# Fertilizers Recommendation System for Disease Prediction

## Professional Readiness for Innovation, Employability and Entrepreneurship

## Team ID: PNT2022TMID13066



| | |
|---|---|
| **DEVA PRIYAN T A** | **(19I211)** |
| **KARTHIKEYAN A** | **(19I227)** |
| **PRITHIK KUMAR R** | **(19I243)** |
| **RUBASHREE R** | **(19I250)** |



Dissertation submitted in partial fulfillment of the requirements for the degree of

**BACHELOR OF TECHNOLOGY**

**Branch: INFORMATION TECHNOLOGY**

of Anna University

**NOVEMBER 2022**


**DEPARTMENT OF INFORMATION TECHNOLOGY**

**PSG COLLEGE OF TECHNOLOGY**

(Autonomous Institution)

**COIMBATORE – 641 004**

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude to **Dr K Prakasan**, Principal, PSG College of Technology, for giving us the opportunity to do our project with various facilities and infrastructure, without which the success of the project would not have been possible.

We extend our heartfelt thanks to **Dr K Umamaheswari**, Professor and Head of the Department of Information Technology, PSG College of Technology, for her unfailing support throughout this project.

We express our sincere thanks to our Programme Coordinator **Dr D Karthika Renuka,** Professor, Department of Information Technology, PSG College of Technology, whose constant support and everlasting enthusiasm made it possible to have completed within the time.

**Commented [SP1]:** Change to, "Dr D.Karthika Renuka, Professor"

We heartfully thank our guide **Dr S Sangeetha**, Professor, Department of Information Technology, who was always there to help us and played a major role in the completion of the project and we wish to thank her for her enduring guidance and priceless advice throughout this project work.

**Commented [SP2]:** Change to, "Ms P.Shruti"

We also wish to express our sincere thanks to our tutor **Dr S Sangeetha,** Assistant Professor, Department of Information Technology, for her guidance that played a vital role in completing my project on time.

Finally, we would like to thank God, Elders, siblings, my faculty members, lab technicians and friends without whom this project work would not have been completed successfully.

**Commented [SP3]:** Change the spacing for this paragraph

# SYNOPSIS

Food is the primary source of energy for any living organisms in the world. Without food, there will be no living organisms. Humans prepare their own food by cultivating the land and those cultivators are called the Farmers. In today's world, farmers are facing lots of challenges and troubles to successfully harvest the crops. One of the biggest obstacle they face is the infection by the disease and most importantly their spread. The early diagnosis and prediction can save lots of crop in most cases. Most of the farmers are unaware of the proper knowledge about those disease. These problems can be overcome by publishing a model that can able to diagnosis the disease of the plant and also be able to recommend a proper remedial through fertilizers for the plants. This will help in greatly enhancing both the quantity and quality of the crops. The model should be in the form of easily accessible web based application and should be easily understandable for the farmers. The model should able to observe the symptoms of the leaf through the image that is uploaded by the farmer and should be able to predict the disease and display it. In addition, the model should recommend the fertilizer for that particular disease. This all can be achieved through the use of Deep Learning Techniques. The image classification can be performed by the CNN deep learning algorithm and using proper deep learning strategies, the disease predicted can be categorized and then the fertilizer can be recommended from the database.

# LIST OF FIGURES

Commented [SP4]: Remove extra space above

# CONTENTS

**CHAPTER** **Page No.**



**Commented [SP5]:** Align all the page numbers in a straight line

**Commented [SP6]:** Align the headings and sub headings properly

Contents

# CHAPTER 1

# INTRODUCTION

### 1.1 PROJECT OVERVIEW:

In this project, two datasets the fruit dataset and the vegetable dataset were collected. Convolutional Neural Networks, a deep learning neural network, is used to train and test the datasets that have been collected (CNN). First, With CNN, the fruit dataset is first trained and then tested. There are 6 courses total, and each class is trained and tested. The vegetable dataset is then tested and trained. Python is the programming language used to train and test datasets. All of the Python code is initially created in the Jupyter notebook that comes with Anaconda Python, and it is then tested in the IBM cloud. Finally, Flask, a Python package, is used to construct a web-based framework. Along with their related files, two html files are created in the templates folder.

### 1.2 PURPOSE:

This study is used to test samples of fruits and vegetables and find out which diseases they may have. The farmers would benefit from an early and simple comprehension of the disease that has contaminated their crop thanks to this feature. Additionally, this model suggests the proper fertilizers for the diseases that are predicted, assisting farmers in preventing significant yield losses.

# CHAPTER 2

# LITERATURE SURVEY

Commented [SP7]: Header for this page is incorrect

### 2.1 EXISTING PROBLEM:

Existing issue Indumathi et al proposed a technique for spotting leaf illnesses and suggested fertilizers to treat them. However, the method's low number of train and test sets leads to subpar accuracy. Pandi Selvi suggested that simple crop disease prediction approach for soil-based fertilizer recommendation system. This approach offers less predictability and accuracy. Shiva Reddy proposed a machine learning based IoT based system for recommending fertilizer and detecting leaf disease that has less than 80% accuracy.

**Literature Survey:**

| S.No | Title | Proposed System | Advantages | Disadvantages |
|---|---|---|---|---|
| 1. | Semi-automatic leaf disease detection and classification system for soybean culture IET Image Processing, 2018 | The suggested approach employs SVM to categorise tree leaves, pinpoint the disease, and provide fertiliser. The suggested approach is contrasted with the currently available CNN-based leaf disease prediction. When compared to current CNN methods, the suggested SVM technique produces better results. | The prediction and diagnosing of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves. | The proposed algorithm is being implemented in this new study using openly available datasets. Additionally, different segmentation methods might be used to increase accuracy. To detect diseases that affect other plant organs, such |

| | | | | |
|---|---|---|---|---|
| | | The accuracy of identifying leaf illness using CNN is 0.6 and SVM is 0.8 for the same set of photos. F-Measure for CNN is 0.7 and 0.8 for SVM. | | stems and fruits, the proposed method might be further developed. |
| 2. | Shloka Gupta, Nishit Jain, Akshay Chopade, Farmer’s Assistant: A Machine Learning Based Application for Agricultural Solutions | In this study, we present the "Farmer's Assistant," a user-friendly online application system built on machine learning and web scraping.<br>We are able to offer numerous functions with our system, including crop recommendation using the Random Forest algorithm, fertiliser advice using a rule-based categorization method, and crop disease detection. utilising the EfficientNet model on photos of leaves. The user can input data using forms on our user interface and receive responses immediately.<br>Additionally, we employ the LIME interpretability | Regarding fertiliser and crop recommendations, We can let users know that the products are available on well-known shopping websites and perhaps even let them purchase crops and fertiliser right from our app. | To identify fine-grained segmentations of the dataset's sick area.<br>The absence of such data makes this impractical.<br>However, we may incorporate a segmentation annotation tool within the application so that users may be able to fill in the gaps.<br>Additionally, unsupervised methods may be employed to identify the image's sick regions. |

| | | | | |
|---|---|---|---|---|
| | | approach to explain our predictions on the disease detection image, which may help explain why our model makes the predictions it does and allow us to use this understanding to enhance datasets and models. | | |
| 3. | Cloud Based Automated Irrigation And Plant Leaf Disease Detection System Using An Android Application. | Detection of Leaf Diseases and Classification using Digital Image Processing International Conference on Innovations in Information, Embedded and Communication Systems(ICIIECS), IEEE, 2017. | The system detects the diseases on citrus leaves with 90% accuracy. | System only able to detect the disease from citrus leaves. |
| 4. | Swapnil Jori1, Rutuja Bhalshankar2, Dipali Dhamale3, Sulochana Sonkamble , Healthy Farm: Leaf Disease Estimation and Fertilizer Recommendation System using Machine Learning. | In the current study, image processing techniques for spotting plant diseases in various plant species are examined and described. The most popular techniques for identifying plant diseases include BPNN, SVM, K-means clustering, and SGDM. | | These methods have problems, some of which include the effect of background information on the final result, refinement of a methodology for a particular plant leaf disease, and automation of a system for ongoing, automatic |

| | | | | |
|---|---|---|---|---|
| | | | | monitoring of plant leaf diseases in actual field settings. |
| 5. | Ms. Kiran R. Gavhale, Ujwalla Gawande, Plant Leaves Disease detection using Image Processing Techniques, January 2014. | Semi-automatic leaf disease detection and classification system for soybean culture IET Image Processing, 2018 | The system helps to compute the disease severity | The system cannot be implemented in real time since it needs leaf photos from an online dataset. |
| 6. | R. Neela, P. Fertilizers Recommendation System For Disease Prediction In Tree Leave International journal of scientific & technology research volume 8, issue 11, november 2019 | The author suggests a strategy that, by recommending the best crops, aids in agricultural production prediction. In order to determine what crop should be put in the field to enhance productivity, it also focuses on soil types. Soil types are crucial for crop yield. Information about the soil can be acquired by factoring in the weather information from the previous year. | It enables us to foresee which crops might thrive in a specific climate. Crop quality can also be increased using data sets relating to weather and disease. We can categorise the data using prediction algorithms according to the disease, and we can predict soil and crops using the data that was taken from the classifier. | Accurate results cannot be predicted because of the varying climatic circumstances through this method. |
| 7. | Duan Yan-e, Design of Intelligent Agriculture Management Information | Cloud Based Automated Irrigation And Plant Leaf Disease Detection System Using An Android | It is simple and cost effective system for plant leaf disease detection | The performance of the system may be impacted by |

| | | | | |
|---|---|---|---|---|
| | System Based on IOT. | Application. International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017 | | any hardware issues. With the help of the cloud and IoT, the current paper suggests an Android application for plant disease diagnosis and watering. They use soil moisture and temperature sensors, and the sensor data is sent to the cloud, for the purpose of monitoring irrigation systems. Additionally, the user can identify plant leaf disease. K-means clustering is used to extract features. |
| 8. | Soil Based Fertilizer Recommendation System for Crop Disease Prediction System. | The proposed system was designed to examine the soil type, identify diseases in the leaves, and then advise the farmers on the fertiliser that would be most helpful to them. One of the main | By balancing crop production, yielding the proper crop at the right time reducing crop scarcity through economic expansion, plant disease control, and planning. Therefore, it is vital | It cannot be extended to incorporate several cultivable crop kinds and performance analysis. |

| | | | | |
|---|---|---|---|---|
| | | causes of lower yields in both quality and quantity is plant disease, particularly on the leaves. as well as the size of the food crops. Finding the leaf disease is a crucial part of keeping agriculture alive. In agriculture, clever analysis and thorough prediction models assist the farmer in producing the right crop at the right time. | to provide symptoms in order to identify the disease in its early stages in order to detect and diagnose plant diseases and to propose fertiliser. | |

## 2.2 REFERENCES:

[1] Kaur, S., Pandey, S., & Goel, S. (2018). Semi-automatic leaf disease detection and classification system for soybean culture. *IET Image Processing*, *12*(6), 1038-1048.

[2] Gupta, S., Chopade, A., Jain, N., & Bhonde, A. (2022). Farmer's Assistant: A Machine Learning Based Application for Agricultural Solutions. *arXiv preprint arXiv:2204.11340*.

[3] Anas, S., Badhusha, I., Zaheema, O. T., Faseela, K., & Shelly, M. (2017, April). Cloud based automated irrigation and plant leaf disease detection system using an android application. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)* (Vol. 2, pp. 211-214). IEEE.

[4] https://github.com/IBM-EPBL/IBM-Project-17508-1659672731.
[5] Gavhale, K.R., & Gawande, U.H. (2014). An Overview of the Research on Plant Leaves Disease detection using Image Processing Techniques. *IOSR Journal of Computer Engineering, 16*, 10-16.

[6] Neela, R., & Nithya, P. (2019). Fertilizers Recommendation System For Disease Prediction In Tree Leave.

[7] D. Yan-e, "Design of Intelligent Agriculture Management Information System Based on IoT," 2011 Fourth International Conference on Intelligent Computation Technology and Automation, 2011, pp. 1045-1049, doi: 10.1109/ICICTA.2011.262.

[8] Selvi, D.P., & Poornima, P. (2021). Soil Based Fertilizer Recommendation System for Crop Disease Prediction System.

**2.3 PROBLEM STATEMENT DEFINITION:**

In today's society, agriculture is the most significant industry. An extensive range of bacterial and fungal diseases harm the majority of plants. Plant diseases severely limited productivity and posed a serious threat to food security. To achieve maximum quantity and optimum quality, early and accurate identification of plant diseases is crucial. The variety of pathogen strains, adjustments to production practices, and insufficient plant protection systems have all contributed to an increase in the number of plant diseases in recent years, as well as the severity of the damage they inflict. An automated technique is now available to recognize many plant diseases by examining the symptoms seen on the plant's leaves. Deep learning algorithms are used to diagnose diseases and provide preventative measures that can save great loss in fields.

# CHAPTER 3

# IDEATION & PROPOSED SOLUTION

### 3.1 EMPATHY MAP CANVAS:


**Fig 3.1 Empathy map**

This Figure 3.1 shows a empathy map that will gives a collaborative visualization for the end user, what they can perform using this project, what challenges have been faced and also what is the real time use for this application.

### 3.2 IDEATION & BRAINSTORMING:

**Step 1:**

Brainstorming phase, the ideas from every group members are gathered.


**Fig 3.2 Brainstorm**

**Step 2:**

Grouping the ideas under the suitable topics for better understanding.

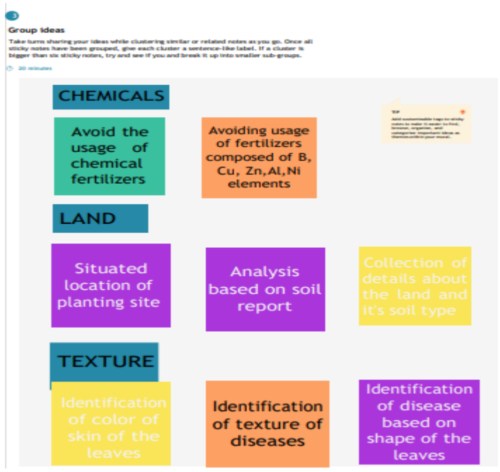



**Fig 3.3 Grouping ideas**

**Step 3:**

Prioritizing the ideas or the features and performing the feasibility study on it.

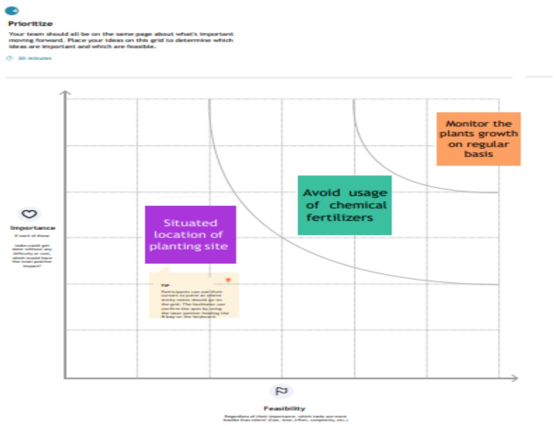



**Fig 3.4 Prioritizing ideas**

### 3.3 PROPOSED SOLUTION:

**Problem Statement:** To identify plant illnesses at an early stage that are brought on by various pests and microbes and to suggest fertilizers to prevent crop destruction.

**Solution Description:** Dividing plant diseases into groups based on the pest. Using the CNN method, the dataset and photos of the plant diseases are compared to pre-existing images.

**Novelty:** Deep learning models are used to automatically detect plant illnesses.

**Social Impact:** It is helpful for farmers to identify certain plant illnesses at an early stage and take action to treat the plants accordingly.

**Scalability of the Solution:** Almost all farmers worldwide should have access to web applications. Web applications have to be inexpensive. The model may predict new diseases based on the prior dataset.

**3.4 PROPOSED SOLUTION FIT:**

The proposed solution should fit into various constraints and should fulfill the demands of various categories of people.

**Customer Segment:** Farmers Are This Application's First Users. This application is simple to use and provides suggestions for applying fertilizer properly.

**Triggers:** Farmers observing their crops are being attacked by disease that can bring in terms of both quantity and quality.

**Emotion:** Boosting the self-confidence for the farmers and bringing a support for them in terms of technology.

**Customer constraints:** Good Networks and good image capturing device are the most important constraints that are required from the customer's side.

**Direct Behavior:** Farmers don't require any additional knowledge of the disease.

**Problem Root Cause:** Various disease on the plants can lead to reducing the quality and quantity of the crops. Also it can spread to other nearby fields.

**Solution:** By predicting the disease and then suggesting the correct fertilizer for it can help the farmer to boost their yields.

# CHAPTER 4

# REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENT:

Following are the functional requirements of the proposed solution .

| Fr.no | Functional requirement | Sub requirement (story/subtask) |
|---|---|---|
| Fr-1 | User registration | Registration through form Registration through Gmail |
| Fr-2 | User confirmation | Confirmation via OTP Confirmation via Email |
| Fr-3 | Capturing image | Capture the image of the leaf And check the parameter of thecaptured image . |
| Fr-4 | Image processing | Upload the image for the prediction of the disease in the leaf. |
| Fr-5 | Leaf identification | Identify the leaf and predict the disease in leaf. |
| Fr-6 | Image description | Suggesting the best fertilizer forthe disease . |

**Fig 4.1 Functional Requirement**

## 4.2 NON-FUNCTIONAL REQUIREMENT:

Following are the non-functional requirement of the proposed solution

| NFr.no | Non-functional requirement | Description |
|---|---|---|
| Nfr-1 | Usability | Datasets of all the leaf is used to detecting the disease that present in the leaf. |

| | | |
|---|---|---|
| Nfr-2 | Security | The information belongs to the user and leaf are securedhighly. |
| Nfr-3 | Reliability | The leaf quality is importantfor the predicting the disease in leaf. |
| Nfr-4 | Performance | The performance is based onthe quality of the leaf used for disease prediction |
| Nfr-5 | Availability | It is available for all user to predict the disease in the plant |
| Nfr-6 | Scalability | Increasing the prediction of the disease in the leaf |

**Fig 4.2 Non-Functional Requirement**

# CHAPTER 5
# PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAM:

The below figure 5.1 clearly depicts the entire flow of the model i.e starting from getting the image as input, followed by preprocessing and then followed by disease detection and then finally suggesting the fertilizer for the disease.

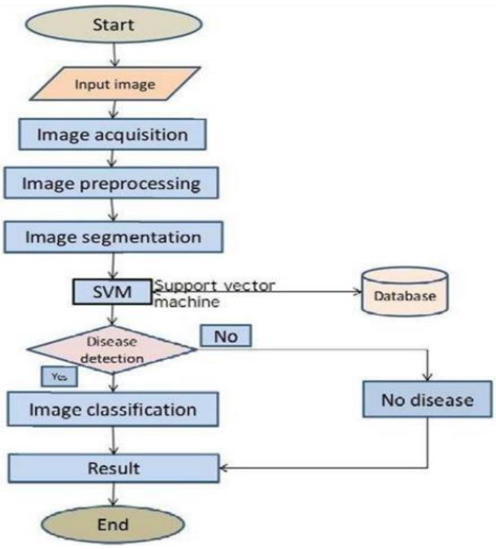



**Fig 5.1 Data Flow Diagram**

### 5.2 SOLUTION & TECHNICAL ARCHITECTURE:

The below figure 5.2 depicts the architecture for the proposed solution. The user can upload the image of the crop sample to the model. The input model will use CNN model and deep learning techniques to predict whether the sample is infected by disease or not.

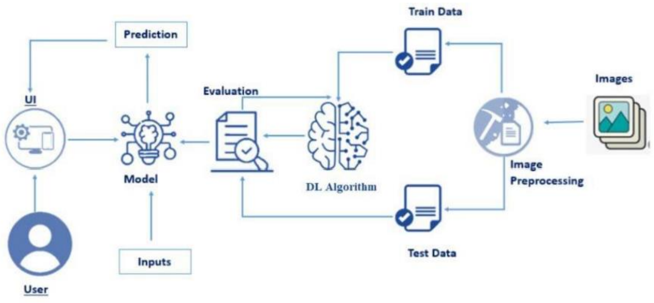



**Fig 5.2 Solution Architecture**

# CHAPTER 6

# PROJECT PLANNING & SCHEDULING

## 6.1 SPRINT PLANNING & ESTIMATION:

The below figure 6.1 depicts the sprint plan for the various functional and non-functional requirements with the assigned priority.

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points (Total) | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Model Creation and Training (Fruits) | | Create a model which can classify diseased fruit plants from given images. I also need to test the model and deploy it on IBM Cloud | 8 | High | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Model Creation and Training (Vegetables) | | Create a model which can classify diseased vegetable plants from given images | 2 | High | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points (Total) | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-2 | Model Creation and Training (Vegetables) | | Create a model which can classify diseased vegetable plants from given images and train on IBM Cloud | 6 | High | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Registration | USN-1 | As a user, I can register by entering my email, password, and confirming my password or via OAuth API | 3 | Medium | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Upload page | USN-2 | As a user, I will be redirected to a page where I can upload my pictures of crops | 4 | High | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Suggestion results | USN-3 | As a user, I can view the results and then obtain the suggestions provided by the ML model | 4 | High | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Base Flask App | | A base Flask web app must be created as an interface for the ML model | 2 | High | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| Sprint-3 | Login | USN-4 | As a user/admin/shopkeeper, I can log into the application by entering email & password | 2 | High | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | User Dashboard | USN-5 | As a user, I can view the previous results and history | 3 | Medium | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Integration | | Integrate Flask, CNN model with Cloudant DB | 5 | Medium | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Containerization | | Containerize Flask app using Docker | 2 | Low | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint-4 | Dashboard (Admin) | USN-6 | As an admin, I can view other user details and uploads for other purposes | 2 | Medium | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Dashboard (Shopkeeper) | USN-7 | As a shopkeeper, I can enter fertilizer products and then update the details if any | 2 | Low | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |
| | Containerization | | Create and deploy Helm charts using Docker Image made before | 2 | Low | Deva Priyan T A, Karthikeyan A, Prithik Kumar R, Rubashree R |

**Fig 6.1 Sprint Planning**

## 6.2 SPRINT DELIVERY SCHEDULE:

The below figure 6.2 shows the Sprint Delivery Schedule for a duration of 6 days and with their duration.

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

**Fig 6.2 Sprint Delivery Schedule**

## 6.3 REPORTS FROM JIRA:

The below figure 6.3 shows the work management board created on JIRA.

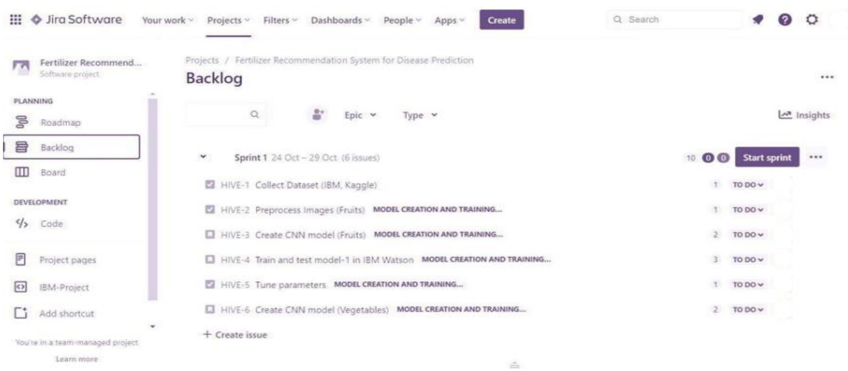



**Fig 6.3 JIRA management board**

# CHAPTER 7

# CODING & SOLUTIONING

## 7.1 FEATURE 1:

The application's registration page is created. User registration is carried out if the user hasn't already done so. Enough work was put into making this process seamless. If the user has registered, he can now log in directly. Email address, name, and password were required for registration. The code to link it to the backend was successful, and this data is stored in Firebase.

## 7.2 FEATURE 2:

The trained machine learning model can predict the output from an image that is uploaded, and the nutrition facts are also displayed on the same page. The model's accuracy was determined to be 95%, and when it was trained on the IBM cloud, it reached 100%.

## 7.3 DATABASE SCHEMA:

The Firebase platform was used. A mechanism for storing and retrieving data that is modelled in ways other than the tabular relations used in relational databases is provided by the Firebase database (NoSQL).

# CHAPTER 8

# TESTING

## 8.1 TEST CASES:

The test cases include invalid email and unrecognizable images. For the image part, a text file or other format files were uploaded as a corner case.

## 8.2 USER ACCEPTANCE TESTING:

10 users of the test application were able to discover the nutritional data for the fruit image they supplied.

## 8.3 INTEGRATION TESTING:

This combined and tested both the registration and prediction modules, which showed to provide accurate results.

# CHAPTER 9

# RESULTS

## 9.1 PERFORMANCE METRICS:

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

```
pred = model.predict_classes(x)
```

```
pred
```

```
[1]
```

**Fig 9.1 Prediction Class**

The above figure 9.1 shows the prediction performance of our model. In this, the predicted class is ‘1’. Similarly, it can predict various categorizes of the diseases.

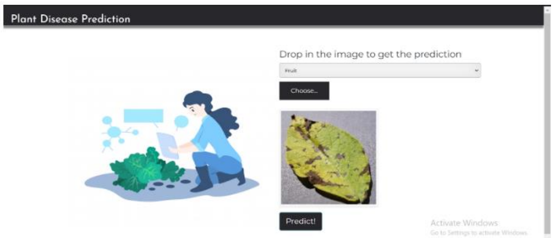



**Fig 9.2 Prediction web page**

The above figure 9.2 shows the prediction web page where the user can upload the image sample for the prediction procedure.

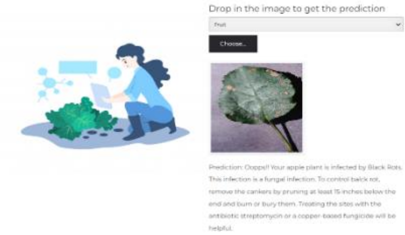



**Fig 9.3 Disease display**

The above figure 9.3 shows the result of the predicted image , it shows the result of the disease displayed in the web application.

# CHAPTER 10

# ADVANTAGES & DISADVANTAGES

## 10.1 ADVANTAGES:

- The suggested model yields extremely high classification accuracy
- It can train and test on very large datasets.
- It can resize very high-quality images within itself.

## 10.2 DISADVANTAGES:

- The proposed model is computationally expensive to train and test.
- The neural network architecture used in this project work is highly complex.

# CHAPTER 11

# CONCLUSIONS

### 11.1 CONCLUSIONS:

The model here involves classifying images from datasets of fruits and vegetables. The number of epochs was increased to boost categorization accuracy. Different classification accuracies are obtained for different batch sizes. The accuracies are increased by adding more convolution layers. The accuracy of classification is also increased by adjusting the number of dense layers. The accuracies are different while varying the size of the train and test datasets.

# CHAPTER 12

# FUTURE SCOPE

## 12.1 FUTURE SCOPE:

The model that is being provided in this project work can be expanded to recognise images. Using python to exe software, the complete model may be turned into application software. With the aid of the OpenCV Python package, real-time image categorization, picture recognition, and video processing are all made feasible. This project's work can be expanded to include security applications including face, iris, and figure print recognition.

# CHAPTER 13

# APPENDIX

**SOURCE CODE:**

index.html

E: > College > Sem_7 > IBM > Github_Submission > Final_Deliverables > Flask > templates > index.html > html > head > style > body

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Plant Disease Prediction and Fertilizer Recommendation</title>
    <link
      href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
      rel="stylesheet"
    />
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link
      href="{{ url_for('static', filename='css/main.css') }}"
      rel="stylesheet"
    />
    <style>
      .bg-dark {
        background-color: #333 !important;
      }
      #result {
        color: #0a1c4ed1;
      }
      body {
        background-color: #c2c5a8;
      }
    </style>
  </head>

  <body>
    <nav class="navbar navbar-dark bg-dark">
      <div class="container">
        <a class="navbar-brand" href="#">Plant Disease Prediction</a>
      </div>
    </nav>
    <div class="container">
      <div id="content" style="margin-top: 2em">
        <div class="container">
```

```
<div class="row">
  <div class="col-sm-6 bd">
    <h3>
      Detect if your plant <br />
      is infected!!
    </h3>
    <br />
    <p>
      Agriculture is one of the major sectors works wide.Over the
      years it has developed and the use of new technologies and
      equipment replaced almost all the traditional methods of
      farming.The plant diseases effect the production.Identification
      of diseases and taking necessary precautions is all done through
      naked eye,which requires labour and laboratries.This application
      helps farmers in detecting the diseases by observing the spots
      on the leaves ,which inturn saves effort and labor costs.
    </p>
    <img
      src="/static/plant-inspection14.jpg"
      height="30%"
      ,width="30%"
      style="border-radius: 50%; margin-right: 20px"
    />
  </div>
  <div class="col-sm-6">
    <div>
      <h4>Upload Image Here To Identify the Plant disease</h4>
      <form
        action="http://localhost:5000/"
        id="upload-file"
        method="post"
        enctype="multipart/form-data"
      >
        <label for="imageUpload" class="upload-label">
          Choose...
        </label>
        <input
          type="file"
          name="image"
```

main.js

E: > College > Sem_7 > IBM > Github_Submission > Final_Deliverables > Flask > static > js > main.js

```javascript
$(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,
```

main.js ×

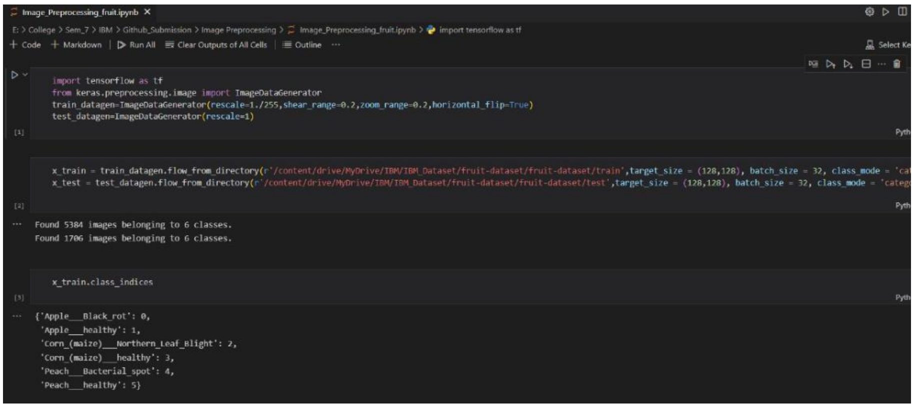
E: > College > Sem_7 > IBM > Github_Submission > Final_Deliverables > Flask > static > js > JS main.js > ...

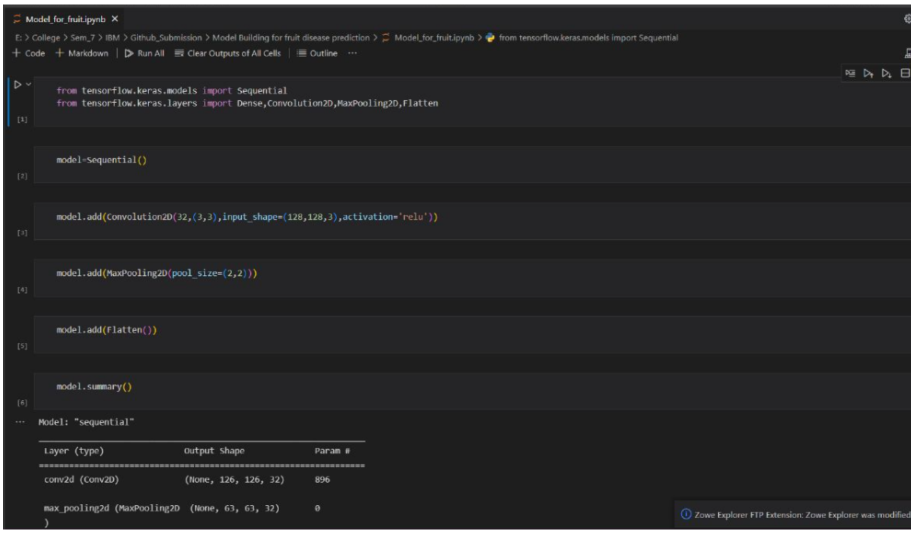
```
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text(' Result:  ' + data);
                console.log('Success!');
            },
        });
    });

});
```

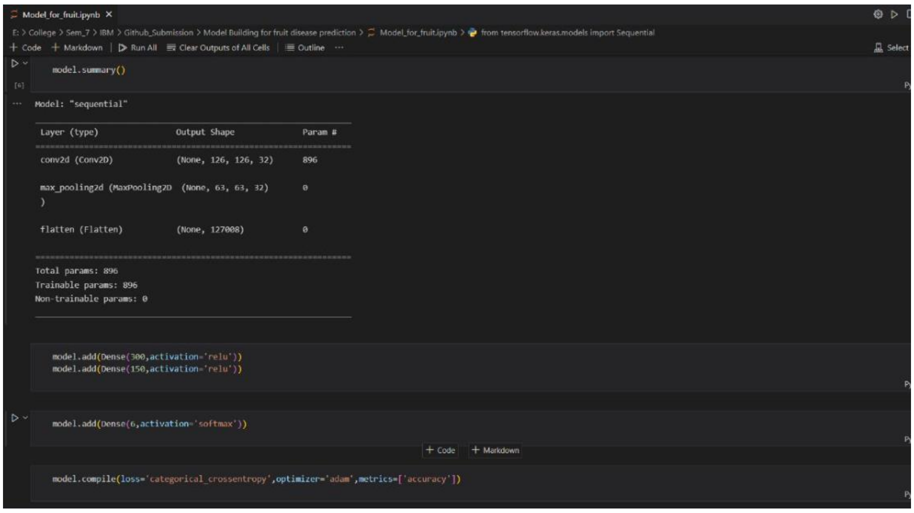
app.py ×

E: > College > Sem_7 > IBM > Github_Submission > Final_Deliverables > Flask > app.py

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask,render_template,request

app=Flask(__name__)

model=load_model("fruit.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=['Apple___Black_rot','Apple___healthy','Corn_(maize)___healthy','Corn_(maize)___Northern_Leaf_Blight','Peach___Bacterial_spot','Peach___healthy']
        if(pred[0]==0):
            text="The Classified Plant Disease is : " +str(index[pred[0]] + ". Fertilizers recommended are Captan and fungicides containing a strobulurin (FRAC Group 11 F
        elif(pred[0]==3):
            text="The Classified Plant Disease is : " +str(index[pred[0]] + ". Fertilizers recommended are Bio-fungicides based on Trichoderma harzianum, or Bacillus subt
        elif(pred[0]==4):
            text="The Classified Plant Disease is : " +str(index[pred[0]] + ". Fertilizers recommended are Copper-based sprays alone or together with an antibiotic can be
        else:
            text="The Classified Plant is : " +str(index[pred[0]] + ". And the plant is healthy.")
    return text

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


Image_Preprocessing_fruit.ipynb
E: > College > Sem_7 > IBM > Github_Submission > Image Preprocessing > Image_Preprocessing_fruit.ipynb > import tensorflow as tf
Code
Markdown
Run All
Clear Outputs of All Cells
Outline
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)
[1]
x_train = train_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/IBM_Dataset/fruit-dataset/fruit-dataset/train',target_size = (128,128), batch_size = 32, class_mode = 'ca
x_test = test_datagen.flow_from_directory(r'/content/drive/MyDrive/IBM/IBM_Dataset/fruit-dataset/fruit-dataset/test',target_size = (128,128), batch_size = 32, class_mode = 'categ
[2]
Found 5384 images belonging to 6 classes.
Found 1706 images belonging to 6 classes.
x_train.class_indices
[3]
{'Apple___Black_rot': 0,
'Apple___healthy': 1,
'Corn_(maize)___Northern_Leaf_Blight': 2,
'Corn_(maize)___healthy': 3,
'Peach___Bacterial_spot': 4,
'Peach___healthy': 5}


Model_for_fruit.ipynb
E: > College > Sem_7 > IBM > Github_Submission > Model Building for fruit disease prediction > Model_for_fruit.ipynb > from tensorflow.keras.models import Sequential
Code
Markdown
Run All
Clear Outputs of All Cells
Outline
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
[1]
model=Sequential()
[2]
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
[3]
model.add(MaxPooling2D(pool_size=(2,2)))
[4]
model.add(Flatten())
[5]
model.summary()
[6]
Model: "sequential"
Layer (type) Output Shape Param #
conv2d (Conv2D) (None, 126, 126, 32) 896
max_pooling2d (MaxPooling2D ) (None, 63, 63, 32) 0
Zowe Explorer FTP Extension: Zowe Explorer was modified



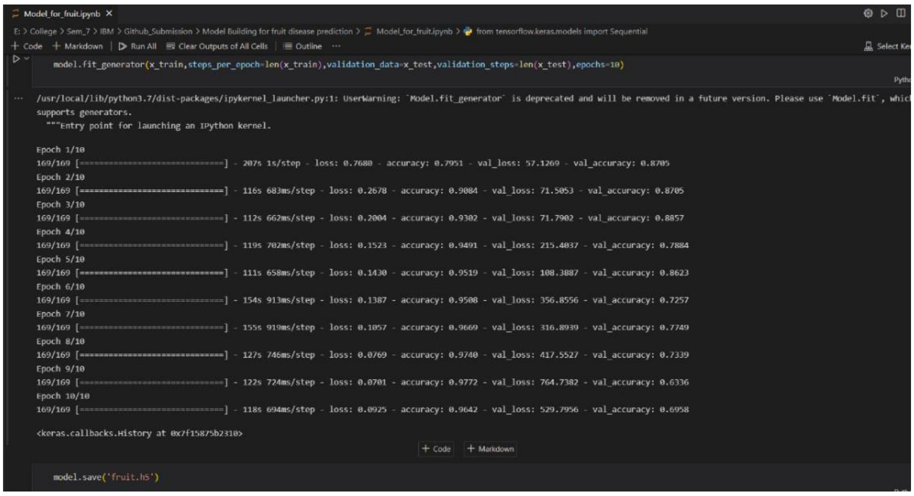
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

```
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
```

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

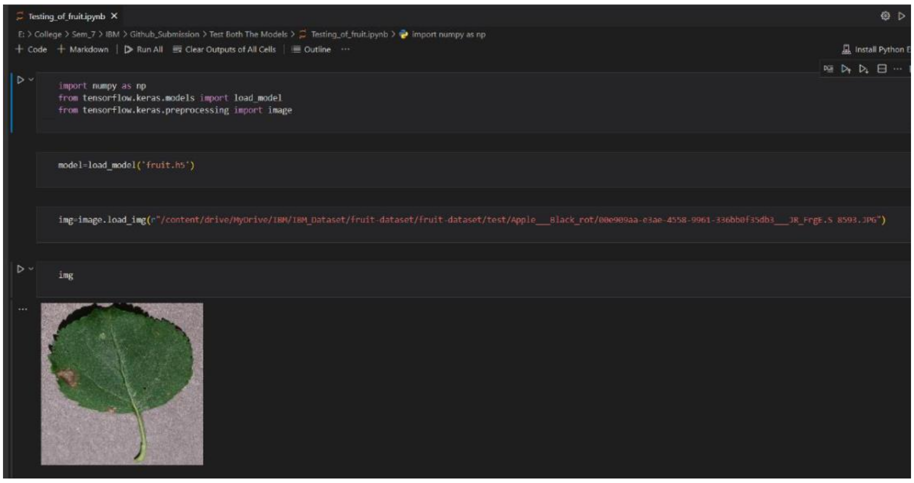
```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

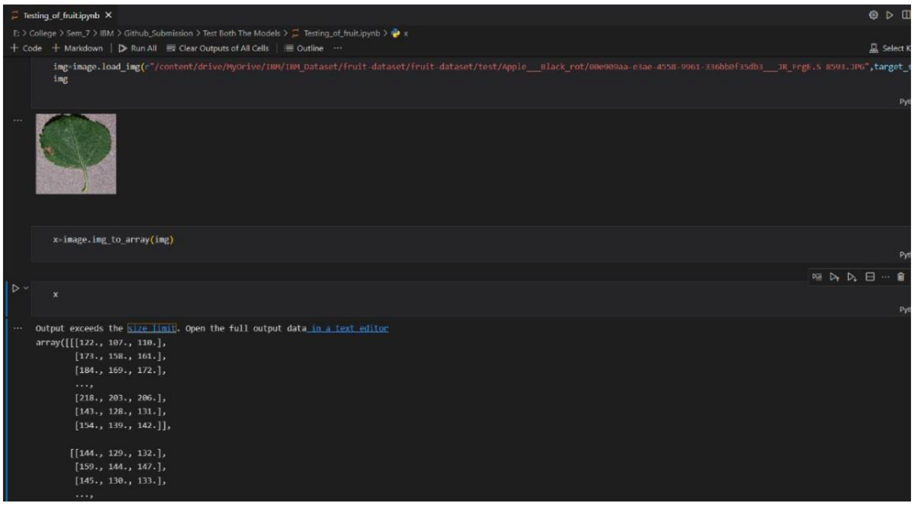


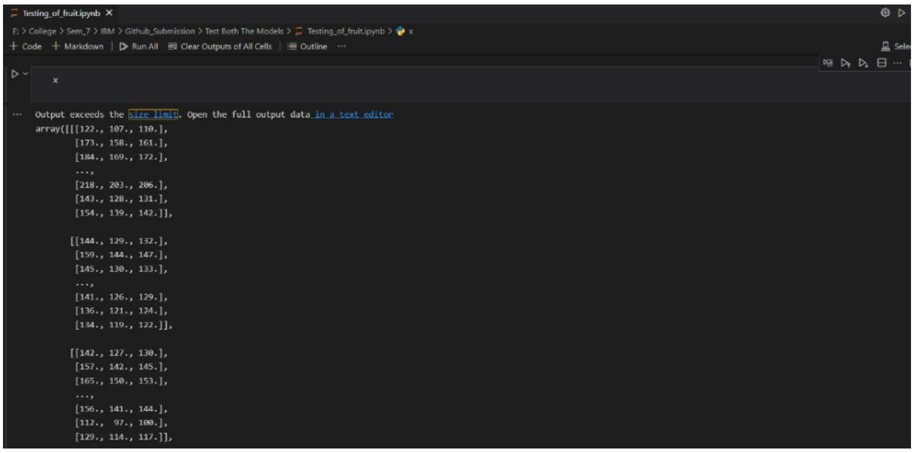
```
model.fit_generator(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=10)
```

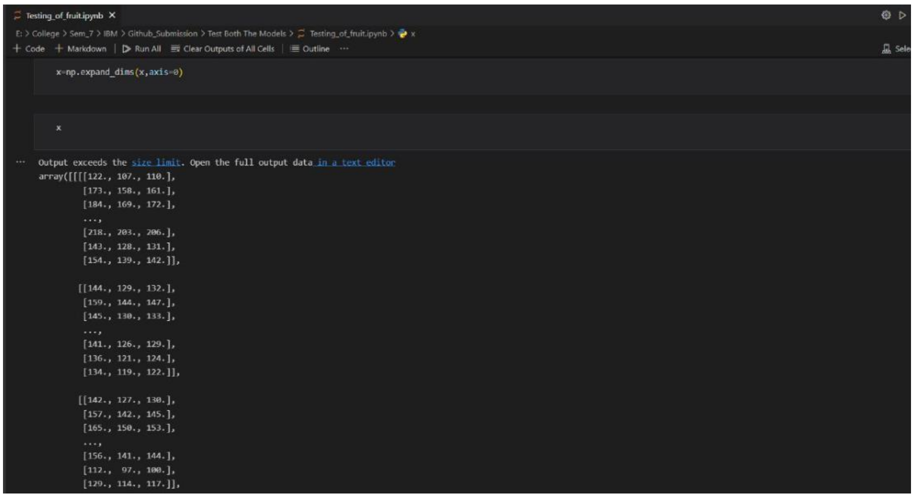
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, whic
supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/10
169/169 [==============================] - 207s 1s/step - loss: 0.7680 - accuracy: 0.7951 - val_loss: 57.1269 - val_accuracy: 0.8705
Epoch 2/10
169/169 [==============================] - 116s 683ms/step - loss: 0.2678 - accuracy: 0.9084 - val_loss: 71.5053 - val_accuracy: 0.8705
Epoch 3/10
169/169 [==============================] - 112s 662ms/step - loss: 0.2004 - accuracy: 0.9302 - val_loss: 71.7902 - val_accuracy: 0.8857
Epoch 4/10
169/169 [==============================] - 119s 702ms/step - loss: 0.1523 - accuracy: 0.9491 - val_loss: 215.4037 - val_accuracy: 0.7884
Epoch 5/10
169/169 [==============================] - 111s 658ms/step - loss: 0.1430 - accuracy: 0.9519 - val_loss: 108.3887 - val_accuracy: 0.8623
Epoch 6/10
169/169 [==============================] - 154s 913ms/step - loss: 0.1387 - accuracy: 0.9508 - val_loss: 356.8556 - val_accuracy: 0.7257
Epoch 7/10
169/169 [==============================] - 155s 919ms/step - loss: 0.1057 - accuracy: 0.9669 - val_loss: 316.8939 - val_accuracy: 0.7749
Epoch 8/10
169/169 [==============================] - 127s 746ms/step - loss: 0.0769 - accuracy: 0.9740 - val_loss: 417.5527 - val_accuracy: 0.7339
Epoch 9/10
169/169 [==============================] - 122s 724ms/step - loss: 0.0701 - accuracy: 0.9772 - val_loss: 764.7382 - val_accuracy: 0.6336
Epoch 10/10
169/169 [==============================] - 118s 694ms/step - loss: 0.0925 - accuracy: 0.9642 - val_loss: 529.7956 - val_accuracy: 0.6958
<keras.callbacks.History at 0x7f15875b2310>
```

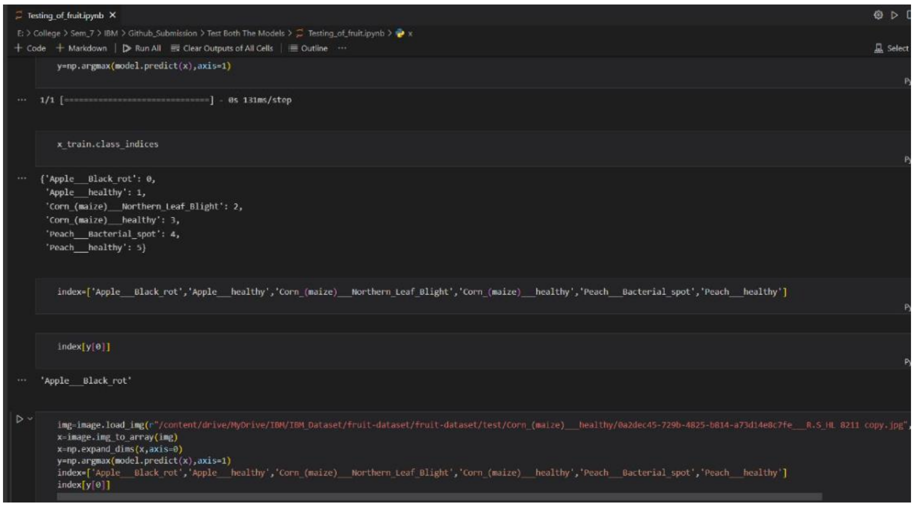
```
model.save('fruit.h5')
```


Testing_of_fruit.ipynb
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('fruit.h5')
img



Testing_of_fruit.ipynb
img
x=image.img_to_array(img)
x
Output exceeds the size limit. Open the full output data in a text editor
array([[[122., 107., 110.],
[171., 158., 161.],
[184., 169., 172.],
...,
[218., 203., 206.],
[143., 128., 131.],
[154., 139., 142.]],
[[144., 129., 132.],
[159., 144., 147.],
[145., 130., 133.],
...,


Testing_of_fruit.ipynb
E: > College > Sem_7 > IBM > Github_Submission > Test Both The Models > Testing_of_fruit.ipynb > x
+ Code + Markdown | Run All Clear Outputs of All Cells | Outline
x
Output exceeds the size limit. Open the full output data in a text editor
array([[[[122., 107., 110.],
[173., 158., 161.],
[184., 169., 172.],
...,
[218., 203., 206.],
[143., 128., 131.],
[154., 139., 142.]],
[[144., 129., 132.],
[159., 144., 147.],
[145., 130., 133.],
...,
[141., 126., 129.],
[136., 121., 124.],
[134., 119., 122.]],
[[142., 127., 130.],
[157., 142., 145.],
[165., 150., 153.],
...,
[156., 141., 144.],
[112., 97., 100.],
[129., 114., 117.]],


Testing_of_fruit.ipynb
E: > College > Sem_7 > IBM > Github_Submission > Test Both The Models > Testing_of_fruit.ipynb > x
+ Code + Markdown | Run All Clear Outputs of All Cells | Outline
x=np.expand_dims(x,axis=0)
x
Output exceeds the size limit. Open the full output data in a text editor
array([[[[122., 107., 110.],
[173., 158., 161.],
[184., 169., 172.],
...,
[218., 203., 206.],
[143., 128., 131.],
[154., 139., 142.]],
[[144., 129., 132.],
[159., 144., 147.],
[145., 130., 133.],
...,
[141., 126., 129.],
[136., 121., 124.],
[134., 119., 122.]],
[[142., 127., 130.],
[157., 142., 145.],
[165., 150., 153.],
...,
[156., 141., 144.],
[112., 97., 100.],
[129., 114., 117.]],





**DEMO VIDEO LINK:** (323) Plant Disease Prediction and Fertilizer Recommendation - YouTube

**GITHUB LINK:** **https://github.com/IBM-EPBL/IBM-Project-12065-1659368857**