

**M.A.M. College of Engineering**,

Tiruchirappalli is approved by the AICTE, New Delhi and affiliated to Anna University, Chennai.

**Department of Computer Science and Engineering**

# REPORT ON

HX 8001 PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP (Nalaiya Thiran Program)

**PROJECT TITLE**

Fertilizer recommendation system for disease prediction

**TEAM ID:** PNT2022TMID45437

**MENTOR:**

T M LATHA **TEAM MEMBERS:**

1. A.Balakumar (Team Leader)
2. P.Ramkumar
3. E.Ranjith
4. P.K.Prakash

**TABLE OF CONTENTS**

**1. INTRODUCTION**

* 1. Project Overview
  2. Purpose

**2. LITERATURE SURVEY**

* 1. Existing problem
  2. References
  3. Problem Statement Definition

**3. IDEATION & PROPOSED SOLUTION**

* 1. Empathy Map Canvas
  2. Ideation & Brainstorming
  3. Proposed Solution
  4. Problem Solution fit

**4. REQUIREMENT ANALYSIS**

* 1. Functional requirement
  2. Non-Functional requirements

1. **PROJECT DESIGN** 
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
2. **PROJECT PLANNING & SCHEDULING** 
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
3. **CODING & SOLUTIONING (Explain the features added in the project along with code)** 
   1. Feature 1
   2. Feature 2
   3. Database Schema (if Applicable)
4. **TESTING** 
   1. Test Cases
   2. User Acceptance Testing
5. **RESULTS** 
   1. Performance Metrics
6. **ADVANTAGES & DISADVANTAGES**
7. **CONCLUSION**
8. **FUTURE SCOPE 13. APPENDIX** Source Code

GitHub & Project Demo Link

**1. INTRODUCTION**

**1.1 PROJECT OVERVIEW**

Fertilizer Recommendation system for disease

Prediction is asimple ML and DL based website which recommends the best crop to grow, fertilizers to use and the diseases caught by your crops. PROBLEM STATEMENT In India, the agriculture industry is extremely vital and crucial for economic and social development and jobs. In India, the agricultural sectorprovides a living for almost 48% of the population. As per the 2019-2020 economic survey, an Indian farmer's median wage in 16 states is Rupees 2500. Most of the Indian population depends on agriculture for their livelihood.

Agriculture gives an opportunity of employment to the village people to develop a country like India on large scale and give a push in the economic sector. The majority of farmers face the problem of planting an inappropriate crop for their land based on a conventional or non- scientific approach. This is achallenging task for a country like India, where agriculture feeds approximately42% of the population. And the outcomes for the farmer of choosing the wrongcrop for land is moving towards metro city for livelihoods, suicide, quitting the agriculture and give land on lease to industrialist or use for the non-agriculture purpose. The outcome of wrong crop selection is less yield and less profit

Modern Technology is enhancing and optimizing the performance of the Artificial Intelligences (AI) Model based crop yield disease prediction system and is helpful forfarmers to prevent the crop from various disease that can identify the disease within 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 thefarmer using the completely friendly and simple user interface using a web application to fetch the solution by own.

In addition to this, the process we planned is to add a validmodule which is fertilizer recommendation for a specific disease. It can give both artificial fertilizer and natural fertilizer in suggestion.

**1.2 PURPOSE**

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

1. **LITERATURE SURVEY**

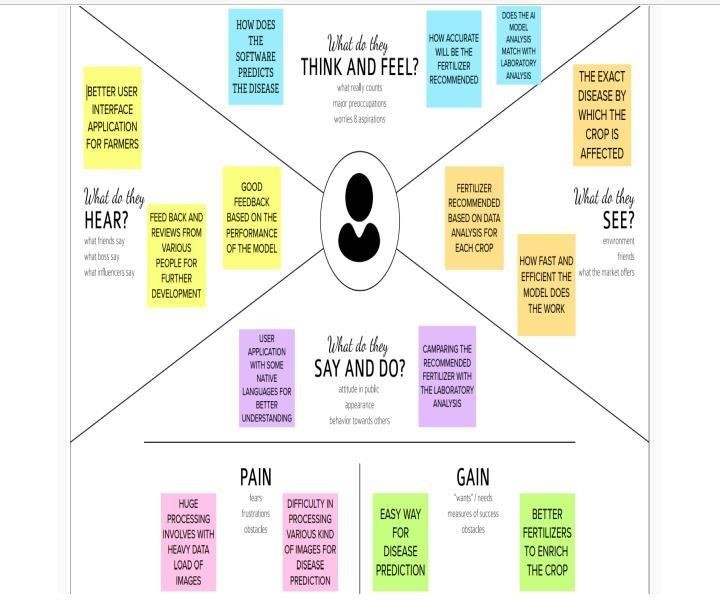
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.no** | **Name of researcher,**  **Year of**  **Publication** | **Paper title** | **Methodology**  **Adopted /**  **Modules Used** | **Observations Noted** |
| 1            2 | Prof. Rakesh  Shirsath, 2017          Ji-chun Zhao,  Jian-xin Guo,  2018 | Agriculture decision support system using  data mining          Big Data  Analysis  Technology | 1. Subscription based system 2. ANN  3. Android application 4. Personalized content      1.Inference engine | 1. Android app with a login module 2. Previously planted crops known to system 3. User feedback mechanism |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3              4              5 | Miftahul  Jannat  Mokarrama,  2017            S.Pudumalar,  E.Ramanujam,  2016            Yogesh  Gandge,  Sandhya,  2017 | Application in  Agricultural  Intelligence  Decision System            RSF: A  Recommendation  System for  Farmers              Crop  Recommendation  System for  Precision  Agriculture            A Study on  Various Data  Mining  Techniques for  Crop Yield  Prediction | 2.Domain expertise  3.Knowledge engineering 4.Knowledge  acquisition module  5.Knowledge base for recommendation  system    1.Location  Detection  2.Data analysis and storage 3.Similar location detection 4.  Recommendation generation module.         1. Random tree 2. CHAID 3. KNN 4. Naïve Bayes 5. WEKA tool         1. Attribute selection | 4.  Maintenance of crop.       1. Large database of crops 2. Processed using Hadoop 3. Professional   knowledge  4. Past experiences 5. Feature selection using  HDFS  6. Future  Scope: Using  Hadoop with  Artificial  Neural Networks.    1.  Physiographic, thermal, crop growing period, crop production rate  2. Seasonal crop database 2. Similar location detection |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 1. Multiple Linear   Regression   1. Decision Tree using ID3 4. SVM 2. Neural   Networks   1. C4.5 | 1. Generating the set of crops 2. Similarity between the crops planted in a region      1. Pre- processing of data 2. Handling missing and out-of-range values 3. Feature extraction   4. Ensemble model to get higher accuracy 5. Rule  generation     1. Selection of agricultural field 2. Selection of crop previously planted 3. Input from user 4. Preprocess 5. Attribute   Selection |

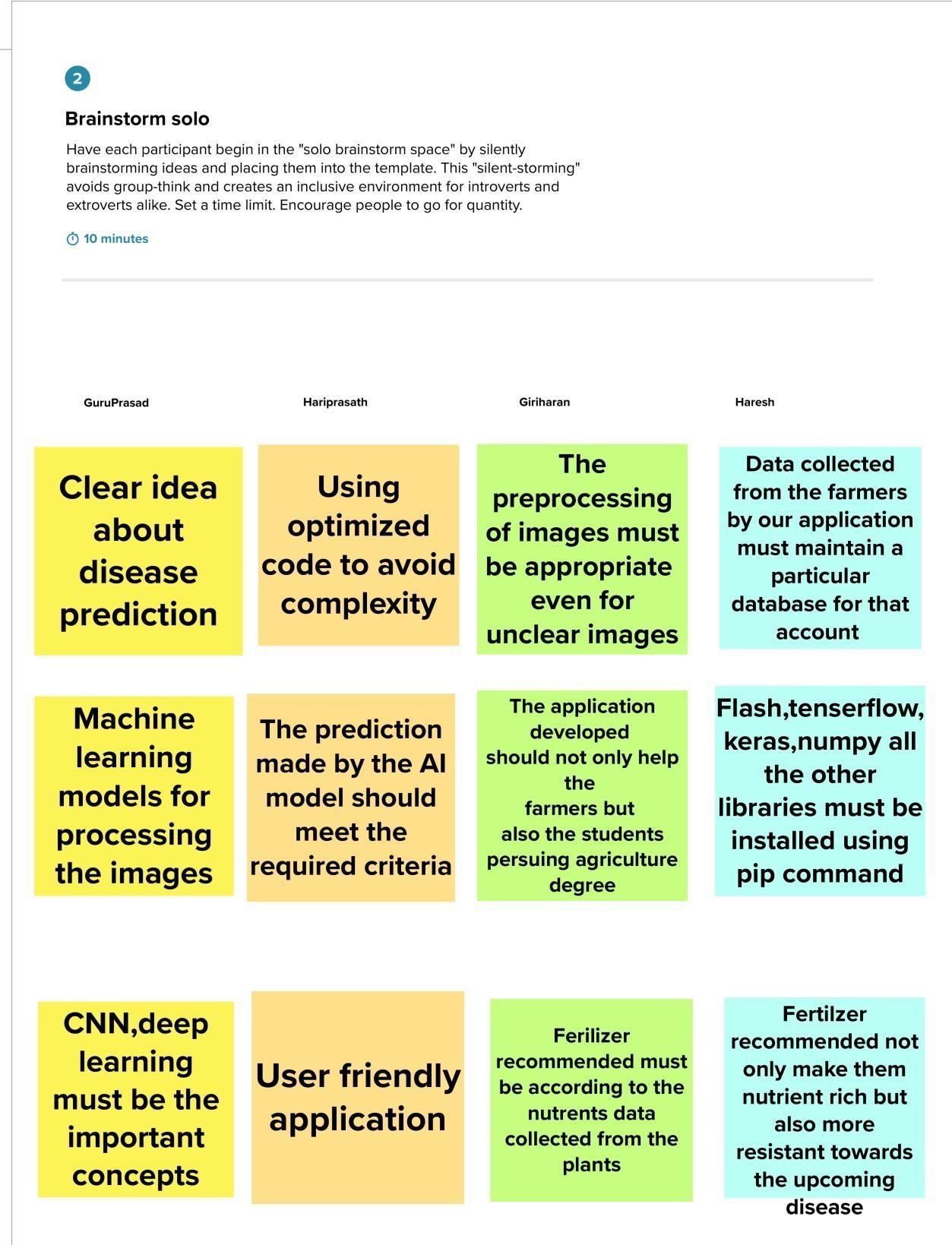
1. **IDEATION & PROPOSED SOLUTION** 
   1. **EMPATHY MAP**

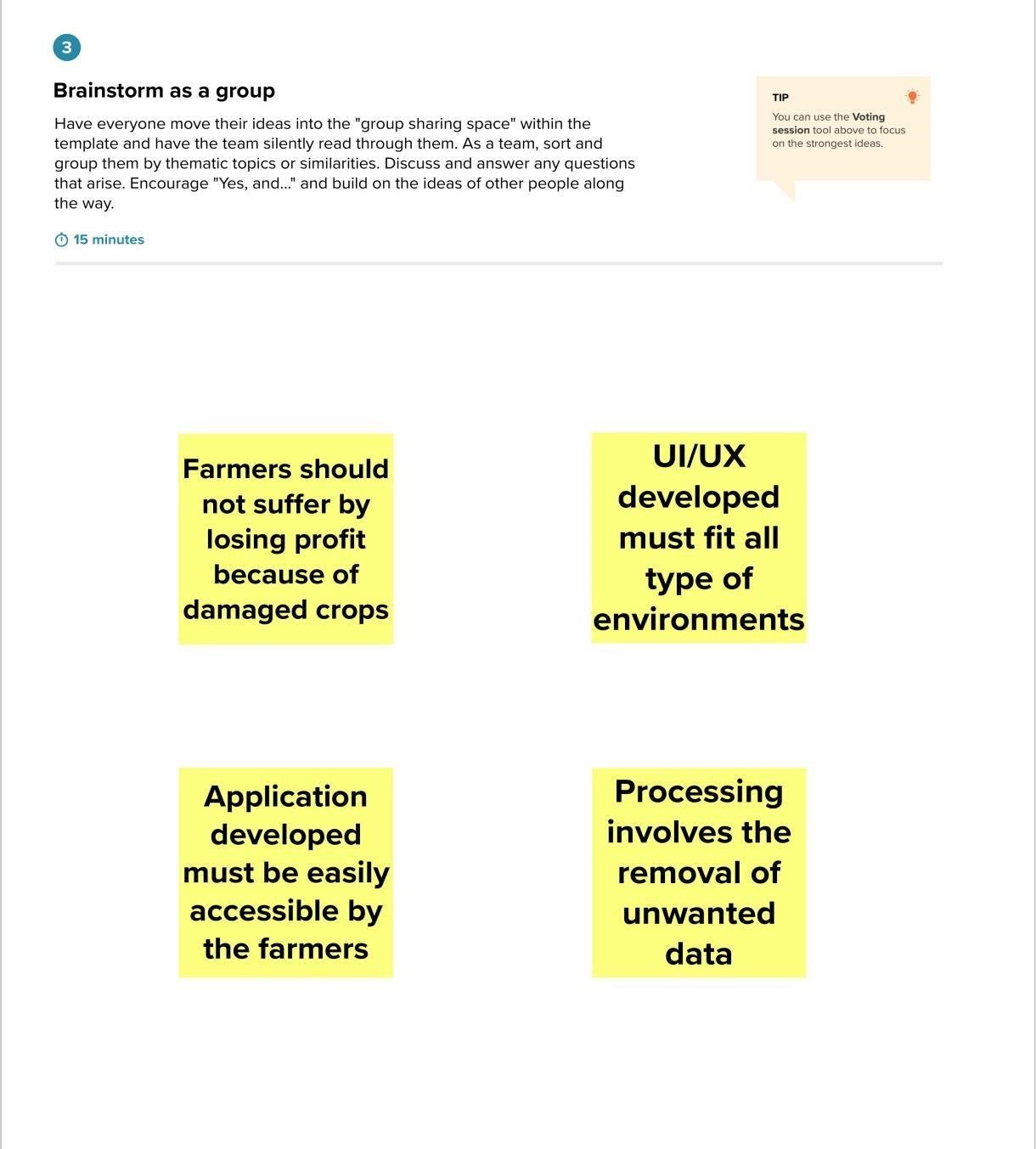
An empathy map is used to gain deeper insights on the customer’sinteraction with the system. It gives an idea on what the user feels and experiences while using the system, what fears the user has respective tothe system, etc. It also specifies how supportive the system environment is and what the users are likely to hear from the people around them regarding the usage of the system



* 1. **IDEATION AND BRAIN STORMING**

Ideation and Brainstorming are performed to generate ideas and solutions.Brainstorming is a group activity unlike ideation.





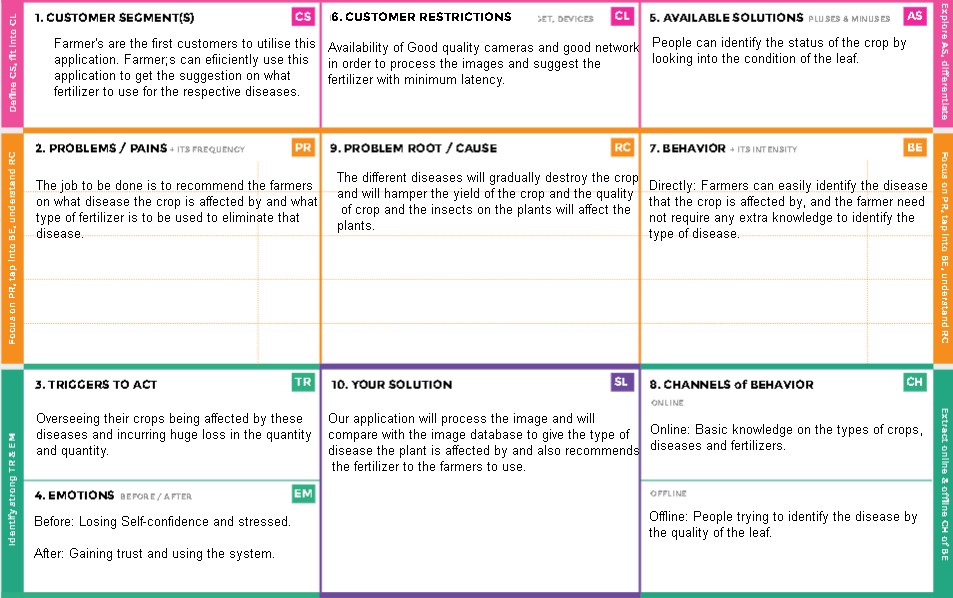
* 1. **PROPOSED SOLUTION**

The solution to the problem is Machine learning utilizing CNN, which is one of the applications of Artificial Intelligence, is being used to implement the proposed system. Crop recommendation is going to recommend you the best crop you can grow in your land as per the soil nutrition value and along with as per the climate in that region. And recommending the best fertilizer for every particular crop is also a challenging task.

One of the most important issues is when a plant gets caught by heterogeneous diseases that effect on less amount of agriculture production and compromises with quality as well. To overcome all these issues this recommendation has been proposed.

Nowadays a lot of research and work is being implemented in the smart and modern agriculture domain. Crop recommendation is characterized by a soil

* 1. **PROBLEM SOLUTION FIT**



1. **REQUIREMENT ANALYSIS**

* 1. **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 |
| FR-2 | User Confirmation | Confirmation via Email |
| FR-3 | User Profile | Filling the profile page after logging in |
| FR-4 | Uploading Dataset (Leaf) | Images of the leaves are to be uploaded |
| FR-5 | Requesting solution | Uploaded images is compared with the pre- defined Model and solution is generated |
| FR-6 | Downloading Solution | The Solution in pdf format which contains the recommendations of fertilizers and the possible diseases. |

***Non-functional Requirements:***

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

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | Description | The system allows the user to perform the tasks easily and efficiently and effectively. |
| NFR-2 | Security | Assuring all data inside the system or its part will be protected against malware attacks or unauthorized access. |

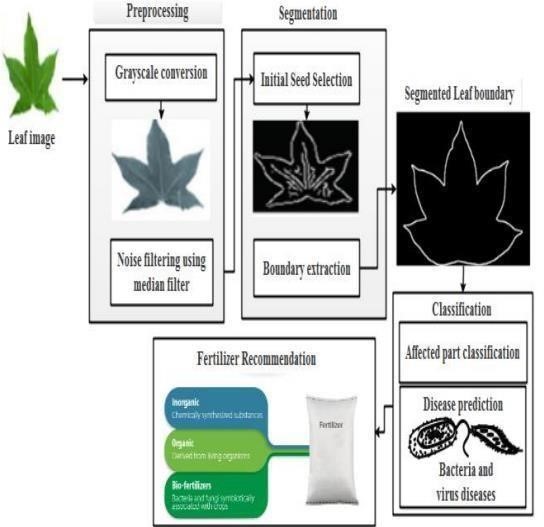
1. **PROJECT DESIGN**

* 1. **DATAFLOW DIAGRAM**

A data flow diagram or DFD(s) maps out the flow of information for any process or system. DFDs help you better understand process or system operation to discover potential problems, improve efficiency, and develop better processes.

**5. SOLUTION ARCHITECTURE**

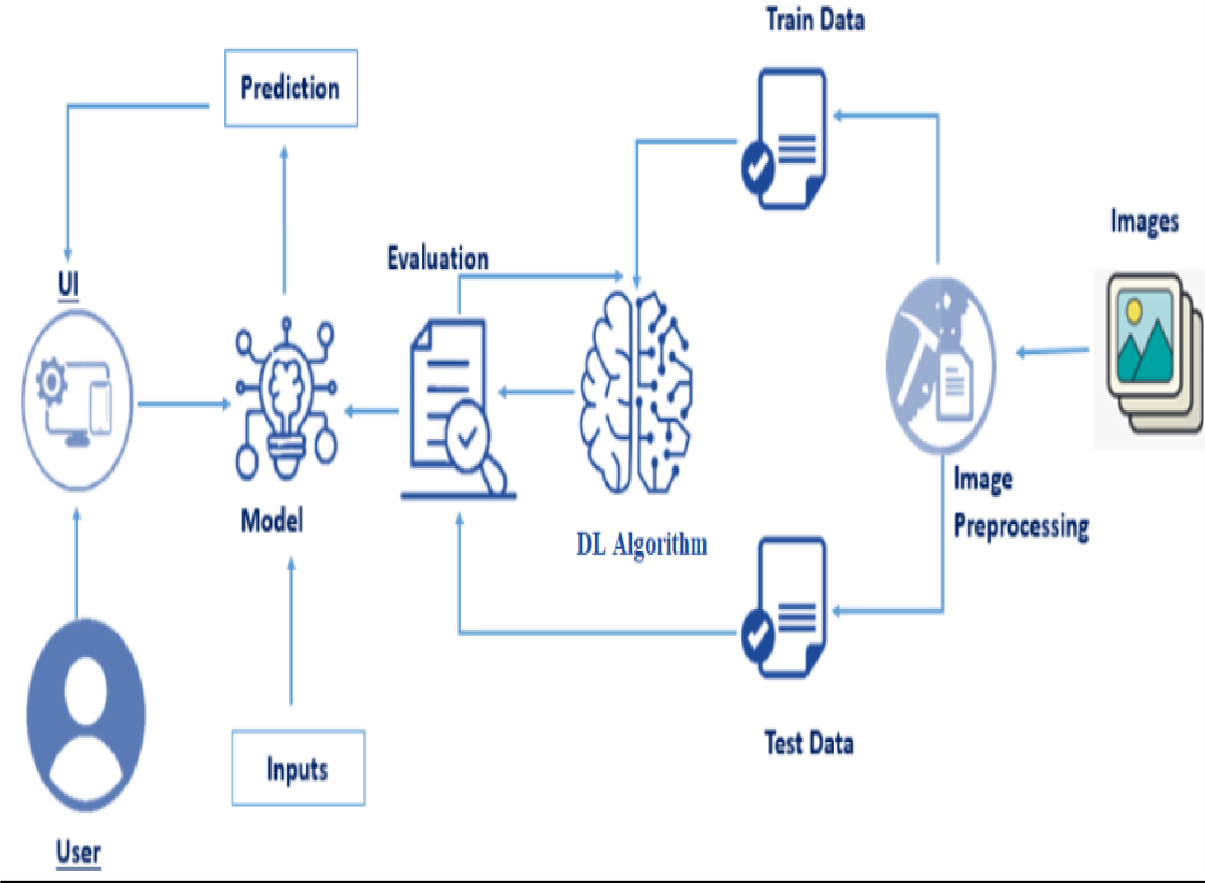
Solution architecture is the process of developing solutions based on predefined processes, guidelines and best practices with the objective that the developed solution fits within the enterprise architecture in terms of information architecture, system portfolios,integration requirements, etc



## 5.2 TECHNICAL ARCHITECTURE

Technical architecture involves the development of a technical blueprint regarding the arrangement, interaction, and

interdependence ofall elements so that system-relevant requirements are met.



**5.3 USER STORIES**

Our project was given out to several customers for experimentation and testing. They had given us satisfactory and valuable feedback in great detail on its range of use, accuracy, simplicity in handling, etc. It was tested and analysed on exhaustive number of leaf types and was proved to be effective in achieving what it was developed for – to predict the disease way before it becomes severe. The data garnered indicating various characteristics of the studied specimen using image processing and related techniques had helped them to develop a clear picture of its transformation when diseased.

It had a well-developed interface, although, complicated in its operation on the inside, very simple on the outside. Translation of all its comprehensive measurements to a inclusive output is highly needed and it surely does fulfil it.

**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 |
| Trader | Confirmation | 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 |
| Farmer | Signing up | USN-3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low | rJIRA-2 |
| Customer (Web user) | Registration | USN-4 | As a web user, I can register for the application through Gmail | I can register for the application through gmail | Medium | Sprint-1 |
| Administrator | Login | USN-5 | As a  administrator, I can log into the | I can log into the | High | Sprint-3 |
| **User Type** | **Functional**  **Requirement**  **(Epic)** | **User**  **Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
|  |  |  | application by entering email & password | application using mail & password |  |  |

**6. PROJECT PLANNING & SCHEDULING**

## 6.1 SPRINT PLANNING AND ESTIMATION

**Sprint1**

In this module we have done the dataset collection for vegetable and fruit disease prediction along with some image processing files in it

**Sprint2**

In this module we have developed the model for fruit and vegetable disease prediction with some CNN model and transfer learning and also we did the testing for both the models

**Sprint3**

In this module we have developed and designed our web application with front end as with HTML and backend with python scripts along with recommendation models trained with IBM

**Sprint4**

This is the final module for our project which contains the trained model for fruit and vegetable disease prediction

**6.3**

**6.2**

**SPRINT**

**DELIVERY**

**SCHEDULE**

The delivery plan of project deliverables is a strategic ele

ment for

every

project

manager.

Thegoal of every project is,

in fact,

to

produce

a

result

that serves

a

specific

purpose.

By

the word“purpose “,

we

mean

the most disparate goals: a software program, a chair, a building, a

translation

and

the

sort.

In Project Spirit Delivery, planning is one of the vital process of

completing the project and show casing thetime line of the project

planning. This delivery pl

an helps us to understand theprocess and work

flow

of

the

project

being

done

by

every

team

mate.

Every Single Module is assigned to the team mates to show case

their

work

and

contribution

indeveloping

the

project.

**REPORTS FROM JIRA**

**Backlog:**

A backlog is a list of issues that’s related to the project and the functions of the

system. It makes it simple to make, store, managea variety of problems

including

the

ones

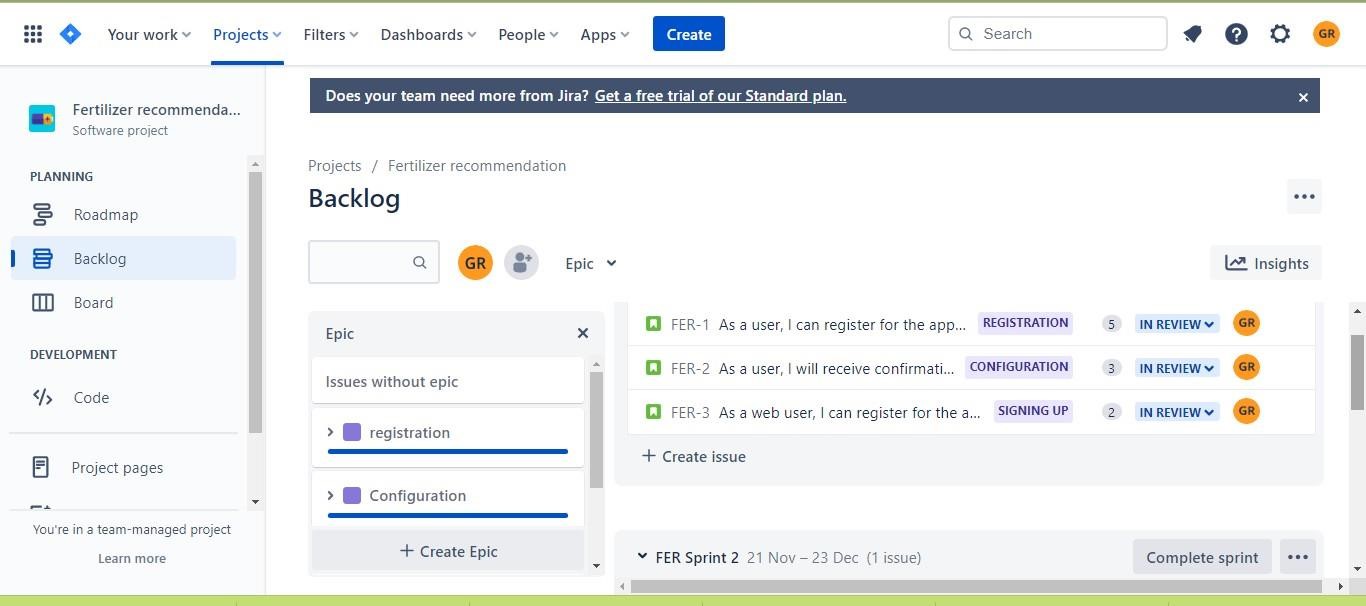
the

team

is

working

on.



Page

20

of

107

import

os

import

numpy

as

np

import

pandas

as

pd

from

tensorflow.keras.models

import

load\_model

#

from

tensorflow.keras.preprocessing

import

image

from

werkzeug.utils

import

secure\_filename

from

flask

import

Flask,

render\_template,

request

app

=

Flask(

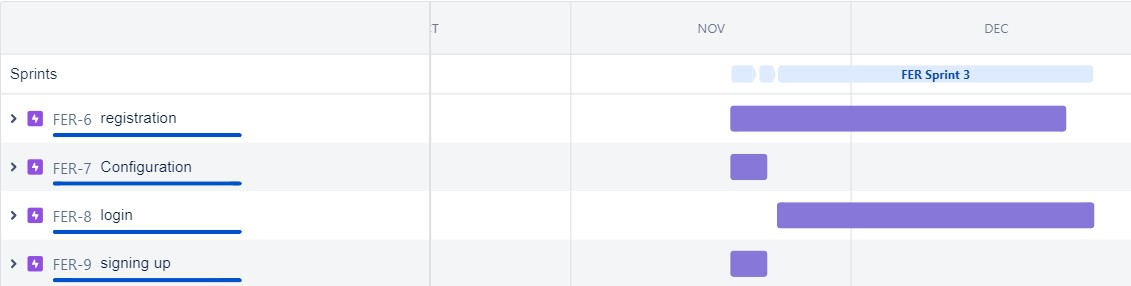
name

)

**Python**

**–**

**app.py:**



|  |
| --- |
| @app.route('/') def home(): return render\_template('home.html')  #prediction page  @app.route('/prediction') def prediction():  return render\_template('predict.html')  @app.route('/predict',methods=['POST']) def predict():  if request.method == 'POST':  # Get the file from post request f = request.files['image']  # Save the file to ./uploads basepath = os.path.dirname( file ) file\_path = os.path.join( basepath, 'uploads', secure\_filename(f.filename))  f.save(file\_path) img = image.load\_img(file\_path, target\_size=(128, 128))  x = image.img\_to\_array(img) x = np.expand\_dims(x, axis=0)  plant=request.form['plant'] print(plant) if(plant=="vegetable"): preds = model.predict(x) preds=np.argmax(preds) print(preds)  df=pd.read\_excel('precautions - veg.xlsx') print(df.iloc[preds]['caution'])  else:  preds = model1.predict(x) preds=np.argmax(preds)  df=pd.read\_excel('precautions - fruits.xlsx') print(df.iloc[preds]['caution'])  return df.iloc[preds]['caution']  if name == " main ":  app.run(debug=False)  s |

|  |
| --- |
| **Feature 1: home.html:**  <!DOCTYPE html>  <html >    <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title> Plant Disease Prediction</title>  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>  <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>  <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>  <link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">  <link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>  <link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>  <link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>  <script type="text/javascript" src="https://gc.kis.v2.scr.kaspersky- labs.com/FD126C42-EBFA-4E12-B309-  BB3FDD723AC1/main.js?attr=AMFGethlf4Q6r2IdpTrTqcDQGNLDU5Cbc3diYnUdLkg5mQrVB\_td22O  HUAsBJSd0oo8OR0zM3rIPeFWfnEY4XCxQu4KOxMSqlshEoIBOzvYw0SsMYpyUv4fnvKEjmJoj\_Y6cI4ov  -6AMOkz3Sh3epkfq0gltfnAPvvQBRdXqRmdqePVjlvvqL28ONZCiS0Qr5t0XGxJ0bSiWVT- rH3cqaKCk05eP1Dx04mieTcjsA\_TtFLx15PUu0ed6soaj-FOO6-  1d4OQxbJYBXUBefiUhzmOYCpsGIs1OyQvA0huo8AUYwYB72dvs07U3O2hq8BmYBv98h13sSo8iXKxyKx4 FUsOMkixjxYP6hu0wwi7yv1E2rei3GHtPl5YwHkWioQIPqvAmrlmaPtFZmF- jE4\_UUCi9IEKws8IduDiqQIFkxfO3YT\_sUC9gWmxKSpGbiebwCgV- wvdGEnbUxY18p9Db6jC6FVKRhqdMBianq63qv-  zZRMZbEpjzQT0DQAH3Yho4o4A00FIW2004q8Q80xt2kV928P\_nBgS9HOgHI5EZxenbjfqANTs1rh8GGhB d7RJaE8-  2AaqT6zbLf2tILJ8j4fk3bV1qsdw0fPmp6foJbDu4343XH36a0VGHsMLeVqcc30PSsE1pJbGE4\_C\_ExQd  0\_uRSA40mRjnFwHdLo9SJc1qghyc5YGQil\_utG48olMy9cC6z-iyKg1EeLKB43u- q4SlUimRnuUsZW7drNWaijSfJPDmkm7lUJ0POwQXPfnLa2\_spc3FisWCOZ7dFuIgDciIu0yF8rio2X0Pz 6pZkGQW4Fwl6vWKrLplmHagJElKXg58YSWwAT2DILilBjuSPiTwCHR9Ya\_mAXW4C03v7xzJlaSK9jneEC qctvKnH3RFgDS8ocfDcY65lXNRkq6v1hrcdv5sM2ek4Kjq4OFgX-wijr-0JdpSDpZlbIK00sPb4- u1B8c7MaCqBcbJAhfmg4utLU67fn5GLoCX\_-5TAWV0ID-\_sC1Vs9glWRPkKmmktJMbVy98XqC5-  DhtE3yd5I9ZM1SEH1gGYLlRjxwzPjWwHE-YH1Nx9lm-  Esq27TK7M86uT8iAe7LgtviO2YsCB0buShHWmjh3RzwMGqNqeymFSxPRK\_sDmTFoVjcaYpGa0kaMwhmmF  Page 27 of 107 |

|  |
| --- |
| 9AtPwGmFaGglv3rryVg0X0bGoXRetnrPpDG7jUoq5zQuXQSedBf9hmNwEqWsSZtI4zNTxjiEkxU0djhPX qByZbnelp\_3z6pqqniLzqj9jzAkvX6wDOW7ZycfDzOt- zNgTxWdtf41P6ZjVu8EWSf65Wqgen5jD4IPXgXGtxkjrSbrqiX-  NxxxfKVJUOoOcEO0F6n3DWD0BMWS8UGOQO8gZZeXCfpuTIGYTD6okyD91kLk5AmhaNTJVKjkHO- dHZqMHxikVhdK6C2PIfg4lEY0yuE3Fjj\_5NNX5ZalIpOl3LN6YQ8Jqis\_UmC\_OXmjW2F5Y4p8VRRKc1HW  2DFaUxBrEgfSwe\_keyaofodrjde\_pfPuDQDryEgGy9DNIhpGUV\_bQJ8jlPxRL7WSpmPU7- IZ1mVN\_onhqq2oI-WTl7ep-8w0GsJH3OhSRyyJC0XC9xtetqVjIHzcbKYFsxOaXT-  LLe7U9oHaXHzjDK3hn-ZNFYwzV\_aoq8180eb" charset="UTF-8"></script><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;  }  .topnav-right a:hover { background-color: #ddd; color: black;  }  .topnav-right a.active { |

|  |
| --- |
| background-color: #565961; color: white; }  .topnav-right { float: right; padding-right:100px;  } body {  background-color:#ffffff; background-repeat: no-repeat; background-size:cover; background-position: 0px 0px;  }  .button { background-color: #28272c; border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 16px; border-radius: 12px;  }  .button:hover { box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);  } form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}  input[type=text], input[type=password] { width: 100%; padding: 12px 20px; display: inline-block; margin-bottom:18px; border: 1px solid #ccc; box-sizing: border-box;  }  button { background-color: #28272c; color: white; padding: 14px 20px; |

|  |
| --- |
| margin-bottom:8px; border: none; cursor: pointer; width: 15%; border-radius:4px;  }  button:hover { opacity: 0.8;  }  .cancelbtn { width: auto; padding: 10px 18px; background-color: #f44336;  }  .imgcontainer { text-align: center; margin: 24px 0 12px 0;  }  img.avatar { width: 30%;  border-radius: 50%;  }  .container { padding: 16px;  }  span.psw { float: right; padding-top: 16px;  }  /\* Change styles for span and cancel button on extra small screens \*/  @media screen and (max-width: 300px) { span.psw { display: block; float: none;  }  .cancelbtn { width: 100%;  } |

|  |
| --- |
| }  .home{ margin:80px;  width: 84%; height: 500px; padding-top:10px; padding-left: 30px;  }  .login{ margin:80px; box-sizing: content-box; width: 84%; height: 420px; padding: 30px;  border: 10px solid blue;  }  .left,.right{ box-sizing: content-box; height: 400px; margin:20px; border: 10px solid blue;  }  .mySlides {display: none;} img {vertical-align: middle;}  /\* Slideshow container \*/  .slideshow-container { max-width: 1000px; position: relative; margin: auto;  }  /\* Caption text \*/  .text { color: #f2f2f2; font-size: 15px; padding: 8px 12px; position: absolute; bottom: 8px; width: 100%; text-align: center; |

|  |
| --- |
| }  /\* The dots/bullets/indicators \*/  .dot { height: 15px; width: 15px; margin: 0 2px; background-color: #bbb; border-radius: 50%; display: inline-block; transition: background-color 0.6s ease;  }  .active { background-color: #717171;  }  /\* Fading animation \*/  .fade {  -webkit-animation-name: fade; -webkit-animation-duration: 1.5s; animation-name: fade; animation-duration: 1.5s;  }  @-webkit-keyframes fade { from {opacity: .4} to {opacity: 1} }  @keyframes fade { from {opacity: .4} to {opacity: 1}  }  /\* On smaller screens, decrease text size \*/  @media only screen and (max-width: 300px) {  .text {font-size: 11px}  }  </style>  </head>  <body style="font-family:'Times New Roman', Times, serif;background- color:#C2C5A8;">  <div class="header"> |

|  |
| --- |
| <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Plant Disease Prediction</div>  <div class="topnav-right"style="padding-top:0.5%;">  <a class="active" href="{{ url\_for('home')}}">Home</a>  <a href="{{ url\_for('prediction')}}">Predict</a> </div>  </div>  <div style="background-color:#ffffff;">  <div style="width:60%;float:left;">  <div style="font-size:50px;font-family:Montserrat;padding-left:20px;text- align:center;padding-top:10%;">  <b>Detect if your plant<br> is infected!!</b></div><br>  <div style="font-size:20px;font-family:Montserrat;padding-left:70px;padding- right:30px;text-align:justify;">Agriculture is one of the major sectors worls wide. Over the years 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.</div><br><br>  </div>  </div>  <div style="width:40%;float:right;"><br><br>  <img src="{{url\_for('static',filename='images/12456.png')}}" style="max- height:100%;max-width:100%;">  </div>  </div>  <div class="home">  <br>  </div>  <script> var slideIndex = 0; showSlides();  function showSlides() { var i; var slides = document.getElementsByClassName("mySlides"); var dots = document.getElementsByClassName("dot"); |

|  |
| --- |
| for (i = 0; i < slides.length; i++) { slides[i].style.display = "none";    }    slideIndex++;  if (slideIndex > slides.length) {slideIndex = 1} for (i = 0; i < dots.length; i++) {  dots[i].className = dots[i].className.replace(" active", "");    } |

slides[slideIndex-1].style.display = "block";

**Feature 2 :**

**Predict.html:**

|  |
| --- |
| <!DOCTYPE html>  <html >  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <title> Plant Disease Prediction</title>  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>  <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>  <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>  <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">  <script type="text/javascript" src="https://gc.kis.v2.scr.kaspersky- labs.com/FD126C42-EBFA-4E12-B309-  BB3FDD723AC1/main.js?attr=3wvf44XdejigWHFj22ANQmgfA-L5oa67wZhZwPtEITSot6t8o- DPZwNcHRFhpa2tgGpDJGis4-1IHYyxyIAN2GE0-kSZKkCLRkbKttCLVN9mKhGFVtGJ3auoiiByn\_jJ- mA447x4TmdjGgz8XvMdLSPF4Gu5xwt0joGxWDXuOEF18Sa5usZGgj4TdDiTfDHpElX3P1eH-  lsevFhUJQEZe3981VXjRKYRn2FrxsYwXGSMBn0sRR9IYup35XYNQkvA6DLQV1lwLc4XuAo0BlJYAfI75R 4O5LwTWuT-uaft0DEQeuV\_f3rKvkrcBkalcpWnyXVLeLyjMz5CqpZ1aSCy1MgVAzWxGb-  GX3eQb0F5qOksANddV\_vhz1Ai4RgptuAfB8mVyuz0nWZzpmwam34lc4NL4tfyWGncKz2taMyGfsK4Mrn0 zfPlY9\_n9FP0lMlAX0IQ8TfbVp4B1vbwnA- |

|  |
| --- |
| RVJq8mxoTjgMgqhKhp6NQY\_8gZULkbqqA0pqUMvfL3\_fZC1PFipLNjCyCGe9YOaU9L7QF4CXeKsRhJXmI  898FhpxB1oI7z0xvndsDLPRsqbNuse\_eGL9tz0Te5HLGhtoXSn5O8pHC99\_XHYofrlismcByzZlmVqVkC NfmbnMjaD9IQf6xAACyjkQ927AOvyDVCZKr- tV6wRZyv\_z7Z1J9AG7SGSLoB34AkMytkYXvpgGn21pGFNhvl3YSmyKYc2XJs89zHbp5fSyXsfasogSEYL bpxCmuvzZKO4haaqouKDcLwBGMFp\_Br095f-  AlhhWOdPDx1ezvTMx1NgS4QO97OmbyQCqHUFWWZLYNgjQ8zpfdBXB17L\_v\_lfmrUWhUiUVc9tRcJy- lpchFJe8Gz7TUOKCRDjbIWtiqXryDeENrJgQ31laXp-  VVYpOI1L55pek2fgk5OCGNzVges5oG4PpMyCIXtJpv32E5rlPTktG4hD8eXmYQECVU1HvSmEiKvuY6T6i 9wdpqg\_AnycRzUXmYdahFT3W7zToIn2RXzNfdOU0zbYBvtJ70TpR4PjfU75lJ0FsnphDuCnero3UYOak7 vYvGYD9YV2md5v-3AmP-eOor2m55JZRH\_Hxpn28x-nDNCOHqVBC6leYuYFBVV\_vL5l- E8n92uWUqwMEzdZPZtAyRaCfz3D2Y0IYn-  ZrnfNTg2M\_zVJePmUu1xdjYh7d1dx7nwclm7wJrBPb3JnX2kvEGYs9SM17MlwzoY1VJq4UzJ2D6oEvhQw  HvG4e1etlS6iLWzhy8RVMfBlTa4DPDOHmTlHhsKbn0UaMyFFCppe79rtIVRctcomnVmQysUwUOhjzlAq3 0-hXJCTqdCWJe2xnxjAuUHVqHSiHiZllZaoOWNCV5Ypx\_eqzn-KyZS3u-  2\_hGLHHNA2AVBWn\_hF3Gz16dw6zA4QSmWZSfDUcNObLJGOSTaDS3Z8jPTloYPFmu8oES6TL1dLlEK5Yhc SGaX4iv6o95drsZGb6bBcWgT7sNFHW6dVE9wdjoDFuBergPIAm0sKaZQ2Ex6j15OWCbE6UaPg- VNfziA2FEPpJaI9hEPI2gdaSuHqovlEOt5mjuFBBOxpK0t8kOZRtsVzqUuJw3VcLjaP6SfG\_KZfgX\_g8T  Ps6CcFhlLRz63oXMQFPW6AA7eudWfygndazedq5B-  6DqSkOT04GTUJNqLcElg6KEEWqxd88BzoQoK28jrAf-xWHNIZv5HmQQYEnyX0U\_cW8HX- hde54TuY\_fY3e5QYu4be-JxTkA4JxWLEagSa7-zs" charset="UTF-8"></script><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;  }  .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="{{url\_for('static',filename='images/789.jpg')}}"  style="height:450px;width:550px"class="img-rounded" alt="Gesture">  </div>  <div class="col-sm-6">  <div>  <h4>Drop in the image to get the prediction </h4> <form action = "" id="upload-file" method="post"  enctype="multipart/form-data">  <select name="plant">  <option value="select" selected>Select plant type</option>  <option value="fruit">Fruit</option>  <option value="vegetable">Vegetable</option>  </select><br>  <label for="imageUpload" class="upload-label" style="background:  #28272c;">  Choose...  </label>  <input type="file" name="image" id="imageUpload" accept=".png,  .jpg, .jpeg">  </form>  <div class="image-section" style="display:none;">  <div class="img-preview">  <div id="imagePreview">  </div>  </div>  <div>  <button type="button" class="btn btn-info btn-lg " id="btn-  predict" style="background: #28272c;">Predict!</button> </div>  </div>  <div class="loader" style="display:none;"></div>  <h3>  <span id="result" style="font-size:17px; "> </span> </h3> |

|  |  |
| --- | --- |
| </div> | Page 39 of 107 |

|  |
| --- |
| </div>        </div>      </div>      </div>      </div>    **final.css:**    .img-preview { width: 256px; height: 256px;  position: relative; border: 5px solid #F8F8F8;  box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1); margin-top: 1em;  margin-bottom: 1em;      }        .img-preview>div { width: 100%;  height: 100%;  background-size: 256px 256px; background-repeat: no-repeat; background-position: center;    }      input[type="file"] {  Page 40 of 107 |

|  |
| --- |
| transition: all .4s; cursor: pointer;    }      .upload-label:hover{ background: #C2C5A8; color: #39D2B4;    }      .loader {    border: 8px solid #f3f3f3; /\* Light grey \*/ border-top: 8px solid #28272c; /\* Blue \*/ border-radius: 50%;  width: 50px; height: 50px; |

**main.js:**

|  |
| --- |
| $(document).ready(function () {      // Init      $('.image-section').hide();      $('.loader').hide();      $('#result').hide();        // Upload Preview function readURL(input) {  if (input.files && input.files[0]) { var reader = new FileReader(); |

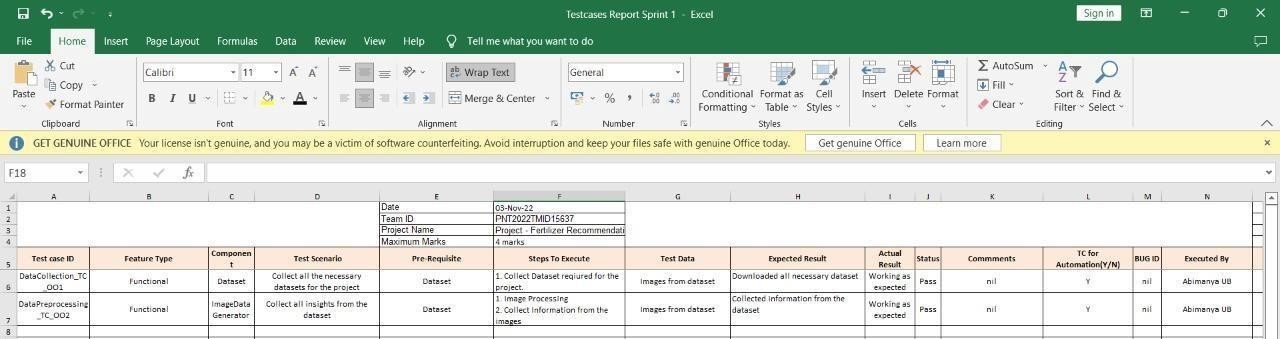
41

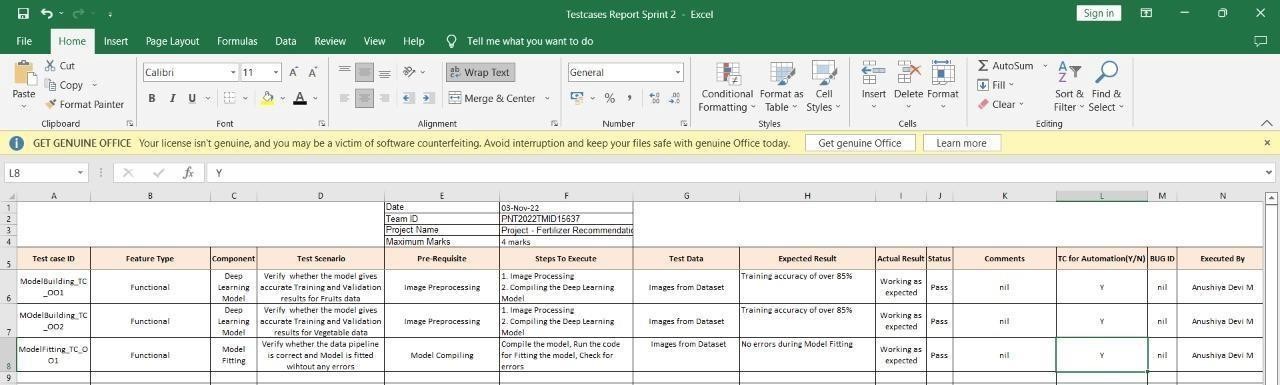
|  |
| --- |
| $("#imageUpload").change(function () {  $('.image-section').show();  $('#btn-predict').show();  $('#result').text(''); $('#result').hide(); readURL(this);  });    // Predict  $('#btn-predict').click(function () { var form\_data = new FormData($('#upload-file')[0]);  // Show loading animation  $(this).hide();  $('.loader').show();    // Make prediction by calling api /predict  $.ajax({ type: 'POST', url: '/predict', data: form\_data, contentType: false, cache: false, processData: false, async: true, success: function (data) {  // Get and display the result  $('.loader').hide();  $('#result').fadeIn(600);  $('#result').text('Prediction: '+data); console.log('Success!');  },  });  });      });  Page 42 of 107 |

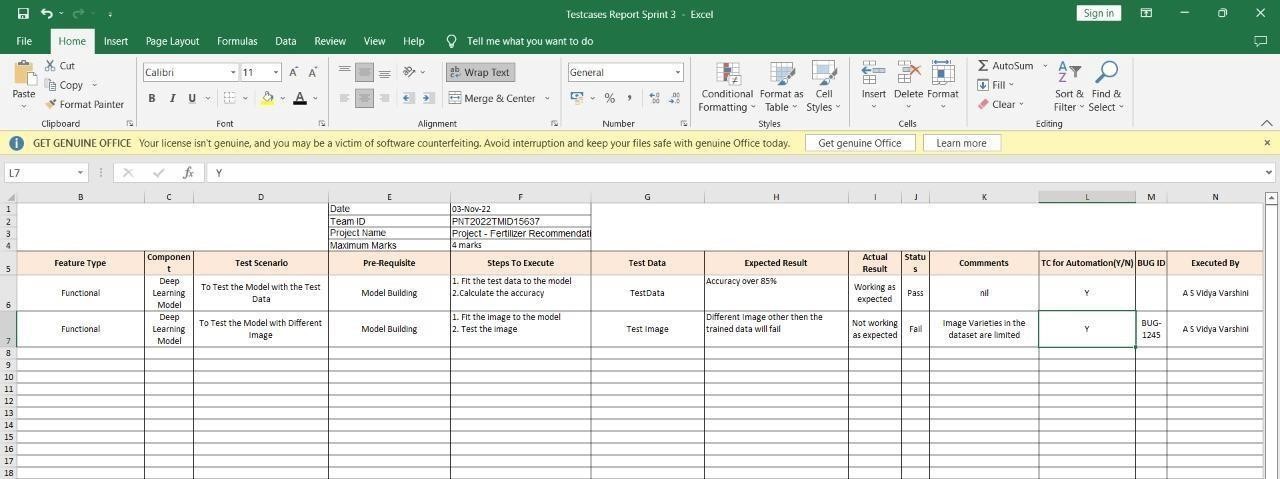
# TESTING

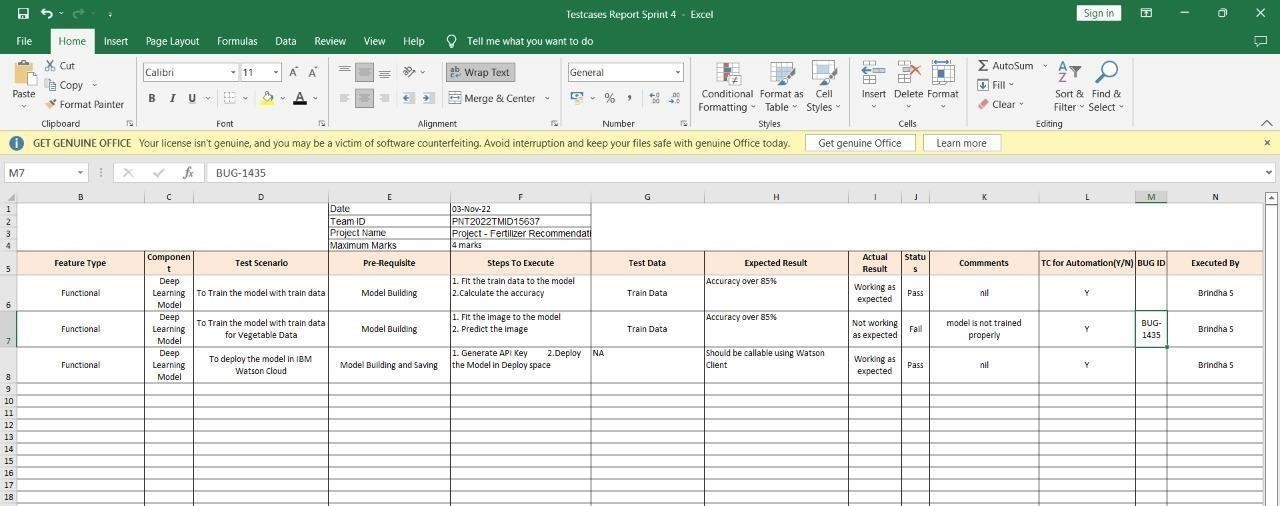
## Test Cases

Test cases are a set of actions performed on a system to determine if it satisfies software requirements and functionscorrectly as it claimed to perform.





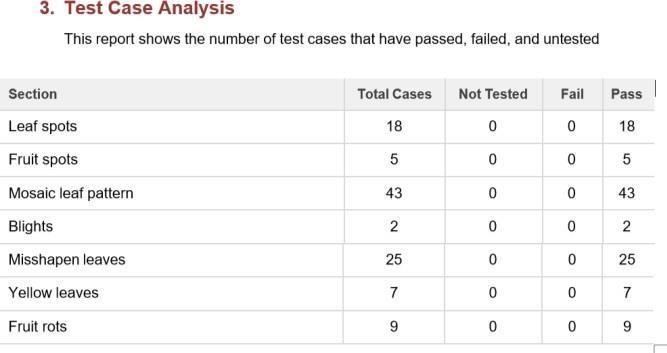
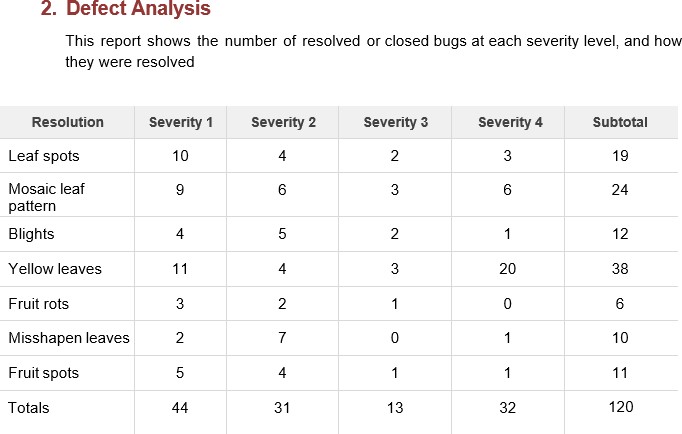
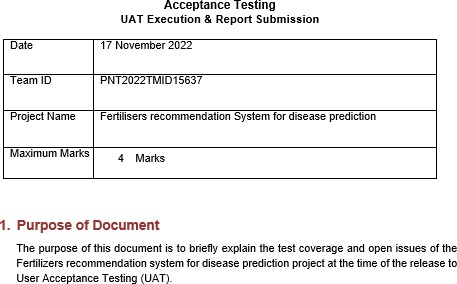




# User Acceptance Testing

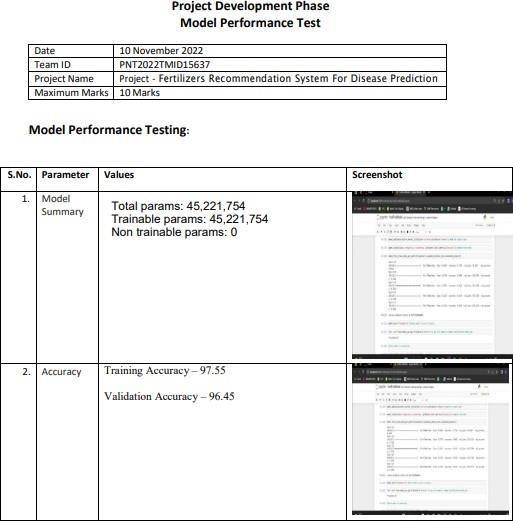
Before deploying the software application to a production environment

the end user or client performs a type of testing known as user acceptance testing, or UAT to ensure whether the software functionalities serve the purpose of development.



# RESULTS Performance Metrics

metrics are a baseline for performance tests. Monitoring the correct parameters will help you detect areas that require increased attention and find ways to improve them.



# ADVANTAGES & DISADVANTAGES

**Advantages:**

* Early detection of plant diseases.
* Proper fertilizer recommendation to prevent or cure the plant infection or disease.
* No need to consult any specialists. • Fully automated system.

**Disadvantages:**

* Requires training the system with large dataset.
* Works only on the pretrained diseases.
* When a plant is infected with multiple diseases the system may not predict all the diseases due to the mixed symptoms.
* Requires a good device connected to the internet.

# CONCLUSION

Hence a system that takes in images as user input, analyses those for certain symptoms and identifies the disease, recommends the fertilizer to counter the deficiency of the nutrients is built and deployed.

# FUTURE SCOPE

The system must be trained with numerous images of plant disease symptoms. In case of presence of multiple diseases, suitable classification must be done to predict each disease accurately and recommend separate fertilizers as a solution to each deficiency or infection.

# APPENDIX Source Code

**Home.html:**

<!DOCTYPE html>

<html >

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1"> <title> Plant Disease Prediction</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link

href='https://fonts.googleapis.com/css?family=Open+Sans+Condens ed:300' rel='stylesheet'type='text/css'>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

<script type="text/javascript" src="https://gc.kis.v2.scr.kaspersky- labs.com/FD126C42-EBFA-4E12-B309-

BB3FDD723AC1/main.js?attr=AMFGethlf4Q6r2IdpTrTqcDQGNLDU5

Cbc3diYnUdLkg5mQrVB\_td

22OHUAsBJSd0oo8OR0zM3rIPeFWfnEY4XCxQu4KOxMSqlshEoIBOzvY w0SsMYpyUv4fnvKEjm Joj\_Y6cI4ov-

6AMOkz3Sh3epkfq0gltfnAPvvQBRdXqRmdqePVjlvvqL28ONZCiS0Qr

5t0XGxJ0bSiWVT- rH3cqaKCk05eP1Dx04mieTcjsA\_TtFLx15PUu0ed6soaj-FOO6- 1d4OQxbJYBXUBefiUhzmOYCpsGIs1OyQvA0huo8AUYwYB72dvs07U3 O2hq8BmYBv98h13sSo8

iXKxyKx4FUsOMkixjxYP6hu0wwi7yv1E2rei3GHtPl5YwHkWioQIPqv AmrlmaPtFZmF- jE4\_UUCi9IEKws8IduDiqQIFkxfO3YT\_sUC9gWmxKSpGbiebwCgV- wvdGEnbUxY18p9Db6jC6FVKRhqdMBianq63qv-

zZRMZbEpjzQT0DQAH3Yho4o4A00FIW2004q8Q80xt2kV928P\_nBg

S9HOgHI5EZxenbjfqANTs1rh8GGhBd7RJaE8-

2AaqT6zbLf2tILJ8j4fk3bV1qsdw0fPmp6foJbDu4343XH36a0VGHsML eVqcc30PSsE1pJbGE4\_C\_E

xQd0\_uRSA40mRjnFwHdLo9SJc1qghyc5YGQil\_utG48olMy9cC6z- iyKg1EeLKB43u-

q4SlUimRnuUsZW7drNWaijSfJPDmkm7lUJ0POwQXPfnLa2\_spc3FisWC

OZ7dFuIgDciIu0yF8rio2X

0Pz6pZkGQW4Fwl6vWKrLplmHagJElKXg58YSWwAT2DILilBjuSPiTwC

HR9Ya\_mAXW4C03v7x

zJlaSK9jneECqctvKnH3RFgDS8ocfDcY65lXNRkq6v1hrcdv5sM2ek4K

jq4OFgX-wijr- 0JdpSDpZlbIK00sPb4-

u1B8c7MaCqBcbJAhfmg4utLU67fn5GLoCX\_-5TAWV0ID- \_sC1Vs9glWRPkKmmktJMbVy98XqC5-

DhtE3yd5I9ZM1SEH1gGYLlRjxwzPjWwHE-YH1Nx9lm- Esq27TK7M86uT8iAe7LgtviO2YsCB0buShHWmjh3RzwMGqNqeymF SxPRK\_sDmTFoVjcaYpGa0

kaMwhmmF9AtPwGmFaGglv3rryVg0X0bGoXRetnrPpDG7jUoq5zQu

XQSedBf9hmNwEqWsSZtI4z

NTxjiEkxU0djhPXqByZbnelp\_3z6pqqniLzqj9jzAkvX6wDOW7ZycfDz

Ot- zNgTxWdtf41P6ZjVu8EWSf65Wqgen5jD4IPXgXGtxkjrSbrqiX- NxxxfKVJUOoOcEO0F6n3DWD0BMWS8UGOQO8gZZeXCfpuTIGYTD6 okyD91kLk5AmhaNTJV KjkHO-

dHZqMHxikVhdK6C2PIfg4lEY0yuE3Fjj\_5NNX5ZalIpOl3LN6YQ8Jqis\_U mC\_OXmjW2F5Y4p8VR

RKc1HW2DFaUxBrEgfSwe\_keyaofodrjde\_pfPuDQDryEgGy9DNIhpG

UV\_bQJ8jlPxRL7WSpmPU7

-IZ1mVN\_onhqq2oI-WTl7ep-

8w0GsJH3OhSRyyJC0XC9xtetqVjIHzcbKYFsxOaXT- LLe7U9oHaXHzjDK3hn-ZNFYwzV\_aoq8180eb" charset="UTF-8"></script><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; }

|  |  |
| --- | --- |
| .topnav-right a:hover { background- color: #ddd; color: black;  }    .topnav-right a.active { background-color:  #565961;color: white;  }    .topnav- right { float: right; padding-right:100px;  }  body {    background- color:#ffffff; background-repeat: no-repeat; background- size:cover; background-position:  0px 0px; }  .button { background-color:  #28272c;border:  none; color: white; padding: 15px 32px;text- align: center; | Page 51 of 107 |

text-decoration: none;display: inline-block; font-size: 16px; border- radius: 12px;

}

.button:hover { box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);

}

form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text],

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

button { background-color:

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

button:ho ver { opacity:

0.8;}

.cancelbt n { width: auto; padding: 10px 18px; background-color:

#f44336;}

.imgcontaine r { text- align: center;

margin: 24px 0 12px 0;}

img.avat ar { width: 30%; border-radius: 50%;}

.container { padding:

16px;}

span.ps w { float: right;

padding-top: 16px;}

/\* Change styles for span and cancel button on extra small screens \*/@media screen and (max- width: 300px) { span.psw {

display: block; float: none;}

.cancelbtn { width:

100%;}}

.home{ margin:80

px; width: 84%; height: 500px; padding- top:10px; padding-left: 30px;}

.login{ margin:80px; box-sizing: content-box;width: 84%;

height: 420px; padding:

30px; border: 10px solid blue;

}

.left,.right{ box-sizing: content-box; height: 400px; margin:20px; border: 10px solid blue;

}

.mySlides {display: none;} img {vertical-align: middle;}

/\* Slideshow container \*/

.slideshow- container { max-width: 1000px; position: relative; margin: auto;

}

/\* Caption text \*/

.text { color: #f2f2f2; font-size: 15px; padding: 8px 12px; position: absolute; bottom: 8px; width: 100%; text-align: center;

}

/\* The dots/bullets/indicators \*/

.dot {

height: 15px; width: 15px; margin: 0 2px; background- color: #bbb; border-radius:

50%; display: inline-block;

transition: background-color 0.6s ease;

}

.active { background-color: #717171;

}

/\* Fading animation \*/

.fade {

-webkit-animation-name: fade;

-webkit-animation- duration: 1.5s; animation-name: fade; animation-duration: 1.5s;

}

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

}

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

}

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

}

</style>

</head>

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

<div class="header">

<div style="width:50%;float:left;font-size:2vw;text- align:left;color:white; padding-top:1%">PlantDisease Prediction</div>

<div class="topnav-right"style="padding-top:0.5%;">

<a class="active" href="{{ url\_for('home')}}">Home</a>

<a href="{{ url\_for('prediction')}}">Predict</a> </div>

</div>

<div style="background-color:#ffffff;">

<div style="width:60%;float:left;">

<div style="font-size:50px;font-family:Montserrat;padding- left:20px;text-align:center;padding-top:10%;">

<b>Detect if your plant<br> is infected!!</b></div><br> <div style="font-size:20px;font-family:Montserrat;padding- left:70px;padding-right:30px;text- align:justify;">Agriculture is one of the major sectors worls wide. Over the years it has developed andthe use of new technologies and equipment replaced almost all the traditional methods of farming. Theplant 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.</div><br><br> </div>

</div>

<div style="width:40%;float:right;"><br><br>

<img src="{{url\_for('static',filename='images/12456.png')}}"

style="max-height:100%;max-width:100%;">

</div>

</div>

<div class="home">

<br>

</div>

<script> var slideIndex =

0; showSlides()

;

function showSlides() { var i;

var slides =

document.getElementsByClassName("mySli des");var dots =

document.getElementsByClassName("dot"); for (i = 0; i <

slides.length; i++) { slides[i].style.display

= "none"; } slideIndex++; if (slideIndex > slides.length) {slideIndex = 1}for (i = 0; i < dots.length; i++) {

dots[i].className = dots[i].className.replace(" active", "");

}

slides[slideIndex- 1].style.display = "block"; dots[slideIndex-1].className

+= " active";

setTimeout(showSlides, 2000); // Change image every 2 seconds

}

</script>

</body>

</html>

**Predict.html:**

<!DOCTYPE html>

<html >

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title> Plant Disease Prediction</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">

<script type="text/javascript" src="https://gc.kis.v2.scr.kaspersky- labs.com/FD126C42-EBFA- 4E12-B309-

BB3FDD723AC1/main.js?attr=3wvf44XdejigWHFj22ANQmgfA-

L5oa67wZhZwPtEITSot6t8o-DPZwNcHRFhpa2tgGpDJGis4-

1IHYyxyIAN2GE0-

kSZKkCLRkbKttCLVN9mKhGFVtGJ3auoiiByn\_jJ-

mA447x4TmdjGgz8XvMdLSPF4Gu5xwt0joGxWDXuOEF18Sa5usZGg j4TdDiTfDHpElX3P1eH-

lsevFhUJQEZe3981VXjRKYRn2FrxsYwXGSMBn0sRR9IYup35XYNQkv

A6DLQV1lwLc4XuAo0B lJYAfI75R4O5LwTWuT- uaft0DEQeuV\_f3rKvkrcBkalcpWnyXVLeLyjMz5CqpZ1aSCy1MgVAz

WxGb-

GX3eQb0F5qOksANddV\_vhz1Ai4RgptuAfB8mVyuz0nWZzpmwam34l c4NL4tfyWGncKz2taMyGfs

K4Mrn0zfPlY9\_n9FP0lMlAX0IQ8TfbVp4B1vbwnA-

RVJq8mxoTjgMgqhKhp6NQY\_8gZULkbqqA0pqUMvfL3\_fZC1PFipLNjC yCGe9YOaU9L7QF4CXe

KsRhJXmI898FhpxB1oI7z0xvndsDLPRsqbNuse\_eGL9tz0Te5HLGhtoX

Sn5O8pHC99\_XHYofrlismc

ByzZlmVqVkCNfmbnMjaD9IQf6xAACyjkQ927AOvyDVCZKr- tV6wRZyv\_z7Z1J9AG7SGSLoB34AkMytkYXvpgGn21pGFNhvl3YSm yKYc2XJs89zHbp5fSyXsfas

ogSEYLbpxCmuvzZKO4haaqouKDcLwBGMFp\_Br095f- AlhhWOdPDx1ezvTMx1NgS4QO97OmbyQCqHUFWWZLYNgjQ8zpf

dBXB17L\_v\_lfmrUWhUiUVc9tRcJy-

lpchFJe8Gz7TUOKCRDjbIWtiqXryDeENrJgQ31laXp-

VVYpOI1L55pek2fgk5OCGNzVges5oG4PpMyCIXtJpv32E5rlPTktG4hD8e XmYQECVU1HvSmEiK

vuY6T6i9wdpqg\_AnycRzUXmYdahFT3W7zToIn2RXzNfdOU0zbYBvt J70TpR4PjfU75lJ0FsnphDu Cnero3UYOak7vYvGYD9YV2md5v- 3AmP-eOor2m55JZRH\_Hxpn28x- nDNCOHqVBC6leYuYFBVV\_vL5l-

E8n92uWUqwMEzdZPZtAyRaCfz3D2Y0IYn-

ZrnfNTg2M\_zVJePmUu1xdjYh7d1dx7nwclm7wJrBPb3JnX2kvEGYs9SM1

7MlwzoY1VJq4UzJ2D6o

EvhQwHvG4e1etlS6iLWzhy8RVMfBlTa4DPDOHmTlHhsKbn0UaMyF FCppe79rtIVRctcomnVmQy sUwUOhjzlAq30-

hXJCTqdCWJe2xnxjAuUHVqHSiHiZllZaoOWNCV5Ypx\_eqzn- KyZS3u-

2\_hGLHHNA2AVBWn\_hF3Gz16dw6zA4QSmWZSfDUcNObLJGOST aDS3Z8jPTloYPFmu8oES6T

L1dLlEK5YhcSGaX4iv6o95drsZGb6bBcWgT7sNFHW6dVE9wdjoDFuBer gPIAm0sKaZQ2Ex6j15O WCbE6UaPg-

VNfziA2FEPpJaI9hEPI2gdaSuHqovlEOt5mjuFBBOxpK0t8kOZRtsVzq

UuJw3VcLjaP6SfG\_KZfgX\_ g8TPs6CcFhlLRz63oXMQFPW6AA7eudWfygndazedq5B- 6DqSkOT04GTUJNqLcElg6KEEWqxd88BzoQoK28jrAf- xWHNIZv5HmQQYEnyX0U\_cW8HX- hde54TuY\_fY3e5QYu4be- JxTkA4JxWLEagSa7-zs" charset="UTF-8"></script><script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></scri pt>

<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+Condens ed: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;  }    .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; | Page 62 of 107 |

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%">PlantDisease 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="{{url\_for('static',filename='images/789.jpg')}}"

style="height:450px;width:550px"class="img-rounded" alt="Gesture">

</div>

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

<div>

<h4>Drop in the image to get the prediction </h4> <form action = "" id="upload-file" method="post"

data"> enctype="multipart/form-

<select name="plant">

<option value="select" selected>Select plant type</option>

<option value="fruit">Fruit</option>

</select><b <option value="vegetable">Vegetable</option>

r>

#28272c <label for="imageUpload" class="upload-label"

;"> style="background: Choose...

</label>

<accept=".png,input type="file" name="image" id="imageUpload"

.jpg, </for

.jpeg"> m>

<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>

</div> </h3>

</div>

</div>

</div>

</div>

</div>

</body>

<footer>

<script src="{{ url\_for('static', filename='js/main.js') }}"

type="text/javascript"></script>

</footer>

</html>

**main.js:**

$(document).ready(function () {

// Init

$('.image-section').hide();

$('.loader').hide();

$('#result').hide();

// Upload Preview function readURL(input) {

if (input.files && input.files[0]) { var

reader = new

FileReader(); reader.onload = function (e) {

$('#imagePreview').css('background-image', 'url(' + e.target.result + ')');

$('#imagePreview').hide();

$('#imagePreview').fadeIn(650);

}

reader.readAsDataURL(input.files[0]);

}

}

$("#imageUpload").change(function () {

$('.image-section').show();

$('#btn-predict').show();

$('#result').text('');

$('#result').hide(); readURL(this);

});

// Predict

$('#btn-predict').click(function () { var form\_data = new FormData($('#upload-file')[0]);

// Show loading animation

$(this).hide();

$('.loader').show();

// Make prediction by calling api /predict

$.ajax({ type: 'POST', url: '/predict', data: form\_data, contentType: false,cache:

false, processData: false,async:

true,

success: function (data) { // Get and display the result $('.loader').hide();

$('#result').fadeIn(600); $('#result').text('Predictio n: '+data); console.log('Success!');

},

});

});

});

**Final.css:**

.img-preview { width: 256px; height: 256px; position: relative; border: 5px solid #F8F8F8; box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);margin-top:

|  |  |
| --- | --- |
| 1em;  margin-bottom: 1em;  }    .img- preview>di v {width:  100%; height: 100%; background-size: 256px 256px; background-repeat:  no-repeat; background-position:  center;  }    input[type="fi le"] { display:  none;  }    .upload-label{ display: inline- block; padding: 12px 30px; background: #28272c;color:  #fff; font-size: 1em;  transition:  all .4s; cursor: pointer; } | Page 69 of 107 |

.upload- label:hover{ background: #C2C5A8; color:

#39D2B4;

}

.loader { border: 8px solid #f3f3f3; /\* Light grey \*/ border-top: 8px solid #28272c; /\* Blue \*/ border-radius: 50%; width: 50px; height: 50px; animation: spin 1s linear infinite;

}

@keyframes spin { 0% { transform:

rotate(0deg); } 100% { transform:

rotate(360deg); }

}

**Python – app.py:**

import os import numpy as npimport pandas as pd

from tensorflow.keras.models import load\_model

# from tensorflow.keras.preprocessing import imagefrom werkzeug.utils import secure\_filename

from flask import Flask,

render\_template, requestapp = Flask( name )

#load both the vegetable and fruit modelsmodel = load\_model("vegetable.h5") model1=load\_model("fruit.h

5")

#home page @app.rout

e('/')def home():

return render\_template('home.html')

#prediction page @app.route('/predi ction')def prediction():

return render\_template('predict.html')

@app.route('/predict',method s=['POST'])def predict():

if request.method == 'POST': # Get the file from post requestf = request.files['image']

# Save the file to ./uploads basepath = os.path.dirname( file )file\_path = os.path.join(

basepath, 'uploads',

secure\_filename(f.filename)) f.save(file\_path)

img = image.load\_img(file\_path, target\_size=(128, 128))

x =

image.img\_to\_array(i mg) x = np.expand\_dims(x, axis=0)

plant=request.form['pl ant']print(plant) if(plant=="vegetable")

:

preds =

model.predict(x) preds=np.argmax (preds) print(preds) df=pd.read\_excel('precaution s - veg.xlsx')

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

else:

preds = model1.predict(x) preds=np.argmax(preds)

df=pd.read\_excel('precautions

- fruits.xlsx')

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

return

df.iloc[preds]['caution']if name == " main ": app.run(debug=False)

**DEPLOYMENT MODEL CODE:**

**Fruit model:**

ls

sample\_dat a/pwd

'/home/wsuser/work'

!pip install keras==2.7.0

!pip install tensorflow==2.5.0

Looking in indexes: https://pypi.org/simple, https://us- python.pkg.dev/colabwheels/public/simple/

Requirement already satisfied: keras==2.7.0 in /usr/local/lib/python3.7/dist- packages (2.7.0)Looking in indexes: https://pypi.org/simple, https://us-

python.pkg.dev/colab wheels/public/simple/ Requirement already satisfied: tensorflow==2.5.0 in

/usr/local/lib/python3.7/dist-packages(2.5.0)

Requirement already satisfied: h5py~=3.1.0 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (3.1.0)

Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages

(fromtensorflow==2.5.0) (3.19.6)

Requirement already satisfied: typing-extensions~=3.7.4 in

/usr/local/lib/python3.7/dist packages(from tensorflow==2.5.0) (3.7.4.3)

Requirement already satisfied: keras-nightly~=2.5.0.dev in

/usr/local/lib/python3.7/dist packages(from tensorflow==2.5.0)

(2.5.0.dev2021032900)

Requirement already satisfied: flatbuffers~=1.12.0 in

/usr/local/lib/python3.7/dist-packages(from tensorflow==2.5.0) (1.12) Requirement already satisfied: gast==0.4.0 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (0.4.0)

Requirement already satisfied: absl-py~=0.10 in /usr/local/lib/python3.7/dist-packages

(fromtensorflow==2.5.0) (0.15.0)

Requirement already satisfied: astunparse~=1.6.3 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (1.6.3)

Requirement already satisfied: tensorflow-estimator<2.6.0,>=2.5.0rc0 in

/usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (2.5.0) Requirement already satisfied: tensorboard~=2.5 in /usr/local/lib/python3.7/dist- packages (from tensorflow==2.5.0)(2.9.1)

Requirement already satisfied: opt-einsum~=3.3.0 in

/usr/local/lib/python3.7/dist-packages(from tensorflow==2.5.0) (3.3.0) Requirement already satisfied: six~=1.15.0 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (1.15.0) Requirement already satisfied: google-pasta~=0.2 in

/usr/local/lib/python3.7/dist-packages(from tensorflow==2.5.0) (0.2.0)

Requirement already satisfied: grpcio~=1.34.0 in /usr/local/lib/python3.7/dist-packages (fromtensorflow==2.5.0) (1.34.1)

Requirement already satisfied: wrapt~=1.12.1 in /usr/local/lib/python3.7/dist-packages (fromtensorflow==2.5.0) (1.12.1)

Requirement already satisfied: termcolor~=1.1.0 in /usr/local/lib/python3.7/dist-packages

(fromtensorflow==2.5.0) (1.1.0)

Requirement already satisfied: keras-preprocessing~=1.1.2 in

/usr/local/lib/python3.7/distpackages (from tensorflow==2.5.0) (1.1.2) Requirement already satisfied: wheel~=0.35 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (0.38.3)

Requirement already satisfied: numpy~=1.19.2 in /usr/local/lib/python3.7/dist-packages

(fromtensorflow==2.5.0) (1.19.5)

Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist- packages (fromh5py~=3.1.0->tensorflow==2.5.0) (1.5.2)

Requirement already satisfied: google-auth<3,>=1.6.3 in

/usr/local/lib/python3.7/dist packages(from tensorboard~=2.5-

>tensorflow==2.5.0) (2.14.1)

Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5- >tensorflow==2.5.0) (0.6.1)Requirement already satisfied: tensorboard- plugin-wit>=1.6.0 in

/usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5- >tensorflow==2.5.0) (1.8.1)Requirement already satisfied: google-auth- oauthlib<0.5,>=0.4.1 in

/usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5-

>tensorflow==2.5.0) (0.4.6) Requirement already satisfied: werkzeug>=1.0.1 in

/usr/local/lib/python3.7/dist-packages (fromtensorboard~=2.5-

>tensorflow==2.5.0) (1.0.1)

Requirement already satisfied: markdown>=2.6.8 in

/usr/local/lib/python3.7/dist-packages(from tensorboard~=2.5-

>tensorflow==2.5.0) (3.4.1)

Requirement already satisfied: requests<3,>=2.21.0 in

/usr/local/lib/python3.7/dist packages(from tensorboard~=2.5-

>tensorflow==2.5.0) (2.23.0)

Requirement already satisfied: setuptools>=41.0.0 in

/usr/local/lib/python3.7/dist-packages(from tensorboard~=2.5-

>tensorflow==2.5.0) (57.4.0)

Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist- packages (from google-auth<3,>=1.6.3->tensorboard~=2.5->tensorflow==2.5.0)

(4.9) Requirement already satisfied: pyasn1-modules>=0.2.1 in

/usr/local/lib/python3.7/dist packages (from google- auth<3,>=1.6.3-

>tensorboard~=2.5->tensorflow==2.5.0) (0.2.8) Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.7/dist packages (from google- auth<3,>=1.6.3-

>tensorboard~=2.5->tensorflow==2.5.0) (5.2.0) Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist packages

(from google-auth- oauthlib<0.5,>=0.4.1->tensorboard~=2.5-

>tensorflow==2.5.0) (1.3.1)

Requirement already satisfied: importlib-metadata>=4.4 in

/usr/local/lib/python3.7/dist packages(from markdown>=2.6.8->tensorboard~=2.5- >tensorflow==2.5.0) (4.13.0) Requirement already satisfied: zipp>=0.5 in

/usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.4- >markdown>=2.6.8->tensorboard~=2.5->tensorflow==2.5.0) (3.10.0)

Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in

/usr/local/lib/python3.7/dist packages(from pyasn1-modules>=0.2.1->google- auth<3,>=1.6.3->tensorboard~=2.5- >tensorflow==2.5.0) (0.4.8)

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in

/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.5-

>tensorflow==2.5.0) (1.24.3)

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist- packages (fromrequests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0) (2.10)

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist- packages (fromrequests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0)

(3.0.4) Requirement already satisfied: certifi>=2017.4.17 in

/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0) (2022.9.24) Requirement alreadysatisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist- packages (from requests- oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-

>tensorboard~=2.5- >tensorflow==2.5.0) (3.2.2)

Image Augmentation

from tensorflow.keras.preprocessing.image import ImageDataGenerator train\_datagen=ImageDataGenerator(rescale=1./255,zoom\_range=0.2,horizontal

\_flip=True,v ertical\_flip=False)

test\_datagen=ImageDataGenerator(rescale=

1./255)ls pwd

/content import os, types import pandas as pd from botocore.client import Configimport ibm\_boto3 def iter (self): return 0

# @hidden\_cell

# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.

# You might want to remove those credentials before you share the notebook. client\_4ff9f1114db24196a9abd4f5c1f0b60a = ibm\_boto3.client(service\_name='s3',

ibm\_api\_key\_id='j4lNXssktSSxQiDx3pbNR\_eFi1SMCDE6MFnBQ\_ EmNCDM',

ibm\_auth\_endpoint="https://iam.cloud.ibm.com/oidc/token", config=Config(signature\_version='oauth'), endpoint\_url='https://s3.private.us.cloud-object- storage.appdomain.cloud') streaming\_body\_1 =

client\_4ff9f1114db24196a9abd4f5c1f0b60a.get\_object(Bucket='trainmodel- donotdelete-pr-cbqe37eh8gzesa', Key='fruit-dataset.zip')['Body']

# Your data file was loaded into a botocore.response.StreamingBody object. # Please read the documentation of ibm\_boto3 and pandas to learn more about the possibil ities to load the data.

# ibm\_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/ # pandas documentation:<http://pandas.pydata.org/>

from io import BytesIOimport

zipfile unzip =

zipfile.ZipFile(BytesIO(streaming\_body\_1.read()),

"r")file\_paths = unzip.namelist() for path in file\_paths:

unzip.extract(pat

h) pwd '/home/wsuser/w ork'import os

filenames = os.listdir('/home/wsuser/work/fruit-dataset/train') x\_train=train\_datagen.flow\_from\_directory("/home/wsuser/work/fruit dataset/train",target\_size=(128,128),class\_mode='categorical',batch\_size=24 ) Found 5384images belonging to 6 classes.

x\_test=test\_datagen.flow\_from\_directory(r"/home/wsuser/work/fruit dataset/test",target\_size=(128,128), class\_mode='categorical',batch\_size =24) Found 1686 images belonging to 6 classes.x\_train.class\_indices

{'Apple Black\_rot': 0, 'Applehealthy': 1, 'Corn\_(maize) Northern\_Leaf\_Blight': 2, 'Corn\_(maize) healthy': 3, 'Peach Bacterial\_spot': 4, 'Peach healthy': 5}

CNN

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import

Dense,Convolution2D,MaxPooling2D,Flattenmodel=Sequential() model.add(Convolution2D(32,(3,3),input\_shape=(128,128,3),activatio n='relu')) model.add(MaxPooling2D(pool\_size=(2,2))) model.add(Flatten(

))

model.summary() Model:

"sequential\_1"

Layer

(type)Output Shape Param #

===============================================================

== conv2d\_1 (Conv2D) (None, 126, 126, 32) 896

max\_pooling2d (MaxPooling2D (None, 63, 63, 32) 0

)

flatten (Flatten) (None, 127008) 0

=============================================================== ==

Total params: 896

Trainable params: 896 Non-trainable params: 0 32\*(3\*3\*3+1)

896

#Hidden Layers model.add(Dense(300,activation='relu')) model.add(Dense(150,activation='relu'))

Output Layers

model.add(Dense(6,activation='softmax'))

model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['accu racy']) len(x\_train)

225

1238/24

51.583333333333336 model.fit\_generator(x\_train,steps\_per\_epoch=len(x\_train),validation\_data=x\_te st,validatio n\_steps=len(x\_test),epochs=10)

/tmp/wsuser/ipykernel\_164/1582812018.py:1: UserWarning:

`Model.fit\_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supportsgenerators.

model.fit\_generator(x\_train,steps\_per\_epoch=len(x\_train),validation\_data=x\_test,validat ion

\_steps=len(x\_test),epochs=1

0)Epoch 1/10

225/225 [==============================] - 118s 520ms/step - loss: 0.8920 - accuracy: 0.8094 - val\_loss: 0.2273 - val\_accuracy:

0.9235Epoch 2/10

225/225 [==============================] - 116s 515ms/step - loss: 0.2367 - accuracy: 0.9179 - val\_loss: 0.2056 - val\_accuracy:

0.9324Epoch 3/10

225/225 [==============================] - 116s 517ms/step - loss: 0.1970 - accuracy: 0.9337 - val\_loss: 0.4972 - val\_accuracy:

0.8754Epoch 4/10

225/225 [==============================] - 117s 521ms/step - loss: 0.1688 - accuracy: 0.9422 - val\_loss: 0.2279 - val\_accuracy:

0.9217Epoch 5/10

225/225 [==============================] - 116s 516ms/step - loss: 0.1438 - accuracy: 0.9487 - val\_loss: 0.1685 - val\_accuracy:

0.9484Epoch 6/10

225/225 [==============================] - 117s 518ms/step - loss: 0.1362 - accuracy: 0.9556 - val\_loss: 0.1176 - val\_accuracy:

0.9662Epoch 7/10

225/225 [==============================] - 116s 515ms/step - loss: 0.1282 - accuracy: 0.9590 - val\_loss: 0.5466 - val\_accuracy:

0.8387Epoch 8/10

225/225 [==============================] - 116s 514ms/step - loss: 0.1282 - accuracy: 0.9597 - val\_loss: 0.1194 - val\_accuracy:

0.9620Epoch 9/10

225/225 [==============================] - 116s 514ms/step - loss: 0.1141 - accuracy: 0.9616 - val\_loss: 0.1478 - val\_accuracy:

0.9508Epoch 10/10

225/225 [==============================] - 116s 516ms/step - loss: 0.0927 - accuracy: 0.9695 - val\_loss: 0.0772 - val\_accuracy: 0.9751 <keras.callbacks.History at 0x7f71e8184070>

Saving Model

ls fruit-dataset/ model.save('fruit. h5')

!tar -zcvf Train-model\_new.tgz

fruit.h5fruit.h5

ls -1 fruit- dataset/

fruit.h5

Train-model\_new.tgz

IBM Cloud Deployment Model

!pip install watson-machine-learning-client – upgradeCollecting watson-machine-learning- client

Downloading watson\_machine\_learning\_client-1.0.391-py3-none-any.whl (538 kB)

|████████████████████████████████| 538 kB 21.2 MB/s eta

0:00:01 Requirement already satisfied: tqdm in /opt/conda/envs/Python- 3.9/lib/python3.9/site packages(from watson-machine-learning-client) (4.62.3)

Requirement already satisfied: certifi in /opt/conda/envs/Python- 3.9/lib/python3.9/site packages(from watson-machine-learning-client) (2022.9.24)

Requirement already satisfied: requests in /opt/conda/envs/Python- 3.9/lib/python3.9/sitepackages (from watson-machine-learning-client) (2.26.0)

Requirement already satisfied: tabulate in /opt/conda/envs/Python- 3.9/lib/python3.9/sitepackages (from watson-machine-learning-client) (0.8.9)

Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python

3.9/lib/python3.9/site- packages (from watson-machine-learning-client) (2.11.0)

Requirement already satisfied: pandasin /opt/conda/envs/Python-

3.9/lib/python3.9/site packages (from watson-machine-learning- client) (1.3.4)

Requirement already satisfied: lomond in /opt/conda/envs/Python- 3.9/lib/python3.9/sitepackages (from watson-machine-learning-client)

(0.3.3)

Requirement already satisfied: boto3 in /opt/conda/envs/Python-

3.9/lib/python3.9/site packages(from watson-machine-learning-client) (1.18.21)

Requirement already satisfied: urllib3 in /opt/conda/envs/Python-

3.9/lib/python3.9/site packages(from watson-machine-learning-client) (1.26.7) Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in

/opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from boto3->watson- machine-learning-client) (0.10.0) Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python 3.9/lib/python3.9/site- packages (from boto3->watson-machine-learning-client) (0.5.0) Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client)

(1.21.41) Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from

botocore<1.22.0,>=1.21.21->boto3->watson machine-learning-client) (2.8.2)

Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-

3.9/lib/python3.9/site packages (from python-dateutil<3.0.0,>=2.1- >botocore<1.22.0,>=1.21.21->boto3->watsonmachine-learning-client) (1.15.0)

Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in

/opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from ibm-cos-sdk- >watson-machine-learning-client) (2.11.0) Requirement already satisfied: ibm- cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python 3.9/lib/python3.9/site- packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)

Requirement already satisfied: charset-normalizer~=2.0.0 in

/opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from requests->watson- machine-learning-client) (2.0.4) Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python 3.9/lib/python3.9/site- packages (from requests->watson- machine-learning-client) (3.3) Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from pandas->watson- machine-learning-client) (2021.3) Requirement already satisfied: numpy>=1.17.3 in

/opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from pandas- >watson-machine-learning-client) (1.19.5) Installing collected packages: watson-machine-learning-client Successfully installed watson-machine- learning-client-1.0.391

from ibm\_watson\_machine\_learning import

APIClientwml\_credentials = {

"url": "https://us-south.ml.cloud.ibm.com",

"apikey":"0P3XkyCFYqABnc48BNG2ReoGAJy- oDXDRuULl4Y\_zFxa"

} client = APIClient(wml\_credentials) def guid\_from\_space\_name(client, space\_name):

space = client.spaces.get\_details()

return(next(item for item in space['resources'] if

item['entity']["name"]==space\_name)['metadata']['id'])

space\_uid = guid\_from\_space\_name(client, 'Trainmodel')print("Space UID = " + space\_uid) Space UID = 616c7d74-e99b-4c09-9922-27394a62c2d0 client.set.default\_space(space\_uid)

‘SUCCESS’

client.software\_specifications.list()

NAME ASSET\_ID TYPE

default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base kernel-spark3.2- scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base pytorch-onnx\_1.3-py3.7- edt 069ea134-3346- 5748-b513-49120e15d288 base scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344- eb7b665ff687 base spark-mllib\_3.0-scala\_2.12

09f4cff0-90a7-5899-b9ed-1ef348aebdee base pytorch-onnx\_rt22.1-py3.9

0b848dd4-e681-5599-be41-b5f6fccc6471 base ai-function\_0.1-py3.60cdb0f1e-

5376-4f4d-92dd-da3b69aa9bda base shiny-r3.6 0e6e79df-875e-4f24-8ae9-

62dcc2148306 base

tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base pytorch\_1.1- py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base tensorflow\_1.15-py3.6-ddl 111e41b3- de2d-5422-a4d6-bf776828c4b7 base runtime-22.1-py3.9 12b83a17-24d8-5082-900f- 0ab31fbfd3cb base scikit- learn\_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base default\_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base pytorch-onnx\_1.3-py3.6 1bc6029a- cc97-56da-b8e0-39c3880dbbe7 base kernel-spark3.3-r3.6 1c9e5454-f216-59dd- a20e- 474a5cdf5988 base pytorch-onnx\_rt22.1-py3.9-edt 1d362186-7ad5-5b59- 8b6c-9d0880bde37f base tensorflow\_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5- 3fbdf1665666 base spark-mllib\_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base tensorflow\_2.4-py3.8-horovod 217c16f6-178f- 56bf-824a-b19f20564c49 base runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0- da66306ce658 base do\_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base autoai-ts\_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base tensorflow\_1.15-py3.6 2b73a275- 7cbf-420b- a912-eae7f436e0bc base kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c- a491-482c8368839a base pytorch\_1.2-py3.6 2c8ef57d-2687-4b7d-acce- 01f94976dac1 base spark-mllib\_2.3 2e51f700- bca0-4b0d-88dc-5c6791338875 base pytorch-onnx\_1.1-py3.6-edt 32983cea-3f32-4400-8965- dde874a8d67e base spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base spark- mllib\_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base xgboost\_0.82-py3.6 39e31acd-5f30- 41dc-ae44-60233c80306e base pytorch-onnx\_1.2-py3.6-edt

40589d0e-7019-4e28-8daa- fb03b6f4fe12 base default\_r36py38 41c247d3-45f8- 5a71-b065-8580229facf0 base autoai-ts\_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base autoai- obm\_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base pmml-3.0\_4.3 493bcb95-16f1-5bc5-bee8- 81b8af80e9c7 base spark-mllib\_2.4-r\_3.6 49403dff-

92e9-4c87-a3d7-a42d0021c095 base xgboost\_0.90-py3.6 4ff8d6c2-1343-4c18- 85e1-689c965304d3 base pytorch-onnx\_1.1-py3.6 50f95b2a-bc16-43bb-bc94- b0bed208c60b base autoai-ts\_3.9-py3.8 52c57136-80fa-572e-8728- a5e7cbb42cde base spark-mllib\_2.4-scala\_2.11 55a70f99-7320-4be5-9fb9- 9edb5a443af5 base spark-mllib\_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base autoai-obm\_2.0 5c2e37fa-80b8-

5e77-840f-d912469614ee base spss-modeler\_18.1 5c3cad7e-507f-4b2a-a9a3- ab53a21dee8b basecuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base autoai-kb\_3.1-py3.7 632d4b22-10aa- 5180-88f0-f52dfb6444d7 base pytorch- onnx\_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-

ea90a478456b base spark-mllib\_2.3-r\_3.6 6586b9e3-ccd6-4f92-900f- 0f8cb2bd6f0c base tensorflow\_2.4-py3.7 65e171d7-72d1-55d9-8ebb- f813d620c9bb base spss-modeler\_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base



Note: Only first 50 records were displayed. To display more use 'limit' parameter. software\_space\_uid =

client.software\_specifications.get\_uid\_by\_name("tensorflow\_rt22.1-py3.9") software\_spec\_uid ‘1eb25b84-d6ed-5dde-b6a5-

3fbdf1665666'ls

fruit-dataset/ fruit.h5 Train-model\_new.tgz

model\_details = client.repository.store\_model(model= 'Train-model\_new.tgz',

meta\_props={client.repository.ModelMetaNames.NAME:"CNN", client.repository.ModelMetaNames.TYPE:"tensorflow\_2.7",

client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID:software\_space\_ uid} ) model\_id = client.repository.get\_model\_id(model\_details) model\_id

'd0aeb6a2-e89c-4f8d-bf2f- a28ca4ea3cca'ls fruit-dataset/ fruit.h5 Train- model\_new.tgzTest The Model import numpy as np

from tensorflow.keras.models import load\_model from tensorflow.keras.preprocessing import imagemodel=load\_model('fruit.h5')

#@title img=image.load\_img(r"C:\Users\LENOVO\Desktop\fruit-dataset\fruit dataset\test\00fca0da-2db3-481b-b98a

9b67bb7b105c RS\_HL

7708.JPG",target\_size=(128,128))img

img=image.load\_img(r"C:\Users\LENOVO\Desktop\ibm\Dataset Plant Disease\fruit dataset\fruit-dataset\test\Apple

healthy\0adc1c5b-8958-47c0-a152- f28078c214f1 RS\_HL7825.JPG",target\_size=(128,128)) img



x=image.img\_to\_array(i mg)X

array([[[ 99., 86., 106.],

[101., 88., 108.],

[118., 105., 125.],

|  |  |
| --- | --- |
| ...,  [ 92., 83., 102.],  [ 93., 84., 103.],  [ 89., 80., 99.]],  [[ 96., 83., 103.],  [ 87., 74., 94.],  [102., 89., 109.],  ...,  [ 88., 79., 98.], [ 89., 80., 99.],  [ 83., 74., 93.]],  [[ 86., 73., 93.],  [ 88., 75., 95.],  [ 98., 85., 105.],  ...,  [107., 98., 117.],  [ 96., 87., 106.],  [ 96., 87., 106.]],  ...,  [[172., 175., 194.],  [173., 176., 195.],  [175., 178., 197.],  ...,  [179., 180., 198.],  [184., 185., 203.],  [179., 180., 198.]], [[172., 175., 194.],  [170., 173., 192.],  [173., 176., 195.],  ...,  [178., 179., 197.],  [182., 183., 201.],  [178., 179., 197.]], [[169., 172., 191.],  [166., 169., 188.],  [168., 171., 190.],  ...,  [187., 188., 206.],  [185., 186., 204.],  [186., 187., 205.]]], dtype=float32) | Page 90 of 107 |

|  |  |
| --- | --- |
| x=np.expand\_dims(x,axis=0)X  array([[[[ 99., 86., 106.],  [101., 88., 108.],  [118., 105., 125.],  ...,  [ 92., 83., 102.], | Page 91 of 107 |

[ 93., 84., 103.],

[ 89., 80., 99.]],

[[ 96., 83., 103.],

[ 87., 74., 94.],

[102., 89., 109.],

...,

[ 88., 79., 98.],

[ 89., 80., 99.],

[ 83., 74., 93.]],

[[ 86., 73., 93.],

[ 88., 75., 95.],

[ 98., 85., 105.],

...,

[107., 98., 117.],

[ 96., 87., 106.],

[ 96., 87., 106.]],

...,

[[172., 175., 194.],

[173., 176., 195.],

[175., 178., 197.],

...,

[179., 180., 198.],

[184., 185., 203.],

[179., 180., 198.]], [[172., 175., 194.],

[170., 173., 192.],

[173., 176., 195.],

...,

[178., 179., 197.],

[182., 183., 201.],

[178., 179., 197.]], [[169., 172., 191.],

[166., 169., 188.],

[168., 171., 190.],

...,

[187., 188., 206.],

[185., 186., 204.],

[186., 187., 205.]]]], dtype=float32) y=np.argmax(model.predict(x),axis=1)

1/1 [==============================] - 0s 105ms/step

|  |
| --- |
| x\_train.class\_indices  {'Apple Black\_rot': 0, 'Applehealthy': 1, 'Corn\_(maize) Northern\_Leaf\_Blight': 2,  'Corn\_(maize) healthy': 3, 'Peach Bacterial\_spot': 4, 'Peach healthy': 5} index=['Apple Black\_rot','Apple  healthy','Corn\_(maize) Northern\_Leaf\_Blight','Corn  \_(maize) healthy','Peach Bacterial\_spot','Peach healthy']index[y[0]]  Page 93 of 107 |

'Apple healthy'

img=image.load\_img(r"C:\LENOVO\Desktop\ibm\Dataset Plant Disease\fruit-dataset\fruitdataset\test\Peach healthy\0a2ed402-5d23-

4e8d-bc98-

b264aea9c3fb Rutg.\_HL 2471.JPG",target\_size=(128,128)) x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) y=np.argmax(model.predict(x),axis=1)

index=['Apple Black\_rot','Apple healthy''Peach Bacterial\_spot','Peach

healthy']index[y[0]]

1/1 [==============================] - 0s 26ms/step

'Peach

health

y'import os

from tensorflow.keras.models import load\_model from

tensorflow.keras.preprocessing import imagefrom flask import

Flask,render\_template,request app=Flask( name ) model=load\_model("fruit.h5")

@app.route(

'/')def index(): return render\_template("index.html") @app.route('/predict',methods=['GET','P OST'])def upload(): if request.method=='POST':

f=request.files['image'] basepath=os.path.dirname(' file ') filepath=os.path.join(basepath,'uploads',f.file name) f.save(filepath) img=image.load\_img(filepath,target\_size=(1 28,128))x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) pred=np.argmax(model.predict(x),axis=1) index=['Apple Black\_rot','Apple healthy',

,'Peach Bacterial\_spot','Peachhealthy'] text="The Classified Fruit disease is : "

+str(index[pred[0]])return text

|  |  |
| --- | --- |
| if name ==' main ': app.run(debug=False)      **vegetable model :**    ls  sample\_dat a/pwd  '/home/wsuser/work'  !pip install keras==2.7.0 | Page 95 of 107 |

!pip install tensorflow==2.5.0

Looking in indexes: https://pypi.org/simple, https://us- python.pkg.dev/colabwheels/public/simple/

Requirement already satisfied: keras==2.7.0 in /usr/local/lib/python3.7/dist- packages (2.7.0)Looking in indexes: https://pypi.org/simple, https://us-

python.pkg.dev/colab wheels/public/simple/ Requirement already satisfied: tensorflow==2.5.0 in

/usr/local/lib/python3.7/dist-packages(2.5.0)

Requirement already satisfied: h5py~=3.1.0 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (3.1.0)

Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages

(fromtensorflow==2.5.0) (3.19.6)

Requirement already satisfied: typing-extensions~=3.7.4 in

/usr/local/lib/python3.7/dist packages(from tensorflow==2.5.0) (3.7.4.3)

Requirement already satisfied: keras-nightly~=2.5.0.dev in

/usr/local/lib/python3.7/dist packages(from tensorflow==2.5.0)

(2.5.0.dev2021032900)

Requirement already satisfied: flatbuffers~=1.12.0 in

/usr/local/lib/python3.7/dist-packages(from tensorflow==2.5.0) (1.12) Requirement already satisfied: gast==0.4.0 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (0.4.0)

Requirement already satisfied: absl-py~=0.10 in /usr/local/lib/python3.7/dist-packages

(fromtensorflow==2.5.0) (0.15.0)

Requirement already satisfied: astunparse~=1.6.3 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (1.6.3)

Requirement already satisfied: tensorflow-estimator<2.6.0,>=2.5.0rc0 in

/usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (2.5.0) Requirement already satisfied: tensorboard~=2.5 in /usr/local/lib/python3.7/dist- packages (from tensorflow==2.5.0)(2.9.1)

Requirement already satisfied: opt-einsum~=3.3.0 in

/usr/local/lib/python3.7/dist-packages(from tensorflow==2.5.0) (3.3.0) Requirement already satisfied: six~=1.15.0 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (1.15.0) Requirement already satisfied: google-pasta~=0.2 in

/usr/local/lib/python3.7/dist-packages(from tensorflow==2.5.0) (0.2.0)

Requirement already satisfied: grpcio~=1.34.0 in /usr/local/lib/python3.7/dist-packages (fromtensorflow==2.5.0) (1.34.1)

Requirement already satisfied: wrapt~=1.12.1 in /usr/local/lib/python3.7/dist-packages (fromtensorflow==2.5.0) (1.12.1)

Requirement already satisfied: termcolor~=1.1.0 in /usr/local/lib/python3.7/dist-packages

(fromtensorflow==2.5.0) (1.1.0)

Requirement already satisfied: keras-preprocessing~=1.1.2 in

/usr/local/lib/python3.7/distpackages (from tensorflow==2.5.0) (1.1.2) Requirement already satisfied: wheel~=0.35 in /usr/local/lib/python3.7/dist- packages (fromtensorflow==2.5.0) (0.38.3)

Requirement already satisfied: numpy~=1.19.2 in /usr/local/lib/python3.7/dist-packages (fromtensorflow==2.5.0) (1.19.5)

Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist- packages (fromh5py~=3.1.0->tensorflow==2.5.0) (1.5.2)

Requirement already satisfied: google-auth<3,>=1.6.3 in

/usr/local/lib/python3.7/dist packages(from tensorboard~=2.5-

>tensorflow==2.5.0) (2.14.1)

Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5- >tensorflow==2.5.0) (0.6.1)Requirement already satisfied: tensorboard- plugin-wit>=1.6.0 in

/usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5- >tensorflow==2.5.0) (1.8.1)Requirement already satisfied: google-auth- oauthlib<0.5,>=0.4.1 in

/usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5-

>tensorflow==2.5.0) (0.4.6) Requirement already satisfied: werkzeug>=1.0.1 in

/usr/local/lib/python3.7/dist-packages (fromtensorboard~=2.5-

>tensorflow==2.5.0) (1.0.1)

Requirement already satisfied: markdown>=2.6.8 in

/usr/local/lib/python3.7/dist-packages(from tensorboard~=2.5-

>tensorflow==2.5.0) (3.4.1)

Requirement already satisfied: requests<3,>=2.21.0 in

/usr/local/lib/python3.7/dist packages(from tensorboard~=2.5-

>tensorflow==2.5.0) (2.23.0)

Requirement already satisfied: setuptools>=41.0.0 in

/usr/local/lib/python3.7/dist-packages(from tensorboard~=2.5-

>tensorflow==2.5.0) (57.4.0)

Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist- packages (from google-auth<3,>=1.6.3->tensorboard~=2.5->tensorflow==2.5.0)

(4.9) Requirement already satisfied: pyasn1-modules>=0.2.1 in

/usr/local/lib/python3.7/dist packages (from google- auth<3,>=1.6.3-

>tensorboard~=2.5->tensorflow==2.5.0) (0.2.8) Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.7/dist packages (from google- auth<3,>=1.6.3-

>tensorboard~=2.5->tensorflow==2.5.0) (5.2.0) Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist packages

(from google-auth- oauthlib<0.5,>=0.4.1->tensorboard~=2.5-

>tensorflow==2.5.0) (1.3.1)

Requirement already satisfied: importlib-metadata>=4.4 in

/usr/local/lib/python3.7/dist packages(from markdown>=2.6.8->tensorboard~=2.5- >tensorflow==2.5.0) (4.13.0) Requirement already satisfied: zipp>=0.5 in

/usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.4- >markdown>=2.6.8->tensorboard~=2.5->tensorflow==2.5.0) (3.10.0)

Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in

/usr/local/lib/python3.7/dist packages(from pyasn1-modules>=0.2.1->google- auth<3,>=1.6.3->tensorboard~=2.5- >tensorflow==2.5.0) (0.4.8)

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in

/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.5-

>tensorflow==2.5.0) (1.24.3)

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist- packages (fromrequests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0) (2.10)

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist- packages (fromrequests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0)

(3.0.4) Requirement already satisfied: certifi>=2017.4.17 in

/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0) (2022.9.24) Requirement alreadysatisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist- packages (from requests- oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1-

>tensorboard~=2.5- >tensorflow==2.5.0) (3.2.2)

Image Augmentation from tensorflow.keras.preprocessing.image import ImageDataGenerator train\_datagen=ImageDataGenerator(rescale=1./255,zoom\_range=0.2,horizontal

\_flip=True,v ertical\_flip=False)

test\_datagen=ImageDataGenerator(rescale=

1./255)ls pwd

/content import os, types import pandas as pd from botocore.client import Configimport ibm\_boto3 def iter (self): return 0#

@hidden\_cell

# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.

# You might want to remove those credentials before you share the notebook. client\_4ff9f1114db24196a9abd4f5c1f0b60a = ibm\_boto3.client(service\_name='s3',

ibm\_api\_key\_id='j4lNXssktSSxQiDx3pbNR\_eFi1SMCDE6MFnBQ\_ EmNCDM',

ibm\_auth\_endpoint="https://iam.cloud.ibm.com/oidc/token", config=Config(signature\_version='oauth'), endpoint\_url='https://s3.private.us.cloud-object- storage.appdomain.cloud') streaming\_body\_1 =

client\_4ff9f1114db24196a9abd4f5c1f0b60a.get\_object(Bucket='trainmodel- donotdelete-pr-cbqe37eh8gzesa', Key='vegetable-dataset.zip')['Body'] # Your data file was loaded into a botocore.response.StreamingBody object. # Please read the documentation of ibm\_boto3 and pandas to learn more about the possibil ities to load the data.

# ibm\_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/ # pandas documentation:<http://pandas.pydata.org/>

from io import BytesIOimport

zipfile

unzip =

zipfile.ZipFile(BytesIO(streaming\_body\_1.read()),

"r")file\_paths = unzip.namelist() for path in file\_paths: unzip.extract(pat

h) pwd '/home/wsuser/w ork'import os

filenames = os.listdir('/home/wsuser/work/vegetable-dataset/train') x\_train=train\_datagen.flow\_from\_directory("/home/wsuser/work/vegetable dataset/train",target\_size=(128,128),class\_mode='categorical',batch\_size=24 ) Found 5384images belonging to 6 classes. x\_test=test\_datagen.flow\_from\_directory(r"/home/wsuser/work/vegetable dataset/test",target\_size=(128,128), class\_mode='categorical',batch\_size =24) Found 1686 images belonging to 6 classes.x\_train.class\_indices

{'Tomato Blight': 0, 'Tomato healthy': 1, 'Corn\_(maize) Northern\_Leaf\_Blight': 2, 'Corn\_(maize) healthy': 3, 'Potato Blight': 4, 'Potato healthy': 5}

CNN

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten model=Sequential()

model.add(Convolution2D(32,(3,3),input\_shape=(128,128,3),activation='relu')) model.add(MaxPooling2D(pool\_size=(2,2)))

model.add(Flatten()) model.summary()

Model: "sequential\_1"

Layer (type)

Output Shape Param #

================================================================= conv2d\_1 (Conv2D) (None, 126, 126, 32) 896

max\_pooling2d (MaxPooling2D (None, 63, 63, 32) 0

)

flatten (Flatten) (None, 127008) 0

=================================================================

Total params: 896

Trainable params: 896

Non-trainable params: 0

32\*(3\*3\*3+1)

896

#Hidden Layers model.add(Dense(300,activation='relu')) model.add(Dense(150,activation='relu'))

Output Layers

model.add(Dense(6,activation='softmax'))

model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['accuracy']) len(x\_train) 225

1238/24

51.583333333333336 model.fit\_generator(x\_train,steps\_per\_epoch=len(x\_train),validation\_data=x\_test,validatio n\_steps=len(x\_test),epochs=10)

/tmp/wsuser/ipykernel\_164/1582812018.py:1: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

model.fit\_generator(x\_train,steps\_per\_epoch=len(x\_train),validation\_data=x\_test,validation \_steps=len(x\_test),epochs=10)

Epoch 1/10

225/225 [==============================] - 118s 520ms/step - loss: 0.8920 - accuracy: 0.8094 - val\_loss: 0.2273 - val\_accuracy: 0.9235

Epoch 2/10

225/225 [==============================] - 116s 515ms/step - loss: 0.2367 - accuracy: 0.9179 - val\_loss: 0.2056 - val\_accuracy: 0.9324

Epoch 3/10

225/225 [==============================] - 116s 517ms/step - loss: 0.1970 - accuracy: 0.9337 - val\_loss: 0.4972 - val\_accuracy: 0.8754

Epoch 4/10

225/225 [==============================] - 117s 521ms/step - loss: 0.1688 - accuracy: 0.9422 - val\_loss: 0.2279 - val\_accuracy: 0.9217

Epoch 5/10

225/225 [==============================] - 116s 516ms/step - loss: 0.1438 - accuracy: 0.9487 - val\_loss: 0.1685 - val\_accuracy: 0.9484

Epoch 6/10

225/225 [==============================] - 117s 518ms/step - loss: 0.1362 - accuracy: 0.9556 - val\_loss: 0.1176 - val\_accuracy: 0.9662

Epoch 7/10

225/225 [==============================] - 116s 515ms/step - loss: 0.1282 - accuracy: 0.9590 - val\_loss: 0.5466 - val\_accuracy: 0.8387

Epoch 8/10

225/225 [==============================] - 116s 514ms/step - loss: 0.1282 - accuracy: 0.9597 - val\_loss: 0.1194 - val\_accuracy: 0.9620

Epoch 9/10

225/225 [==============================] - 116s 514ms/step - loss: 0.1141 - accuracy: 0.9616 - val\_loss: 0.1478 - val\_accuracy: 0.9508

Epoch 10/10

225/225 [==============================] - 116s 516ms/step - loss: 0.0927 - accuracy: 0.9695 - val\_loss: 0.0772 - val\_accuracy: 0.9751

<keras.callbacks.History at 0x7f71e8184070>

Saving Model

ls

vegetable-dataset/ model.save('vegetable.h5')

!tar -zcvf Train-model\_new.tgz vegetable.h5 vegetable.h5

ls -1

vegetable-dataset/ vegetable.h5 Train-model\_new.tgz

IBM Cloud Deployment Model

!pip install watson-machine-learning-client –upgrade

Collecting watson-machine-learning-client

Downloading watson\_machine\_learning\_client-1.0.391-py3-none-any.whl (538 kB)

|████████████████████████████████| 538 kB 21.2 MB/s eta 0:00:01

Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site packages

(from watson-machine-learning-client) (4.62.3)

Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site packages

(from watson-machine-learning-client) (2022.9.24)

Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site packages (from watson-machine-learning-client) (2.26.0)

Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site packages (from watson-machine-learning-client) (0.8.9)

Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python 3.9/lib/python3.9/site- packages (from watson-machine-learning-client) (2.11.0) Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site packages (from watson-machine-learning- client) (1.3.4)

Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site packages (from watson-machine-learning-client) (0.3.3)

Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site packages

(from watson-machine-learning-client) (1.18.21)

Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site packages

(from watson-machine-learning-client) (1.26.7)

Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python

3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)

Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python

3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)

Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python

3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python

3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson machine- learning-client) (2.8.2)

Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson machine-learning-client) (1.15.0)

Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python

3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)

Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python

3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0) Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python

3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (2.0.4) Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python 3.9/lib/python3.9/site- packages (from requests->watson-machine-learning-client) (3.3) Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from pandas->watson- machine-learning-client) (2021.3) Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python 3.9/lib/python3.9/site-packages (from pandas->watson-machine- learning-client) (1.19.5) Installing collected packages: watson-machine-learning-client

Successfully installed watson-machine-learning-client-1.0.391 from ibm\_watson\_machine\_learning import APIClient wml\_credentials = {

"url": "https://us-south.ml.cloud.ibm.com",

"apikey":"0P3XkyCFYqABnc48BNG2ReoGAJy-oDXDRuULl4Y\_zFxa"

} client = APIClient(wml\_credentials) def guid\_from\_space\_name(client, space\_name):

space = client.spaces.get\_details() return(next(item for item in space['resources'] if item['entity']["name"]==space\_name)['m

etadata']['id'])

space\_uid = guid\_from\_space\_name(client, 'Trainmodel') print("Space UID = " + space\_uid)

Space UID = 616c7d74-e99b-4c09-9922-27394a62c2d0 client.set.default\_space(space\_uid)

‘SUCCESS’

client.software\_specifications.list()

NAME ASSET\_ID TYPE

default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base pytorch-onnx\_1.3-py3.7-edt 069ea134-3346- 5748-b513-49120e15d288 base scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344- eb7b665ff687 base spark-mllib\_3.0-scala\_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base pytorch-onnx\_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base ai-function\_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base shiny-r3.6 0e6e79df-875e-4f24-8ae9-

62dcc2148306 base tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base pytorch\_1.1- py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base tensorflow\_1.15-py3.6-ddl 111e41b3- de2d-5422-a4d6-bf776828c4b7 base runtime-22.1-py3.9 12b83a17-24d8-5082-900f- 0ab31fbfd3cb base scikit-learn\_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base default\_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base pytorch-onnx\_1.3-py3.6 1bc6029a- cc97-56da-b8e0-39c3880dbbe7 base kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e- 474a5cdf5988 base pytorch-onnx\_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base tensorflow\_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base spark-mllib\_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base tensorflow\_2.4-py3.8-horovod 217c16f6-178f- 56bf-824a-b19f20564c49 base runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-

da66306ce658 base do\_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base autoai-ts\_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base tensorflow\_1.15-py3.6 2b73a275-7cbf-420b- a912-eae7f436e0bc base kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base pytorch\_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base spark-mllib\_2.3 2e51f700- bca0-4b0d-88dc-5c6791338875 base pytorch-onnx\_1.1-py3.6-edt 32983cea-3f32-4400-8965- dde874a8d67e base spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base spark- mllib\_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base xgboost\_0.82-py3.6 39e31acd-5f30- 41dc-ae44-60233c80306e base pytorch-onnx\_1.2-py3.6-edt 40589d0e-7019-4e28-8daa- fb03b6f4fe12 base default\_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base autoai-ts\_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base autoai-obm\_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base pmml-3.0\_4.3 493bcb95-16f1-5bc5-bee8- 81b8af80e9c7 base spark-mllib\_2.4-r\_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base xgboost\_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base pytorch-onnx\_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base autoai-ts\_3.9-py3.8 52c57136-80fa-572e-8728- a5e7cbb42cde base spark-mllib\_2.4-scala\_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base spark-mllib\_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base autoai-obm\_2.0 5c2e37fa-80b8- 5e77-840f-d912469614ee base spss-modeler\_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base autoai-kb\_3.1-py3.7 632d4b22-10aa-

5180-88f0-f52dfb6444d7 base pytorch-onnx\_1.7-py3.8 634d3cdc-b562-5bf9-a2d4- ea90a478456b base spark-mllib\_2.3-r\_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base

tensorflow\_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base spss-modeler\_18.2

687eddc9-028a-4117-b9dd-e57b36f1efa5 base

Note: Only first 50 records were displayed. To display more use 'limit' parameter. software\_space\_uid = client.software\_specifications.get\_uid\_by\_name("tensorflow\_rt22.1- py3.9") software\_spec\_uid

‘1eb25b84-d6ed-5dde-b6a5-3fbdf1665666' ls

vegetable-dataset/ vegetable.h5 Train-model\_new.tgz

model\_details = client.repository.store\_model(model= 'Train-model\_new.tgz', meta\_props={ client.repository.ModelMetaNames.NAME:"CNN", client.repository.ModelMetaNames.TYPE:"tensorflow\_2.7",

client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID:software\_space\_uid} ) model\_id = client.repository.get\_model\_id(model\_details) model\_id

'd0aeb6a2-e89c-4f8d-bf2f-a28ca4ea3cca' ls

vegetable-dataset/ vegetable.h5 Train-model\_new.tgz Test The Model import numpy as np

from tensorflow.keras.models import load\_model from tensorflow.keras.preprocessing import image model=load\_model('vegetable.h5')

#@title

img=image.load\_img(r"C:\Users\LENOVO\Desktop\vegetable-dataset\vegetable dataset\test\00fca0da-2db3-481b-b98a

9b67bb7b105c RS\_HL 7708.JPG",target\_size=(128,128)) img



img=image.load\_img(r"C:\Users\LENOVO\Desktop\ibm\Dataset Plant Disease\vegetable dataset\vegetable-dataset\test\Tomato healthy\0adc1c5b-8958-47c0-a152- f28078c214f1 RS\_HL 7825.JPG",target\_size=(128,128)) img



x=image.img\_to\_array(img) X

array([[[ 99., 86., 106.],

[101., 88., 108.],

[118., 105., 125.],

...,

[ 92., 83., 102.],

[ 93., 84., 103.],

[ 89., 80., 99.]],

[[ 96., 83., 103.],

[ 87., 74., 94.],

[102., 89., 109.],

...,

[ 88., 79., 98.],

[ 89., 80., 99.],

[ 83., 74., 93.]], [[ 86., 73., 93.],

[ 88., 75., 95.],

[ 98., 85., 105.],

...,

[107., 98., 117.],

[ 96., 87., 106.],

[ 96., 87., 106.]],

...,

[[172., 175., 194.],

[173., 176., 195.],

[175., 178., 197.],

...,

[179., 180., 198.],

[184., 185., 203.],

[179., 180., 198.]], [[172., 175., 194.],

[170., 173., 192.], [173., 176., 195.],

...,

[178., 179., 197.],

[182., 183., 201.],

[178., 179., 197.]], [[169., 172., 191.],

[166., 169., 188.],

[168., 171., 190.],

...,

[187., 188., 206.],

[185., 186., 204.],

[186., 187., 205.]]], dtype=float32) x=np.expand\_dims(x,axis=0)

X

array([[[[ 99., 86., 106.],

[101., 88., 108.],

[118., 105., 125.],

...,

[ 92., 83., 102.],

[ 93., 84., 103.],

[ 89., 80., 99.]],

[[ 96., 83., 103.],

[ 87., 74., 94.],

[102., 89., 109.],

...,

[ 88., 79., 98.],

[ 89., 80., 99.],

[ 83., 74., 93.]], [[ 86., 73., 93.],

[ 88., 75., 95.],

[ 98., 85., 105.],

...,

[107., 98., 117.],

[ 96., 87., 106.],

[ 96., 87., 106.]],

...,

[[172., 175., 194.],

[173., 176., 195.],

[175., 178., 197.],

...,

[179., 180., 198.],

[184., 185., 203.],

[179., 180., 198.]],

[[172., 175., 194.],

[170., 173., 192.],

[173., 176., 195.],

...,

[178., 179., 197.],

[182., 183., 201.],

[178., 179., 197.]],

[[169., 172., 191.],

[166., 169., 188.],

[168., 171., 190.],

...,

[187., 188., 206.],

[185., 186., 204.],

[186., 187., 205.]]]], dtype=float32)

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

1/1 [==============================] - 0s 105ms/step x\_train.class\_indices

{'Tomato Blight': 0, 'Tomato healthy': 1, 'Corn\_(maize) Northern\_Leaf\_Blight': 2,

'Corn\_(maize) healthy': 3, 'Potato Blight': 4, 'Potato healthy': 5} index=['Tomato Blight','Tomato healthy','Corn\_(maize) Northern\_Leaf\_Blight','Cor n\_(maize) healthy','Potato Blight','Potato healthy'] index[y[0]]

'Tomato healthy' img=image.load\_img(r"C:\LENOVO\Desktop\ibm\Dataset Plant Disease\vegetable dataset\vegetable-dataset\test\Potato healthy\0a2ed402-5d23-4e8d-bc98- b264aea9c3fb Rutg.\_HL 2471.JPG",target\_size=(128,128)) x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) y=np.argmax(model.predict(x),axis=1)

index=['Tomato Blight','Tomato healthy''Potato Blight','Potato healthy'] index[y[0]]

1/1 [==============================] - 0s 26ms/step

'Potato healthy' import os

from tensorflow.keras.models import load\_model from tensorflow.keras.preprocessing import image from flask import Flask,render\_template,request app=Flask( name )

model=load\_model("vegetable.h5")

@app.route('/') def index(): return render\_template("index.html") @app.route('/predict',methods=['GET','POST']) def upload(): if request.method=='POST':

f=request.files['image'] basepath=os.path.dirname(' file ')

filepath=os.path.join(basepath,'uploads',f.filename) f.save(filepath) img=image.load\_img(filepath,target\_size=(128,128)) x=image.img\_to\_array(img) x=np.expand\_dims(x,axis=0) pred=np.argmax(model.predict(x),axis=1) index=['Tomato Blight','Tomato healthy', ,'Potato Blight','Potato healthy'] text="The Classified Vegetable disease is : " +str(index[pred[0]]) return text

if name ==' main ': app.run(debug=False) ibmapp.py

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 app = Flask(\_name\_)

#load both the vegetable and fruit models model = load\_model("IBM-vegetable.h5") model1=load\_model("IBM-fruit.h5")

#home page @app.route('/') def home(): return render\_template('home.html')

#prediction page @app.route('/prediction') def prediction():

return render\_template('predict.html') @app.route('/predict',methods=['POST']) def predict():

if request.method == 'POST': # Get the file from post request f = request.files['image']

# Save the file to ./uploads basepath = os.path.dirname(\_file\_) file\_path = os.path.join(

basepath, 'uploads', secure\_filename(f.filename)) f.save(file\_path) img = image.load\_img(file\_path, target\_size=(128, 128)) x = image.img\_to\_array(img) x = np.expand\_dims(x, axis=0)

plant=request.form['plant'] print(plant) if(plant=="vegetable"): preds = model.predict(x) preds=np.argmax(preds)

print(preds) df=pd.read\_excel('precautions - veg.xlsx') print(df.iloc[preds]['caution']) else:

preds = model1.predict(x) preds=np.argmax(preds)

df=pd.read\_excel('precautions - fruits.xlsx') print(df.iloc[preds]['caution'] return df.iloc[preds]['caution']

if \_name\_ == "\_main\_": app.run(debug=False)

**GitHub Link:**

**https://github.com/IBM-EPBL/IBM-Project-184531659685411**

**Project Demo Link**

**https://drive.google.com/file/d/1wZ2\_sxbCYNoV3Qs**

**XTqyCd7r57k99V2Eu/view?usp=sharing**