.models import load_model\n",
"from tensorflow.keras.preprocessing import image"
],
"metadata": {

"id": "dc5TlYMOAyYP"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"img=image.load_img(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Peach___Bacterial_spot/002eddd0-b6b3-474c-be08-423e53e24f82___Rutg._Bact.S 1955.JPG',grayscale=False,target_size=(128,128))"
],
"metadata": {
"id": "hkchp0KFA4BH"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"img"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 145
},
"id": "bbrjxYYVBEOo",
"outputId": "422ce069-29c4-4567-8ae4-2d1e0a07b969"
},
"execution_count": null,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"<PIL.Image.Image image mode=RGB size=128x128 at 0x7FDD95F65F90>"
],

},
"metadata": {},
"execution_count": 26
}
]
},

{
  "cell_type": "code",
  "source": [
    "x=image.img_to_array(img)\n",
    "x=nps.expand_dims(x,axis=0)"
  ],
  "metadata": {
    "id": "vf_RYp_sBH6X"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "pred=(model.predict(x) > 0.5).astype(\"int32\")"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "ChVQFCil0aWQ",
    "outputId": "21243be6-25f1-43ba-c5b3-fa28fac9003b"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "1/1 [==============================] - 0s 103ms/step\n"
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "pred"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "G_RYIxCi0gGX",
    "outputId": "2b121000-3886-45ed-a8d5-106a584d8ea7"

```
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([[0, 0, 0, 0, 1, 0]], dtype=int32)"
        ]
      },
      "metadata": {},
      "execution_count": 29
    }
  ]
},
{
  "cell_type": "code",
  "source": [
  "x_test.class_indices"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "C5KGHPYVXm8_",
    "outputId": "0c419f63-070e-4543-9742-3ecb64258308"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "{'Apple___Black_rot': 0,\n",
          " 'Apple___healthy': 1,\n",
          " 'Corn_(maize)___Northern_Leaf_Blight': 2,\n",
          " 'Corn_(maize)___healthy': 3,\n",
          " 'Peach___Bacterial_spot': 4,\n",
          " 'Peach___healthy': 5}"
        ]
      },
      "metadata": {},
      "execution_count": 30
    }
  ]
},
```

{
"cell_type": "code",
"source": [
"img=image.load_img(r\"/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Peach___Bacterial_spot/002eddd0-b6b3-474c-be08-423e53e24f82___Rutg._Bact.S 1955.JPG\",target_size=(128,128))\n",
"x=image.img_to_array(img)\n",
"x=np.expand_dims(x,axis=0)\n",
"y=np.argmax(model.predict(x),axis=1)\n",
"index=['Apple___Black_rot', 'Apple___healthy', 'Corn_(maize)___healthy', 'Corn_(maize)___Northern_Leaf_Blight', 'Peach___Bacterial_spot', 'Peach___healthy']\n",
"index[y[0]]"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 53
},
"id": "Om3vOOrmBr0w",
"outputId": "73b4c3b6-8760-4969-cda3-460acd8f5892"
},
"execution_count": null,
"outputs": [
{
"output_type": "stream",
"name": "stdout",
"text": [
"1/1 [==========] - 0s 24ms/step\n"
]
},
{
"output_type": "execute_result",
"data": {
"text/plain": [
"'Peach___Bacterial_spot'"
],
"application/vnd.google.colaboratory.intrinsic+json": {
"type": "string"
}
},
"metadata": {},
"execution_count": 31
}
]
}
]

}

TEST THE VEG DATASET:


```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
   "colab": {
     "provenance": []
   },
   "kernelspec": {
     "name": "python3",
     "display_name": "Python 3"
   },
   "language_info": {
     "name": "python"
   }
 },
 "cells": [
   {
     "cell_type": "markdown",
     "source": [
       "# Team ID : PNT2022TMID38633\n",
       "# Testing the vegetable dataset"
     ],
     "metadata": {
       "id": "tiYy5faiOzRC"
     }
   },
   {
     "cell_type": "code",
     "source": [
       "from keras.preprocessing import image\n",
       "from tensorflow.keras.preprocessing.image import img_to_array\n",
       "from tensorflow.keras.preprocessing import image\n",
       "from tensorflow.keras.models import load_model\n",
       "import numpy as nps"
     ],
     "metadata": {
       "id": "JlGDwlK_rVqz"
     },
     "execution_count": null,
```

"outputs": []
},
{
"cell_type": "code",
"source": [
"model=load_model(r'/content/vegetable.h5')"
],
"metadata": {
"id": "MAgC5W2jrZor"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"import numpy as np\n",
"from tensorflow.keras.models import load_model\n",
"from tensorflow.keras.preprocessing import image"
],
"metadata": {
"id": "BGUqiK3X3NSo"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"img=image.load_img(r'/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/Veg-dataset/Veg-dataset/test_set/Tomato___Bacterial_spot/b0049dbf-cdef-4a85-b6e1-b37ed6bc4cfa___UF.GRC_BS_Lab Leaf 0696.JPG',grayscale=False,target_size=(128,128))"
],
"metadata": {
"id": "TubDLMr-s7On"
},
"execution_count": null,
"outputs": []
},
{
"cell_type": "code",
"source": [
"img"
],

"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/",

          "outputs": [
            {
              "output_type": "stream",
              "name": "stdout",
              "text": [
                "1/1 [==============================] - 0s 34ms/step\n"
              ]
            }
          ]
        },
        {
          "cell_type": "code",
          "source": [
            "pred"
          ],
          "metadata": {
            "colab": {
              "base_uri": "https://localhost:8080/"
            },
            "id": "G_RYIxCi0gGX",
            "outputId": "88f6b57a-86e9-478d-c011-7563b1141662"
          },
          "execution_count": null,
          "outputs": [
            {
              "output_type": "execute_result",
              "data": {
                "text/plain": [
                  "array([[0, 0, 0, 0, 0, 0, 0, 0, 1]], dtype=int32)"
                ]
              },
              "metadata": {},
              "execution_count": 64
            }
          ]
        },
        {
          "cell_type": "code",
          "source": [
          "x_test.class_indices"
          ],
          "metadata": {

"colab": {
"base_uri": "https://localhost:8080/"
},
"id": "C5KGHPYVXm8_",
"outputId": "120c40ad-72d2-4b4f-ec01-d3f7bb0217c1"
},
"execution_count": null,
"outputs": [
{
"output_type": "execute_result",
"data": {
"text/plain": [
"{'Pepper,_bell___Bacterial_spot': 0,\n",
" 'Pepper,_bell___healthy': 1,\n",
" 'Potato___Early_blight': 2,\n",
" 'Potato___Late_blight': 3,\n",
" 'Potato___healthy': 4,\n",
" 'Tomato___Bacterial_spot': 5,\n",
" 'Tomato___Late_blight': 6,\n",
" 'Tomato___Leaf_Mold': 7,\n",
" 'Tomato___Septoria_leaf_spot': 8}"
]
},
"metadata": {},
"execution_count": 69
}
]
},
{
"cell_type": "code",
"source": [
"img=image.load_img(r\"/content/drive/MyDrive/DataSet/Dataset Plant Disease/fruit-dataset/Veg-dataset/Veg-dataset/test_set/Tomato___Bacterial_spot/b0049dbf-cdef-4a85-b6e1-b37ed6bc4cfa___UF.GRC_BS_Lab Leaf 0696.JPG\",target_size=(128,128))\n",
"x=image.img_to_array(img)\n",
"x=np.expand_dims(x,axis=0)\n",
"y=np.argmax(model.predict(x),axis=1)\n",

"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']\n",
"index[y[0]]"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",

```
        "height": 53
      },
      "id": "jnlDVMCg3iAP",
      "outputId": "9879ba1e-2527-45db-c187-dc05ebaf45f5"
    },
    "execution_count": null,
    "outputs": [
      {
        "output_type": "stream",
        "name":  "stdout",
        "text": [
          "1/1 [==============================] - 0s 31ms/step\n"
        ]
      },
      {
        "output_type": "execute_result",
        "data": {
          "text/plain": [
            "'Tomato___Septoria_leaf_spot'"
          ],
          "application/vnd.google.colaboratory.intrinsic+json": {
            "type": "string"
          }
        },
        "metadata": {},
        "execution_count": 70
      }
    ]
  }
]
}
```

PROJECT DESIGN :

Solution & Technical Architecture

Data Flow Daigrams

User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Dashboard	USN-6	As a user, can see the task obtained for their project	I can register and open the dashboard	High	Sprint-1
Customer (Web user)	Mobile Responsiveness	USN-7	A story that produces working code that has actual, demonstrable user value.	The conditions that a software product must meet to be accepted by a user, a customer, or other systems.	Low	Sprint-2
Customer Care Executive	Authentication	USN-8	As user, want to invite their friends and they do their work.	Applying the minimum amount of fertilizer necessary to maximize yield in the current year.	Medium	Sprint-1
Administrator	Transaction	USN-9	As a user, should transact their current payments.	User should accept the transaction	low	Sprint-2
	Handling	USN-10	As a user, should handle their project with correct manner	Team member should be ready to handle	Low	Sprint-2
	Business Rules	USN-11	As a user should know about the business rules.	I can handle the business rules.	Medium	Sprint-1
	Certification Requirement	USN-12	As a user, should need the certificate.	I can work for my project to get the certificate.	High	Sprint-1

PROJECT PLANNING & SCHEDULING :

Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
Sprint-1	Model Creation and Training (Fruits)		Create a model which can classify diseased fruit plants from given images. I also need to test the model and deploy it on IBM Cloud	8	High	Lalith Kishore Hari haran Durga Gomathy
	Model Creation and Training (Vegetables)		Create a model which can classify diseased vegetable plants from given images	2	High	Lalith Kishore Hari haran Durga Gomathy

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points (Total)	Priority	Team Members
Sprint-2	Model Creation and Training (Vegetables)		Create a model which can classify diseased vegetable plants from given images and train onIBM Cloud	6	High	Lalith Kishore Hari haran Durga Gomathy
	Registration	USN-1	As a user, I can register by entering my email, password, and confirming my password or via OAuth API	3	Medium	Lalith Kishore Hari haran Durga Gomathy
	Upload page	USN-2	As a user, I will be redirected to a page where Ican upload my pictures of crops	4	High	Lalith Kishore Hari haran Durga Gomathy
	Suggestion results	USN-3	As a user, I can view the results and then obtainthe suggestions provided by the ML model	4	High	Lalith Kishore Hari haran Durga Gomathy
	Base Flask App		A base Flask web app must be created as an interface for the ML model	2	High	Lalith Kishore Hari haran Durga Gomathy
Sprint-3	Login	USN-4	As a user/admin/shopkeeper, I can log into the application by entering email & password	2	High	Lalith Kishore Hari haran Durga Gomathy
	User Dashboard	USN-5	As a user, I can view the previous results and history	3	Medium	Lalith Kishore Hari haran Durga Gomathy
	Integration		Integrate Flask, CNN model with Cloudant DB	5	Medium	Lalith Kishore Hari haran Durga Gomathy
Sprint-4	Dashboard (Admin)	USN-6	As an admin, I can view other user details and uploads for other purposes	2	Medium	Lalith Kishore Hari haran Durga Gomathy
	Dashboard (Shopkeeper)	USN-7	As a shopkeeper, I can enter fertilizer productsand then update the details if any	2	Low	Lalith Kishore Hari haran Durga Gomathy
	Containerization		Create and deploy Helm charts using Docker Image made before	2	Low	Lalith Kishore Hari haran Durga Gomathy

Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	30 Oct 2022
Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	15	06 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	10	20 Nov 2022

FINAL CODE:

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

app = Flask(__name__)
global sess

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



model = load_model(r"C:\Users\Sree Ram\OneDrive\Desktop\IBM Project\fruit.h5")
model1=load_model(r"C:\Users\Sree Ram\OneDrive\Desktop\IBM Project\vegetable.h5")


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


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

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

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

        f = request.files['image']


        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'Dataset Plant Disease', 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(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=True) import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, request, render_template, redirect, url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app = Flask(__name__)
global sess

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

```
model = load_model(r"C:\Users\Sree Ram\OneDrive\Desktop\IBM Project\fruit.h5")
model1=load_model(r"C:\Users\Sree Ram\OneDrive\Desktop\IBM Project\vegetable.h5")


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


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

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

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

        f = request.files['image']


        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'Dataset Plant Disease', 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(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=True)
```

Reports from JIRA

6.RESULTS

Final findings(output) of the project given below in the form of screenshot: Training and Testing of Fruit dataset

Figure.6.2 Test the Fruit dataset

Figure.6.1. Fit a model for Fruit dataset

Train and Test Vegetable dataset

Figure.6.3. Train the Vegetable dataset

Figure.6.4. Test the Vegetable dataset

Train and Test Vegetable dataset IBM Cloud

Due to CUH limit exceeds, I have downloaded the notebooks and opened in Jupyter notebook

(i). Fruit dataset

Figure.6.5. Training Fruit Dataset in IBM Cloud

Figure.6.6. Training Vegetable Dataset in IBM Cloud

Out put

https://harvestify.herokuapp.com/crop-recommend
Enter the value (example:50)
Phosphorous
Enter the value (example:50)
Pottasium
Enter the value (example:50)
ph level
Enter the value
Rainfall (in mm)
Enter the value
State
Select State
City
Predict

HARVESTIFY
Home
Fertilizer
Get informed advice on fertilizer based on soil
Nitrogen
Enter the value (example:50)
Phosphorous
Enter the value (example:50)
Pottasium
Enter the value (example:50)
Crop you want to grow
Select crop
Predict

Harvestify - Disease Detection
https://harvestify.herokuapp.com/disease-predict
HARVESTIFY
Home
Crop
Fertilizer
Disease
Find out which disease has been caught by your plant
Please Upload The Image
Choose File
No file chosen
Predict

Open
test › Peach__Bacterial_spot
Search Peach__Bacterial_spot
Organize
New folder
Desktop
.idea
Application Bu
Assignment_2
bank
Data Collectio
Data_raw
Final deliverabl
histogram_file
IBM
Image Process
intensity_transf
File name:
All files
Open
Cancel
Home
Crop
been caught by your plant
ne Image

Crop: Peach

Disease: Bacterial Spot

Cause of disease:

1. The disease is caused by four species of Xanthomonas (X. euvesicatoria, X. gardneri, X. perforans, and X. vesicatoria). In North Carolina, X. perforans is the predominant species associated with bacterial spot on tomato and X. euvesicatoria is the predominant species associated with the disease on pepper.
2. All four bacteria are strictly aerobic, gram-negative rods with a long whip-like flagellum (tail) that allows them to move in water, which allows them to invade wet plant tissue and cause infection.

7.ADVANTAGES & DISADVANTAGES

List of advantages

- The proposed model here produces very high accuracy of classification.
- Very large datasets can also be trained and tested.
- Images of very high can be resized within the proposed itself.

List of disadvantages

- For training and testing, the proposed model requires very high computational time.
- The neural network architecture used in this project work has high complexity.

8.APPLICATIONS

1. The trained network model used to classify the image patterns with high accuracy.
2. The proposed model not only used for plant disease classification but also for other image pattern classification such as animal classification.

3. This project work application involves not only image classification but also for pattern recognition.

9.CONCLUSIONS

The model proposed here involves image classification of fruit datasets and vegetable datasets. The following points are observed during model testing and training:

- The accuracy of classification increased by increasing the number of epochs.
- For different batch sizes, different classification accuracies are obtained.
- The accuracies are increased by increasing more convolution layers.
- The accuracy of classification also increased by varying dense layers.
- Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
- Accuracies are different while varying the size of the train and test datasets.

10.FUTURE SCOPE

The proposed model in this project work can be extended to image recognition. The entire model can be converted to application software using python to exe software. The real time image classification, image recognition and video processing are possible with help OpenCV python library. This project work can be extended for security applications such as figure print recognition, iris recognition and face recognition.

11.BIBILOGRAPHY

[1]. R Indumathi 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.

[2]. 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.

[3]. 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- 0056.

APPENDIX

A. Source Code (Jupyter notebook python code)

fruit.ipynb (due to limited page size the code vegetable.ipynb uploaded in github)

```
#!/usr/bin/env python

# coding: utf-8

# In[1]: pwd

# In[2]: cd E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset

# # Apply ImageDataGenerator functionality to Train and Test set

# # Preprocessing # In[3]: from keras.preprocessing.image import ImageDataGenerator train_datagen =
```

ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizonta l_fli p=True) test_datagen = ImageDataGenerator(rescale=1) # In[4]: pwd

In[5]: x_train = train_datagen.flow_from_directory('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruitdataset/train',target_size=(128,128),batch_size=32,class_mode='cate gorical')

In[6]: x_test=test_datagen.flow_from_directory('E:/IBM_MY_COURSE/Project/Datas et Plant Disease/fruit-dataset/fruit-dataset/test',target_size=(128,128), batch_size=32,class_mode='categorical') # # Import the models

In[7]: from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten

Initializing the models 10

In[8]: model=Sequential()

Add CNN Layers

In[9]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

In[10]: x_train.class_indices

Add Pooling layer

In[11]: model.add(MaxPool2D(pool_size=(2,2)))

Add Flatten layer # In[12]: model.add(Flatten())

Add Dense Layer

In[21]: model.add(Dense(40, kernel_initializer='uniform',activation='relu')) model.add(Dense(20, kernel_initializer='random_uniform',activation='relu'))

Add Output Layer # In[24]: model.add(Dense(6,activation='softmax', kernel_initializer='random_uniform'))

Compile the model # In[25]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accur acy']) # In[26]: len(x_train)

In[27]: 5384/32

Fit the Model

In[28]: model.fit_generator(x_train,steps_per_epoch=168,validation_data=x_test,validat ion_st eps=52,epochs=3)

Save the Model

In[29]: model.save("fruit.h5")

In[30]: ls

Test the Model

In[32]: from keras.preprocessing import image from tensorflow.keras.preprocessing.image import img_to_array from tensorflow.keras.models import load_model import numpy as np

In[33]: model = load_model("fruit.h5")

Test Apple_Healthy Class images

In[37]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Apple___healthy/00fca0da-2db3-481b-b98a9b67bb7b105c___RS_HL 7708.JPG',target_size=(128,128)) 11

In[39]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0)

In[40]: pred = model.predict_classes(x)

In[41]: pred

In[45]: index =['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_spot','Peach___healthy']

In[46]: print('the given image belogs to=',index[pred[0]])

Test Apple Black Rot class images # In[54]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Apple___Black_rot/0f3d45f4-e121-42cd-a5b6- be2f866a0574___JR_FrgE.S 2870.JPG',target_size=(128,128))

In[55]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]])

Test Corn Northern leaf Blight class images

In[56]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Corn_(maize)___Northern_Leaf_Blight/00a14441-7a62- 4034-bc40-b196aeab2785___RS_NLB 3932.JPG',target_size=(128,128))

In[57]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]])

#Test Com Healthy class images
image.load_img(E/IBM_MY_COURSE/Project/Dataset In[58]:
img Disease/fruitdataset/fruitdataset/test/Corn_(maize)__healthy/0a68ef5a-027c-41ae-b227-159dae77d3dd RS HL 7969 copy.jpg',target_size=(128,128))

In[59]:x=image.img_to_array(img) x-np.expand_dims(x.axis-0) pred = model predict_classes(x) print(the given image belogs to=',index/pred[0]])

Test Peach Bacterial spot class images In[60]: img
image.load_img("E:/IBM MY COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Peach Bacterial spot 00dde 106-692e-4c67-b2e8-569c924caf49 Rutg. Bact S 1228.JPG',target_size=(128,128)) 12

#In[61]: x-image.img_to_array(img) x-np.expand dims(x,axis=0)) pred = model predict classes(x) print(the given image belogs to index(pred[0]])

Test Peach Healthy class images

In[62]: img = image.load_img(E/IBM MY COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Peachhealthy/1a07ce54-f4fd-41cf- b088-144f6bf71859 Rutg HL 3543.JPG target_size=(128,128))

In[63]; x=image_img_to_array(img) x-np.expand_dims(x,axis=0)) pred =

model predict classes(x) print(the given image belogs to=',index[pred[0])

GITHUB I’D : https://github.com/IBM-EPBL/IBM-Project-50181-1660897941.git

PROJECT DEMO LINK : https://youtu.be/95YWnFOXdmc