PROJECT DOCUMENTATION

Fertilizer Recommendation System For Disease Prediction

Team Id:-PNT2022TMID26295

Submitted By:-

Tanya A – 211519205168 - Team Lead

Princy A – 211519205115

Kruthika M - 211519205081

Evagelin C - 211519205047

Table of Contents:-

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

1. INTRODUCTION:-

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

1.2 Purpose This project is used to test the fruits and vegetables samples and identify the different diseases. Also, this project recommends fertilizers for predicted diseases.

2. LITERATURE SURVEY:-

2.1 Existing problem Indumathi proposed a method for leaf disease detection and suggest fertilizers to cure leaf diseases. But the method involves less number of train and test sets which results in poor accuracy. Pandi selvi proposed asimple prediction method for soil-based fertilizer recommendation system for predicted crop diseases. This method gives less accuracy and prediction. Shiva reddyproposed an IoT based system for leaf disease detection and fertilizer recommendation which is based on Machine Learning techniques yields less 80 percentage accuracies.

2.2 Proposed solution

Proposed solution In this project work, a deep learning based neural network is used to train the collected datasets and test the same. The deep learning based neural network is CNN which gives more than 90% classification accuracies. By increasing the more number of dense layers and by modifying hyperparameters such as number of epochs, batch size, the accuracy rate can be increased to 95% to 98%.

REFERENCES:-

Survey 1 :
1.TITLE: CROFED - Crop and Fertilizer Recommendation and Disease diagnosis system using Machine Learning and Internet of Things
Year:2022
AUTHORS: TARANJEET SINGH1 , SAURABH ANAND2 , ANMOL SEHGAL3 , SIDDHESH MAHAJAN4 , PROF. PRANOTI KAVIMANDAN5
DESCRIPTION: The major problems that the farmers of our country are currently facing includes Crop Failure, Lack of adequate knowledge, Crop damage due to ignorance/carelessness, Lack of professional assistance, Inaccessibility to agro-tech solutions. CROFED will help the farmers to deal with these problems by providing following aids: Crop Recommendation system, Fertiliser suggestion system, Crop Disease Detection System. We will develop an IOT device that will examine the quality of soil and can also detect crop diseases on scanning the leaves of the crops. Soil testing is significant since it allows for the determination of soil fertility and hence crop prediction. Soil pH is a measure of the acidity and alkalinity in soils. pH levels range from 0 to 14, with 7 being neutral, below 7 acidic and above 7 alkaline. We have proposed a system which will have a device which gives pH value and we will estimate Nitrogen (N), Phosphorus (P) and Potassium (K) from the pH of that soil.

Survey 2 :
Title: Farmer's Assistant: A Machine Learning Based Application for Agricultural Solutions **Year:**2022
Authors: Shloka Gupta, Nishit Jain, Akshay Chopade, Aparna Bhonde
DESCRIPTION: Farmers face several challenges when growing crops like uncertain irrigation, poor soil quality, etc. Especially in India, a major fraction of farmers do not have the knowledge to select appropriate crops and fertilizers. Moreover, crop failure due to disease causes a significant loss to the farmers, as well as the consumers. While there have been recent developments in the automated detection of these diseases using Machine Learning techniques, the utilization of Deep Learning has not been fully explored. Additionally, such models are not easy to use because of the high-quality data used in their training, lack of computational power, and poor generalizability of the models. To this end, we create an open-source easy-to-use web application to address some of these issues which may help improve crop

production. In particular, we support crop recommendation, fertilizer recommendation, plant disease prediction, and an interactive news-feed. In addition, we also use interpretability techniques in an attempt to explain the prediction made by our disease detection model.

Survey 3 :
Title: Soil Based Fertilizer Recommendation System for Crop Disease Prediction System
YEAR:2021
 AUTHORS: Dr.P. Pandi Selvi1 , P. Poornima2
 DESCRIPTION: Agriculture is the main aspect for the economic development of a country. Agriculture is the heart and life of most Indians. But in recent days, the field was going down due to various natural calamities. In order to overcome the problem, various issues in this field need to be addressed. The soil type, fertilizer recommendation, diseases in plants and leaves. All these features need to be considered. Our proposed system was organized in such a way, to analyze the soil type, diseases in the leaves and finally to recommend the appropriate fertilizer to the farmers, that may be of great help to them. Plant disease, especially on leaves, is one of the major factors that reduce the yield in both quality and quantity of the food crops. Finding the leaf disease is an important role to preserve agriculture. Smart analysis and Comprehensive prediction model in agriculture helps the farmer to yield right crop at the right time. The main benefits of the proposed system are as follows: Yield right crop at the right time, Balancing the crop production, control plant disease, Economic growth, and planning to reduce the crop scarcity.

Survey 4 :
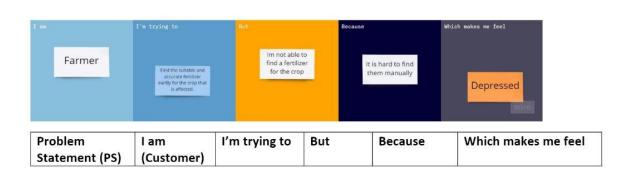**Title:** IOT based Crop Recommendation, Crop Disease Prediction and Its Solution
YEAR:2020
Authors: Rani Holambe1, Pooja Patil2, Padmaja Pawar3, Saurabh Salunkhe 4, Mr. Hrushikesh Joshi5
DESCRIPTION: Agriculture plays a vital role in India. India is the world's largest producer of different crops but still, it uses traditional framing methods therefore crop yield becomes down. Hence, with the introduction of newer seed varieties, new methods of agriculture crop production have increased. But without using the smarter ways, the agricultural field still having an imperfection. And due to these farmers need a smarter way to increase crop production. The system used different sensors that measured pH level, soil moisture, temperature, and humidity. After designing the system for pH, soil moisture, humidity, and temperature sensor, the next step is to build a model for crop recommendation. The databasewill compose of recommended minimum and maximum values of temperature, humidity, pH, soil moisture recommended for growing crops commonly planted in the country.Hence, to maximize the crop yield some smart methods came into the picture used in IoT and Machine Learning. In this paper, we will review the algorithms like Random Forest, Decision Tree, ANN to get better accuracy for the system.

2.3.PROBLEM STATEMENT DEFINITION:-

Agriculture plays an important role in economy as well as it is considered the backbone of many countries. Agriculture is the art of growing plants. 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. Therefore, early and accurate prediction of crop disease is necessary. Hence, a system is to be developed to identify plant disease and recommend the fertilizers.

Example:

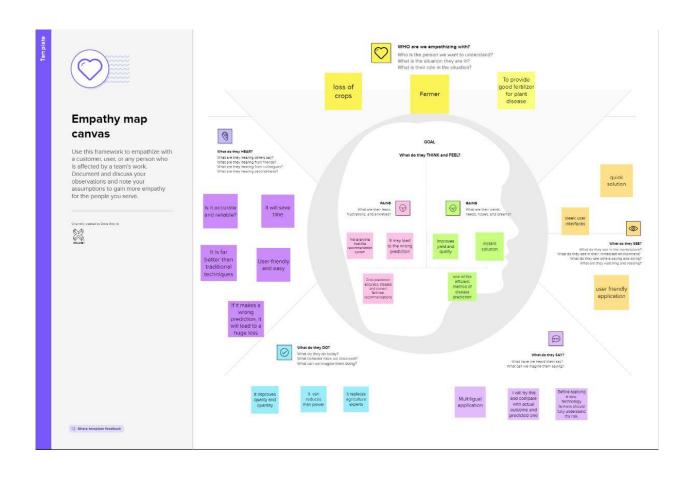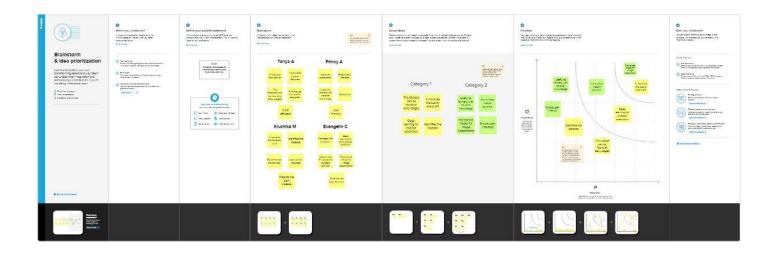| Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel

| PS-1 | Farmer | Find the suitable and accurate fertilizer earlily for the crop that is affected. | Im not able to find a fertilizer for the crop. | It is hard to find them manually. | Depressed |

3.IDEATION PHASE & PROPOSED SOLUTION :

3.1 Empathy Map Canvas :

3.2 IDEATION AND BRAINSTORMING:-

3.3 PROPOSED SOLUTION:-

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Identifying the disease in plants is hard to find.Because,disease in plants reduced the quantity and quality of the productivity.
2.	Idea / Solution description	One of the solution of the problem is to identify the plant disease in early stage using the correct fertilizer.
3.	Novelty / Uniqueness	This application can suggest good fertilizer for the plant disease by recognizing the images.
4.	Social Impact / Customer Satisfaction	It helps the farmers to identify the disease in early stage and increase the quality and quantity of crops in efficient way.
5.	Business Model (Revenue Model)	The application is recommended to the farmers in the subscription basis.
6.	Scalability of the Solution	This application can be improved by introducing the online purchases of crops,fertilizer easily.

3.4 PROPOSED SOLUTION FIT

Project Title: Fertilizer Recommendation System For Disease Prediction   **Project Design Phase-I – Solution Fit**   **Team ID:** PNT2022TMID26295

Define CS,fit into CC			Explore AS, differentiate
1. CUSTOMER SEGMENT(S) CS Farmers are the first customers for this kind of Application. Farmers can easily use this Applications and get suggestions for fertilizers to be used correctly	**6. CUSTOMER CONSTRAINTS** CC Availabilty of good networks.Capturing the image in a required pixels to get a accurate prediction of disease in the plant.	**5. AVAILABLE SOLUTIONS** AS People judge the disease in plants by identifying through the change of leaf's quality	

Focus on J&P, tap into BE, understand RC			Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P
 	
This application focuses on helping the farmers who needs a better recommendation system for fertilizers for the infected plants. Identifying the plant diseases is one of the greatest problem here.
	9. PROBLEM ROOT CAUSE RC
	
	Various diseases on the plants can lead to reducing the quality of it and the production of the plants can also decrease. The insects on the plants can spread the disease.
		7. BEHAVIOUR BE
		
		Directly:
		        Farmers can easily identify the disease by the application and they will not need any additional knowledge for predicting the diseases.
		
		Indirectly:
		        Farmers can get the results through online immediately.	

3. TRIGGERS TR
 	
Seeing their crops being affected by the diseases and facing the huge loss in quality and production.
 	
4. EMOTIONS: BEFORE / AFTER EM
 	
Before: Losing slef-confidence, Distress
 	
After: Gaining self-confidence, Relief
	10. SOLUTION SL
	
	Using the fertilizer is one of the solution for the plant diseases. Our Application use the image of the infected plants by identifying the disease and suggest the suitable fertilizers for that disease.
		8.CHANNELS of BEHAVIOUR CH
		Online:
		        Basic knowledge on the plant and fertilizers
		
		Offline:
		        People try to identify the disease by the quality of the leaves.

4.REQUIREMENT ANALYSIS:-

Functional Requirements:

Following are the functional requirements of the proposed solution.

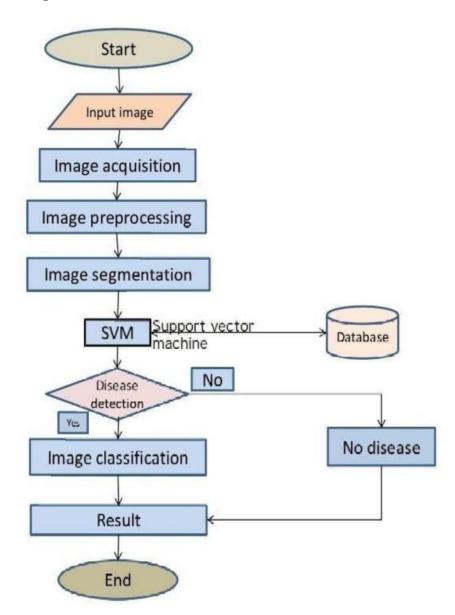| FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User confirmation	Confirmation via OTP Confirmation via Email
FR-3	Capturing image	Capture the image of the leaf And check the parameter of the captured image .
FR-4	Image processing	Upload the image for the prediction of the disease in the leaf.
FR-5	Leaf identification	Identify the leaf and predict the disease in leaf.

Non-functional Requirements:

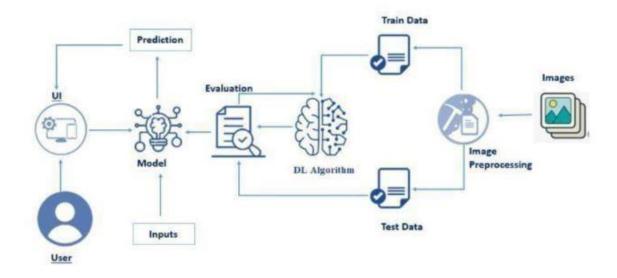Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User can know the range before entering their own report and can easily navigate.
NFR-2	**Security**	The user data is very secured .It is accessed only the user and company
NFR-3	**Reliability**	The result or the report given is very reliable
NFR-4	**Performance**	Deals with the front-end load time and fast return of result of the report
NFR-5	**Availability**	All the reports and result will be available in user's log in
NFR-6	**Scalability**	Collecting response from user

5.PROJECT DESIGN:-
5.1 DATA FLOW DIAGRAM:-

Data Flow Diagrams:

5.2 TECHNOLOGY ARCHITECTURE:-

Table-1:Components& Technologies:

1.	User Interface	Web UI	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application logic-1	Image Preprocessing	Python
3.	Application logic-2	CNN Model	IBM Watson STT service
4.	Application logic-3	Web UI Application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	File storage	File storage requirements	IBM DB2, IBM Cloudant etc.
7.	External Api	Purpose of External API used in the application	IBM Block Storage or Other Storage Service or Local Filesystem

8.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
9.	Infrastructure (Server)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2:Application characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Flask Frameworks
2.	Security Implementations	CSRF Protection,Secure Flag For Cookies	Flask-WTF, Session Cookie Secure
3.	Scalable Architecture	Micro-Services	Micro Web Application FrameworkBy Flask

5.3 USER STORIES:-

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	I can register as a user on the website with eitheran email address or a phone number and password.	I can createmy account.	High	Sprint-3
	Login	USN-2	With theprovided Login credentials, I canaccessthe website as a user.	I can log in andaccess myaccount .	High	Sprint-3
	Upload image	USN-3	I can post my data as a userin formats likepdf and doc.	I can uploadmy data.	Medium	Sprint-3

Customer (Web user)	Admin Login	USN-4	. As a user, I can login to web dashboard just Like website dashboard	I can log in and analyze the user data.	High	Sprint-3
	Data collection	USN-5	As a user, I can login to myweb dashboard with the login credentials	I can collect the dataset.	Low	Sprint-1
	Create model	USN-6	As a user, I can view the web application where i can upload my images for getting the suggestion of the fertilizer	I can create andtrain the model.	High	Sprint-1
	Test the model	USN-7	As a user, the fertilizer recommended to me Is in high accurate	I can testthe model.	High	Sprint-2
Administrator	Diagnosis	USN-8	I can access the application's diagnosisresults as a userand continue with treatments..	I can access my dashboard	High	Sprint-2

6.PROJECT PLANNING AND SCHEDULING:-

6.1 SPRINT PLANNING AND ESTIMATION:-

Milestone:
Modern Technology are increasing and optimizing the Performance of the Artificial Intelligences (AI) Model. Based Crop Yield Disease Prediction System, is helpful for farmers to prevent the crop from the various Disease which can identify the Disease with in a process of capturing the Image at the plant and Machine Learning Algorithm will give affected Disease Name. In this Project Milestone will be given the Best Solution for the farmer using the complete friendly and simple user interface web application to fetching the solution by own. In addition, process we are planned to add a valid Module that is Fertilizer recommendation for the Specific Disease. It can give both Artificial fertilizer and Natural Fertilizer in suggestion manner.

Activity List:
In Project Management Planning is an Important task to scheduling the phrase of the project to the Team Member. In this Activity can shows the various activity are allocated and Done by the Team Members! In Project we can Split into the Four Step of Phrases are Phrase 1: Information Collection and Requirement Analysis. Phrase 2: Project Planning and Developing Modules. Phrase 3: Implementing the High Accuracy Deep Learning Algorithm to Perform. Phrase 4: Deploying the Model on Cloud and Testing the Model and UI Performance

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

Velocity:

Imagine we have a 10-daysprint duration, and the velocityof the team is 20 (points per sprint). Let's calculate the team'saverage velocity (AV)periteration unit (story points per day).

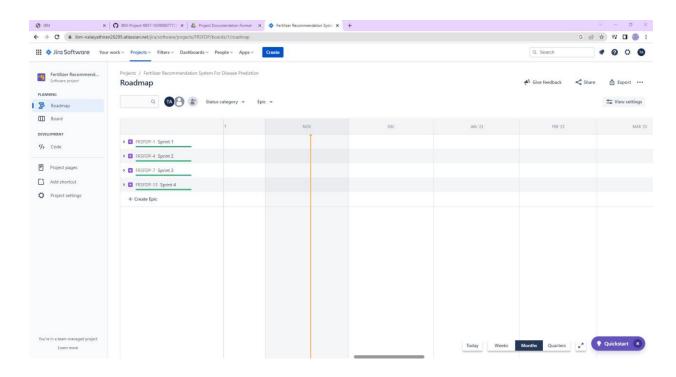$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

AV=20/6=3.33points per day.

6.2 Burn Down Chart & JIRA :

Burndown Chart

A burn down chart plots the amount of work remaining to perform against the amount of time. In agile software development approaches like Scrum, it is frequently employed. Burn down charts, however, can beused for any project that makes observable progress over time.

JIRA SCREENSHOTS:-

JIRA Folder is created to show the Scrum methodologies and Burn Down chart progress.

7.CODING AND SOLUTIONING:-

Feature 1:-
We have devloped a website which authenticates users and help them upload and predict the disease.

Feature 2:-
We have devloped a multilayer deep convolutional nueral network that classifies the user image of leaves to which extense has the disease has been

affected.The model will classify the images into and report them on asking for prediction.

8.TESTING:-
8.1 TEST CASES:-
8.2 USER ACCEPTANCE TESTING:-

1. Purpose of Document:-

The purpose of this document is to briefly explain the test coverage and open issues of the [Fertilizer Recommendation system for plant disease prediction] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis:-

This shows how many bugs were fixed or closed at each severity level and how they were fixed.

Resolution	Severity 1	Severity 2	Severity 3	Severity4	Subtotal
Leaf spots	10	4	2	3	19
Mosaic leaf pattern	9	6	3	6	24
Misshapen leaves	2	7	0	1	10
Yellow leaves	11	4	3	20	38
Fruit rots	3	2	1	0	6
Fruit spots	5	3	1	1	10
Blights	4	5	2	1	12
Totals	44	31	13	32	119

3.Test-CaseAnalysis

This report shows the number of test cases that have passed, failed,and untested.

Section	TotalCases	Not Tested	Fail	Pass
Leaf spots	17	0	0	17
Mosaic leaf pattern	51	0	0	51
Misshapen leaves	20	0	0	20

Yellow leaves	7	0	0	7
Fruit rots	9	0	0	9
Fruit spots	4	0	0	4
Blights	2	0	0	2

9.RESULTS:-
9.1 Performance Metrics:-

Model Performance Testing:

S. NO	Parameter	Values	Screenshot
1.	Model Summary	**Total params: 38,160,755** **Trainable params: 38,160,755** **Non-trainable params: 0**	
2.	Accuracy	Training Accuracy – **0.9031** Validation Accuracy – **loss 2081.9343**	loss: 0.2834 - accuracy: 0.9031
3.	Confidence Score(Only Yolo Projects)	Class Detected - Confidence Score -	-- -- -- --

10.ADVANTAGES AND DISADVANTAGES:-
10.1 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.

10.2 DISADVANTAGES:-

●   For training and testing, the proposed model requires very high computationaltime.

● The neural network architecture used in this project work has high complexity.

11.CONCLUSION:-

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.

12.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.

13.APPENDIX:-

app.py:-

```
!unzip '/content/drive/MyDrive/ibm dataset/Fertilizers_Recommendation_ System_For_Disease_ Prediction.zip'
```

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,z
oom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)

x_train=train_datagen.flow_from_directory('/content/Dataset Plant
Disease/Veg-dataset/Veg-
dataset/train_set',target_size=(128,128),batch_size=2,class_mode=
'categorical')
x_test=test_datagen.flow_from_directory('/content/Dataset Plant
Disease/Veg-dataset/Veg-
dataset/test_set',target_size=(128,128),batch_size=2,class_mode='
categorical')

from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten

from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,z
oom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)

x_train=train_datagen.flow_from_directory('/content/Dataset Plant
Disease/Veg-dataset/Veg-
dataset/train_set',target_size=(128,128),batch_size=16,class_mode
='categorical')
x_test=test_datagen.flow_from_directory('/content/Dataset Plant
Disease/Veg-dataset/Veg-
dataset/test_set',target_size=(128,128),batch_size=16,class_mode=
'categorical')

model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activati
on='relu'))
```

```python
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=300,kernel_initializer='uniform',activation
='relu'))

model.add(Dense(units=150,kernel_initializer='uniform',activation
='relu'))
model.add(Dense(units=75,kernel_initializer='uniform',activation=
'relu'))
model.add(Dense(units=9,kernel_initializer='uniform',activation='
softmax'))
model.compile(loss='categorical_crossentropy',optimizer="adam",me
trics=["accuracy"])
model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_
test,validation_steps=27)

model.save('fruit.h5')

model.summary()

from keras.preprocessing import image
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as nps

model=load_model('fruit.h5')

img=image.load_img('/content/Dataset Plant Disease/fruit-
dataset/fruit-dataset/test/Apple___healthy/011d02f3-5c3c-4484-
a384-b1a0a0dbdec1___RS_HL
7544.JPG',grayscale=False,target_size=(128,128))

img

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

pred=(model.predict(x) > 0.5).astype("int32")

pred

import requests
```

```python
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__)
model = load_model("fruit.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['images']
    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=="fruit"):
      preds=model.predict_classess(x)
      print(preds)
      df=pd.read_excel('precautions-veg.xlsx')
      print (df.iloc[preds[0]]['cautions'])
    else:
      pred=model1.predict_classes(x)
      df=pd.read_excel('precautions-fruits.xlsx')
```

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

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

home.html:-

```html
<!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, 172, 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>
            
           
           
           
            &nb
sp;            
            &nb
sp;         

            &nb
sp;            
  
 <li id="abc"> plant Disease
Prediction          &nbsp
;            
           
            &nb
sp;            
            &nb
sp;            
            &nb
sp;            
            &nb
sp;            
            &nb
sp;            
            &nb
sp;            
            &nb
sp;            
      <a href="firstpage.html" id="abc">
home</a>
```

```html
        <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 > Agriculture is one of the major sectors works 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
laboratries.This application helps farmers in detecting the diseases by
observing the spots on the leaves ,which inturn saves effort and labor
costs.</p>
<img src="E:\Tany\A IBM\template\fvp.jpg" height="300" width="300">
</div>
</div>
</div>
```

predict.html:-

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Plant Disease Prediction</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>
  <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
```

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

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

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

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

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

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

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

.container {
  margin-top:40px;
  padding: 16px;
}
select {
      width: 100%;
      margin-bottom: 10px;
      background: rgba(255,255,255,255);
      border: none;
      outline: none;
      padding: 10px;
      font-size: 13px;
      color: #000000;
```

```
        text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
        border: 1px solid rgba(0,0,0,0.3);
        border-radius: 4px;
        box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
        -webkit-transition: box-shadow .5s ease;
        -moz-transition: box-shadow .5s ease;
        -o-transition: box-shadow .5s ease;
        -ms-transition: box-shadow .5s ease;
        transition: box-shadow .5s ease;
}


</style>
</head>

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

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


  </div>
</div>
<div class="container">
        <div id="content" style="margin-top:2em">
                <div class="container">
                  <div class="row">
                        <div class="col-sm-6 bd" >

                          <br>
                                <img src="E:\Tany\A IBM\template\predict.jpg"
style="height:450px;width:550px"class="img-rounded" alt="Gesture">
                        </div>
                        <div class="col-sm-6">
                                <div>
                                        <h4>Drop in the image to get the prediction
</h4>
                        <form action = "" id="upload-file" method="post"
enctype="multipart/form-data">
                                <select name="plant">
```

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

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

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

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

                </div>
                        </div>

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

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

</html>
```

Python Notebook screenshots:-

```
In [1]:  from keras.preprocessing.image import ImageDataGenerator
         train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
         test_datagen=ImageDataGenerator(rescale=1)
```

```
In [2]:  x_train=train_datagen.flow_from_directory(r'C:\Users\princ\OneDrive\Desktop\Dataset Plant Disease\Veg-dataset\Veg-dataset\train_set',target_size=(128,
         x_test=test_datagen.flow_from_directory(r'C:\Users\princ\OneDrive\Desktop\Dataset Plant Disease\Veg-dataset\Veg-dataset\test_set',target_size=(128,128

         Found 11386 images belonging to 9 classes.
         Found 3416 images belonging to 9 classes.
```

```
In [3]:  from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import Convolution2D
         from keras.layers import MaxPooling2D
         from keras.layers import Flatten
```

```
In [4]:  model=Sequential()
```

```
In [5]:  model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

```
In [6]:  model.add(MaxPooling2D(pool_size=(2,2)))
```

```
In [7]:  model.add(Flatten())
```

```
In [8]:  model.add(Dense(units=300,kernel_initializer='uniform',activation='relu'))
```

```
In [8]:   model.add(Dense(units=300,kernel_initializer='uniform',activation='relu'))

In [9]:   model.add(Dense(units=150,kernel_initializer='uniform',activation='relu'))

In [10]:  model.add(Dense(units=75,kernel_initializer='uniform',activation='relu'))

In [11]:  model.add(Dense(units=9,kernel_initializer='uniform',activation='softmax'))

In [12]:  model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])

In [13]:  model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,validation_steps=27)

          Epoch 1/20
          89/89 [==============================] - 37s 407ms/step - loss: 1.8913 - accuracy: 0.2963 - val_loss: 201.2470 - val_accuracy: 0.3634
          Epoch 2/20
          89/89 [==============================] - 34s 386ms/step - loss: 1.3260 - accuracy: 0.5105 - val_loss: 144.9129 - val_accuracy: 0.5104
          Epoch 3/20
          89/89 [==============================] - 31s 348ms/step - loss: 1.0517 - accuracy: 0.6204 - val_loss: 452.8666 - val_accuracy: 0.2465
          Epoch 4/20
          89/89 [==============================] - 30s 340ms/step - loss: 0.9506 - accuracy: 0.6570 - val_loss: 1062.1256 - val_accuracy: 0.2801
          Epoch 5/20
          89/89 [==============================] - 33s 368ms/step - loss: 0.7732 - accuracy: 0.7268 - val_loss: 713.5864 - val_accuracy: 0.3264
          Epoch 6/20
          89/89 [==============================] - 33s 372ms/step - loss: 0.6780 - accuracy: 0.7574 - val_loss: 1175.1545 - val_accuracy: 0.2801
          Epoch 7/20
          89/89 [==============================] - 31s 352ms/step - loss: 0.6116 - accuracy: 0.7745 - val_loss: 1305.8286 - val_accuracy: 0.2743
          Epoch 8/20
          89/89 [==============================] - 29s 323ms/step - loss: 0.5763 - accuracy: 0.7903 - val_loss: 1302.3727 - val_accuracy: 0.2269
          Epoch 9/20
          89/89 [==============================] - 30s 342ms/step - loss: 0.5767 - accuracy: 0.7946 - val_loss: 1277.9509 - val_accuracy: 0.3056

          Epoch 10/20
          89/89 [==============================] - 31s 348ms/step - loss: 0.5556 - accuracy: 0.8013 - val_loss: 1235.4139 - val_accuracy: 0.3113
          Epoch 11/20
          89/89 [==============================] - 32s 355ms/step - loss: 0.4403 - accuracy: 0.8392 - val_loss: 1530.7563 - val_accuracy: 0.3461
          Epoch 12/20
          89/89 [==============================] - 31s 344ms/step - loss: 0.4323 - accuracy: 0.8522 - val_loss: 1574.8284 - val_accuracy: 0.3125
          Epoch 13/20
          89/89 [==============================] - 29s 320ms/step - loss: 0.4533 - accuracy: 0.8343 - val_loss: 1657.8656 - val_accuracy: 0.2627
          Epoch 14/20
          89/89 [==============================] - 29s 324ms/step - loss: 0.4202 - accuracy: 0.8498 - val_loss: 1603.1924 - val_accuracy: 0.3021
          Epoch 15/20
          89/89 [==============================] - 26s 289ms/step - loss: 0.3590 - accuracy: 0.8792 - val_loss: 1485.8334 - val_accuracy: 0.3623
          Epoch 16/20
          89/89 [==============================] - 25s 281ms/step - loss: 0.3680 - accuracy: 0.8694 - val_loss: 2185.7332 - val_accuracy: 0.2708
          Epoch 17/20
          89/89 [==============================] - 24s 270ms/step - loss: 0.3488 - accuracy: 0.8792 - val_loss: 2317.3721 - val_accuracy: 0.2870
          Epoch 18/20
          89/89 [==============================] - 24s 264ms/step - loss: 0.3258 - accuracy: 0.8902 - val_loss: 1627.2305 - val_accuracy: 0.3009
          Epoch 19/20
          89/89 [==============================] - 26s 293ms/step - loss: 0.3268 - accuracy: 0.8869 - val_loss: 1958.1587 - val_accuracy: 0.2546
          Epoch 20/20
          89/89 [==============================] - 28s 313ms/step - loss: 0.2834 - accuracy: 0.9031 - val_loss: 2081.9343 - val_accuracy: 0.3090

Out[13]:

In [14]:  model.save('vegetabledata.h5')
```

```
89/89 [==============================] - 26s 293ms/step - loss: 0.3268 - accuracy: 0.8869 - val_loss: 1958.1587 - val_accuracy: 0.2546
Epoch 20/20
89/89 [==============================] - 28s 313ms/step - loss: 0.2834 - accuracy: 0.9031 - val_loss: 2081.9343 - val_accuracy: 0.3090
```

Out[13]:

In [14]:
```python
model.save('vegetabledata.h5')
```

In [15]:
```python
model.summary()
```

```
Model: "sequential"
_____
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 32)      896
_____
max_pooling2d (MaxPooling2D) (None, 63, 63, 32)        0
_____
flatten (Flatten)            (None, 127008)            0
_____
dense (Dense)                (None, 300)               38102700
_____
dense_1 (Dense)              (None, 150)               45150
_____
dense_2 (Dense)              (None, 75)                11325
_____
dense_3 (Dense)              (None, 9)                 684
=================================================================
Total params: 38,160,755
Trainable params: 38,160,755
Non-trainable params: 0
_____
```

Fruit:

In [1]:
```python
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)
```

In [2]:
```python
x_train=train_datagen.flow_from_directory(r'C:\Users\princ\OneDrive\Desktop\Dataset Plant Disease\fruit-dataset\fruit-dataset\train',target_size=(128,
x_test=test_datagen.flow_from_directory(r'C:\Users\princ\OneDrive\Desktop\Dataset Plant Disease\fruit-dataset\fruit-dataset\test',target_size=(128,128
```

```
Found 5384 images belonging to 6 classes.
Found 1686 images belonging to 6 classes.
```

In [3]:
```python
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
```

In [4]:
```python
model=Sequential()
```

In [5]:
```python
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

In [6]:
```python
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [7]:
```python
model.add(Flatten())
```

In [8]:
```python
model.add(Dense(units=40,kernel_initializer='uniform',activation='relu'))
model.add(Dense(units=70,kernel_initializer='random_uniform',activation='relu'))
model.add(Dense(units=6,kernel_initializer='random_uniform',activation='softmax'))
```

In [9]:
```python
model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])
```

```
In [10]:   model.fit(x_train,steps_per_epoch=168,epochs=3,validation_data=x_test,validation_steps=52)

           Epoch 1/3
           168/168 [==============================] - 78s 460ms/step - loss: 0.8677 - accuracy: 0.6741 - val_loss: 62.5237 - val_accuracy: 0.8191
           Epoch 2/3
           168/168 [==============================] - 28s 165ms/step - loss: 0.3775 - accuracy: 0.8707 - val_loss: 86.6816 - val_accuracy: 0.8071
           Epoch 3/3
           168/168 [==============================] - 28s 166ms/step - loss: 0.2731 - accuracy: 0.9088 - val_loss: 231.7529 - val_accuracy: 0.7362

Out[10]:

In [11]:   model.save('fruitdata.h5')

In [12]:   model.summary()

           Model: "sequential"
           _____
           Layer (type)                Output Shape              Param #
           =================================================================
           conv2d (Conv2D)             (None, 126, 126, 32)      896
           _____
           max_pooling2d (MaxPooling2D) (None, 63, 63, 32)       0
           _____
           flatten (Flatten)           (None, 127008)            0
           _____
           dense (Dense)               (None, 40)                5080360
           _____
           dense_1 (Dense)             (None, 70)                2870
           _____
           dense_2 (Dense)             (None, 6)                 426
           =================================================================
           Total params: 5,084,552
           Trainable params: 5,084,552
           Non-trainable params: 0
           _____
```

GITHUB LINK:- https://github.com/IBM-EPBL/IBM-Project-9857-1659080777

DEMO LINK: https://drive.google.com/file/d/1J9jY8OijfUu-zAPiRvT5xFQzofEK6vMH/view?usp=share_link