SBM COLLEGE OF ENGINEERING AND TECHNOLOGY

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND
COMMUNICATION ENGINEERING.

FERTILIZER RECOMMENDATION SYSTEM FOR DIESEASE PREDICTION

A PROJECT REPORT BY

P.NAGAJOTHI - 921619106041
M.BHARATHI PRIYA - 921619106009
T.DIVYADHARSHINI - 921619106018
S.YUVASHREE - 921619106067

TABLE OF CONTENTS

- INTRODUCTION
- LITERATURE SURVEY
- PROBLEM STATEMENT
- EMPATHY MAP
- BRAINSTROM
- PROPOSED SOLUTION
- PROBLEM SOLUTION FIT
- DATAFLOW DIAGRAM
- SOLUTION REQUIREMENTS
- TECHNOLOGY STACK
- SOLUTION ARCHITECTURE
- SPRINT DELIVERABLE PLAN
- DATASET
- DOCUMENTAION
- FERRTILIZER RECOMMENTATION SYSTEM FOR DISEASE PREDICTION
- FINAL CODE
- IBM PNT2022TMID49029
- PROJECT REPORT
- DEMO LINK

LITERATURE SURVEY

1)In Year 2022,"A Machine Learning Based New Recommendation System To The Farmer".

AUTHORS: D.N.V.S.L.S indra, M.Sobhana, A.H.L.Swaroop, V.PhaniKumar.

Totally 54% of India's land area is deemed arable, making it the world's largest agrarian economy. Soil infertility owing to over fertilization, as well as a lack of access and awareness of contemporary agricultural practices, are the different factors that contribute to low agricultural production. The main purpose of this research work is to develop a machine learning-based recommendation system to increase agricultural productivity. A variety of datasets were used in this study to design and develop advanced models to estimate the crop, recommend fertiliser, and identify plant disease. An algorithm called MobileNet uses an image of a leaf to identify the disease present in a plant. The XGBoost model predicts a suitable crop based on the local soil nutrients and rainfall. Random Forest [RF] model was used to propose fertilizer and develop ideas for improving soil fertility depending on nutrients present in the soil. When compared to other approaches, the proposed model delivers a high level of accuracy.

2) In Year 2022,"Crop Prediction And Disease Detection System".

AUTHORS: Sambhav Bhansali, Punit Shah, Jinay Shah, Priyal Vyas,Poonam Thakre.

Economy of India highly depends on agriculture. Still traditional ways of recommendations are used for agriculture. Currently, farmers use traditional ways of approximations for amount of fertilizer used and the type of crop to be sown. Agriculture extremely depends on the type of soil and climatic condition of the region. Therefore, it becomes vital to create advancement in this field. With the help of Machine Learning and Deep Learning Techniques we will create a Web-App which will be one-stop solutions for information regarding the agriculture. Crop and fertilizer recommendation system will help the farmers in increasing their yield production. We are going to take the soil parameters along with the weathers API to figure out the most suitable crop for that region. Using the decision tree and navies bayes algorithm we will make the recommendation model which will use the N-K-P, Ph. value.Basis on the crop and region of farming we will recommend the fertilizer and its uses to boost the yield productivity for farmers. Sometimes due to unwanted excess of rainfall or the pest attack can cause disease to crops. We will use the image classification technique where the user can upload the picture of the affected plant/crop and the system will figure out the type of disease which will be done using Support Vector Machine (SVM) or using the neural network techniques. And this disease detection will suggest that how that plant/crop can be cure or prevent. The aim is to make a common system for all the features and provide the results with the best accuracy for all the crops over most of the regions all over the India. Also, the price and news section will keep the farmers updated with daily market prices and government schemes and policies related to the agriculture and farming.

3) In Year 2022,"Agro-Farm-Crop,Fertilizer & Disease Prediction".

AUTHORS: SanidhyaPurohit, DeepSanghani, NamanSenjaliya, Prof.Anuradha Kapoor

Data mining is a rising studies area in crop yield analyis Yield predicition is a complete essential problem in agriculture. Any farmer is interested in knowing how muchyield he is about to expect also, it will end-user-helpful to farmers for indicting which fertilizers to be used as well as knowing the crop diseases all at one place. The project comes with a model to be precise and accurate in predicting crop, fertilizers, Crop disease and deliver the end-user with proper recommendations about the required fertilizer ratio based on atmospheric and soil parameters of the land which enhance to increase the crop yield and increase farmer revenue.

4) In Year 2021,"Neural Network Based Fertilizers Recommendation System For Disorder Classification And Prediction In Petal Images".

AUTHORS: N.Valarmathi, M.Vengateshwaran, Kalaimani Shanmugam, R.Sudha.

The point of farming isn't just to take care of the ever-developing populace but at the same time is a basic wellspring of vitality and an answer for the emergency of an Earth-wide temperature boost. Determination of plant ailment is basic for early finding and control of it. The unaided eye method is generally utilized for the conclusion of ailments. This methodology requires experts who can recognize varieties in leaf shading. Ordinarily a similar malady is characterized by a few specialists as a different sickness. This arrangement is exorbitant, in light of the fact that it requires nonstop expert management. Makers need to follow their yields and perceive the primary signs at modest costs so as to abstain from spreading even a plant malady and spare a lot of income. Recruiting qualified ranchers can't be reasonable especially in far off geologically detached zones. AI calculations in an image can give a substitute strategy to following plants and an expert can deal with such a way to deal with offer their types of assistance at a lower cost. It incorporates picture division which incorporates the dynamic shape strategy and the picture arrangement approach which incorporates a neural system calculation for foreseeing various kinds of ailments. Or on the other hand grow the way to deal with suggest the composts dependent on the examination of power with estimations.

5) In Year 2020,"Design and Implementation of Fertilizer Recommendation System for Farmer".

AUTHORS: Dr.S.UshaKiruthikika, Dr.S.Kanaga Suba Raja,S.R. Ronak,S.Rengarajen, P.Ravindran.

India is an agrarian nation. But creating a profitable yield for the farmer in each crop cycle is becoming a major challenge on various factors. Picking the reasonable fertilizer for the land and yield is an important and basic part of agriculture. Deciding the supplement levels in soil utilizing lab hardware can be restrictively costly, particularly in developing nations. The current frameworks on deciding soil nutrient substance and proposal for fertilizer isn't sufficiently proficient efficient enough. This paper introduces a compelling technique for estimation of nutrient dimension in soil and suggestion for appropriate fertilizer. The proposed methodologies comprise of four stages: soil analysis, data pre-processing, data analysis and Recommendation. The soil sample is analyzed using an IoT based device utilizing NPK sensor with two electrodes are set to calculate collect the NPK ratio of the soil nutrient and for pre-processing, the data gathered from sensors are figured into correct dataset and machine learning algorithm is utilized to recognize the reasonable fertilizer. This venture is extremely valuable to farmer to pick the right fertilizer toward the start of product cycle and amplify the yield.

6) In Year 2019,"Fertilizers Recommentation System For Disease Prediction in Tree Leave".

AUTHORS: R.Neela , P.Nithya.

Agriculture is the main aspect of country development. Many people lead their life from agriculture field, which gives fully related to agricultural products. Plant disease, especially on leaves, is one of the major factors of reductions in both quality and quantity of the food crops. In agricultural aspects, if the plant is affected by leaf disease then it reduces the growth of the agricultural level. Finding the leaf disease is an important role of agriculture preservation. After pre-processing using a median filter, segmentation is done by Guided Active Contour method and finally, the leaf disease is identified by using Support Vector Machine. The disease-based similarity measure is used for fertilizer recommendation.

7) In Year 2018," Soil Toxicity Prediction and Recommendation System Using Data Mining in Precision Agriculture".

AUTHORS: Mayuri Pawar, Geetha Chillarage.

India is agricultural land. India ranks second worldwide in agriculture output, but GDP share is declining. There are many factors contribute for declining agriculture GDP which are inadequate irrigation, inadequate power supply, changing environmental conditions, conventional agricultural method etc. In this paper, the proposed system can help farmers by making them aware about their soil conditions. Farmers can maximize crops yield by knowing proportion of nutrients present in the soil. Soil toxicity

affects the soil nutrients which indirectly affects crops health. The proposed system predicts the level of toxicity present in the soil and makes farmer aware about it. Many farmers are depending on rainfall which is the one of the factor for poor growth and decreases crops yield. Thus the proposed system recommends the farmer about the crop, fertility of soil, level of toxicity and water supply. For this recommendation system, sensor's accuracy is very important as well as classification algorithm. For classification, decision tree J48 algorithm is used which is simple to implement and having more accuracy as compared with other classification algorithms. Issue of power supply can be overcome by using solar panel system.

8) In Year 2016,"Crop Recommendation System for Precision Agriculture".

AUTHORS:S.Pudumalar,E.Ramanujam,R.HarineRajashree,C.Kavya, T.Kiruthika,J.Nisha.

Data mining is the practice of examining and deriving purposeful information from the data. Data mining finds its application in various fields like finance, retail, medicine, agriculture etc. Data mining in agriculture is used for analyzing the various biotic and abiotic factors. Agriculture in India plays a predominant role in economy and employment. The common problem existing among the Indian farmers are they don’t choose the right crop based on their soil requirements. Due to this they face a serious setback in productivity. This problem of the farmers has been addressed through precision agriculture. Precision agriculture is a modern farming technique that uses research data of soil characteristics, soil types, crop yield data collection and suggests the farmers the right crop based on their sitespecific parameters. This reduces the wrong choice on a crop and increase in productivity. In this paper, this problem is solved by proposing a recommendation system through an ensemble model with majority voiting technique using random tree.

Fertilizer Recomentation System for Disease Prediction

PROBLEM STATEMENT:

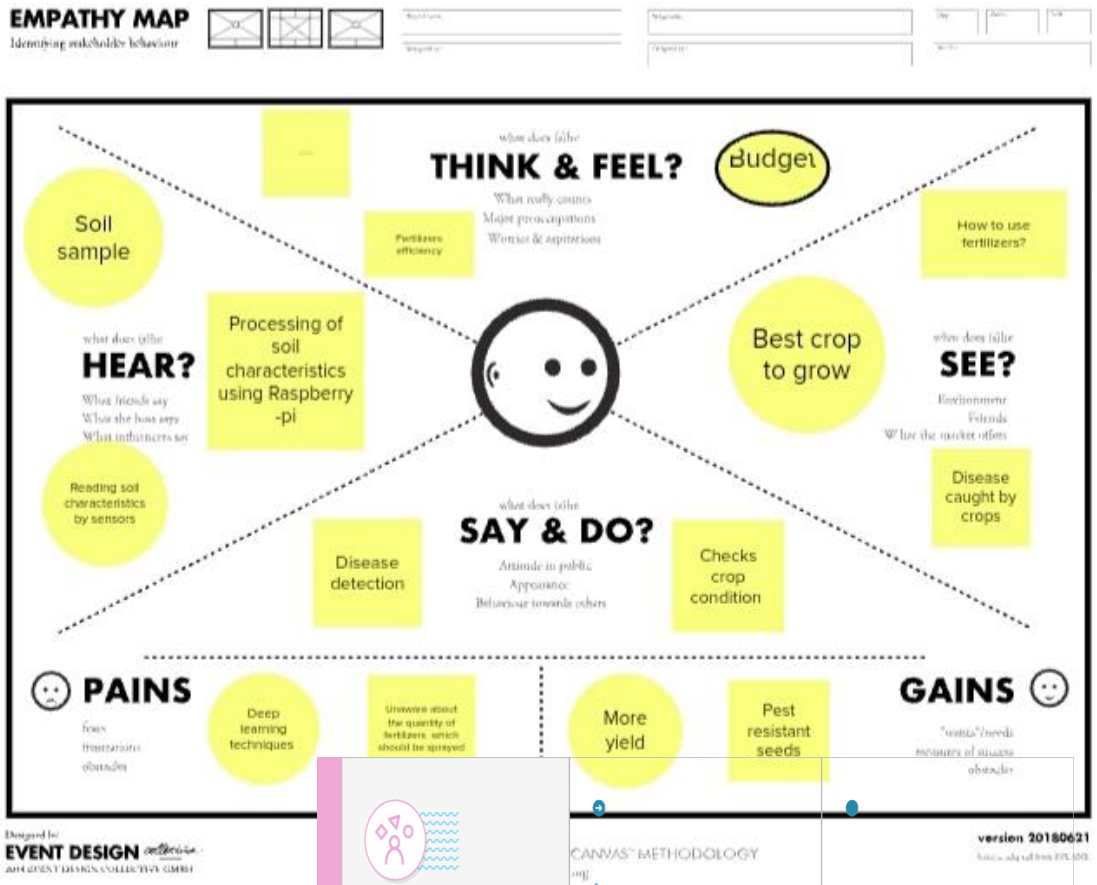
Mr.Narasimma Rao is a 65 years old man. He had a own farmingland and do Agriculture for past 30 Years , In this 30 Years he Faced a problem in Choosing Fertilizers and Controlling of Plant Disease.

- Narasimma Rao wants to know the better recommendation for fertilizers for plants with the disease.
- He has faced huge losses for a long time.
- This problem is usually faced by most farmers.
- Mr. Narasimma Rao needs to know the result immediately.

Who does the problem affect?	Persons who do Agriculture
What are the boundaries of the problem?	People who Grow Crops and facing Issues of Plant Disease
What is the issue?	In agricultural aspects, if the plant is affected by leaf disease, then it reduces the growth and productiveness. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants.
When does the issue occur?	During the development of the crops as they will be affected by various diseases.

Where does the issue occur?	The issue occurs in agriculture practicing areas, particularly in rural regions.
Why is it important that we fix the problem?	It is required for the growth of better quality food products. It is important to maximise the crop yield.
What solution to solve this issue?	An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.
What methodology used to solve the issue?	Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

Empathy Map

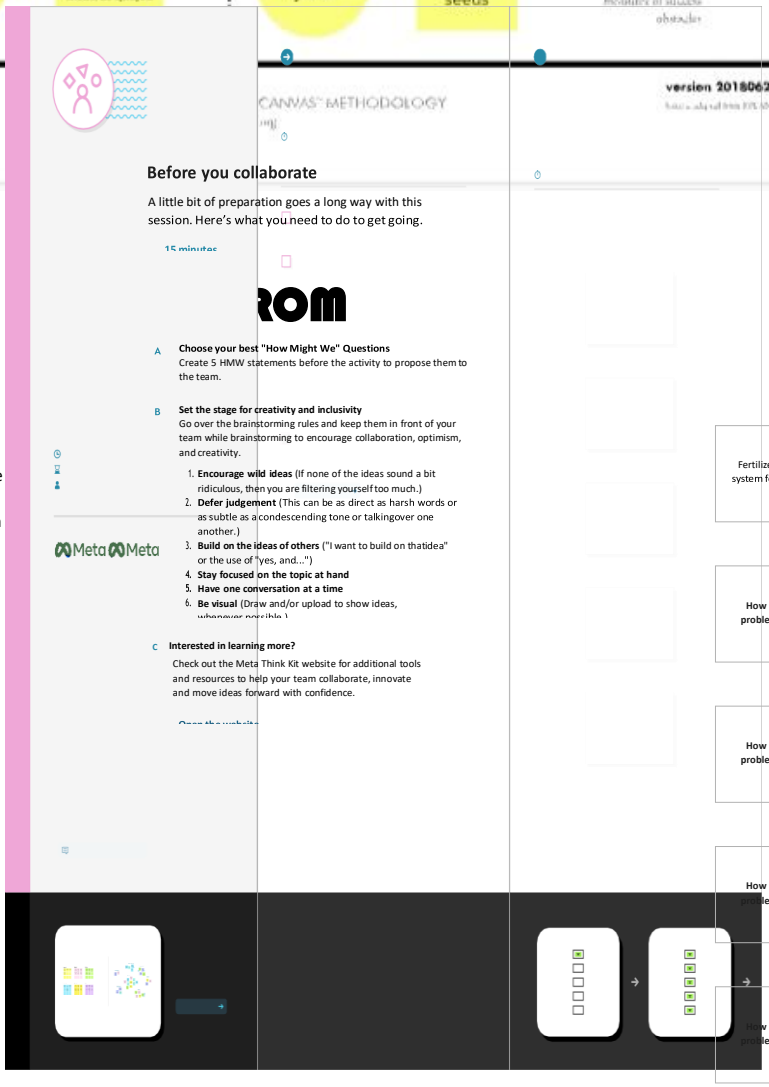


Conducting a brainstorm

Executing a brainstorm isn't unique; holding a productive brainstorm is. Great brainstorms are ones that setthe stage for fresh and generative thinking through simple guidelines and an open and collaborative environment. Use this when you're just kicking-off a new project and want to hit the ground running withbig ideas that will move your

- **15 minutes** to prepare
- **30-60 minutes** to collaborate
- **3-8 people** recommended

Created in partnership with



Project Design Phase-I
Proposed Solution
Template

Date	18 oct 2022
Team ID	PNT2022TMID49029
Project Name	Project – Fertilizer recommendation system fordisease predication
Maximum Marks	2 Marks

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Agriculture is having a great impact on the country's economy . Different diseases affect plant that reduces their production and is a major threat to foodsecurity. The major drawbacks that the farmers of our country are currently facing inludes crop failure,lack of adequate knowledge,crop damage due to ignorance,lack of professional assistance, inaccesibility to agro-tech solutions.Most of the diseases are detected in later stagethat to manually which is time consuming and results in heavy loss so it is important to build an automated system that detects disease at early stage and provides fertilizer recommendation accordingly. Farmer usually detect the crop diseases and plant disease with their naked eye which makesthem take tough decisions on which fertilizer touse. It is necessary to develop crop yield prediction and fertilizer recommendation system which predicts crop yield based on soil nutrients crop yield data and recommend fertilizer for selectedcrop based on different dataset like fertilizer data,location data,and crop yield data.

2.	Idea / Solution description	Implementation of artificial intelligence for identification of pests and recommendation ofinsecticides using TPF-CNN. The combination of two major things requiredin farming in one system is spraying proper insecticides and adding the needed fertilizer amount to the soil.

		Implementation of soil sensor for soil NPKnutrient analysis and recommendation of fertilizers accordingly. An automated system is built that takes the input as pictures of leaves which is uploaded bythe user, identifies different disease on plants by checking the symptoms shown on the leavesof the plant. Deep learning techniques are used to identifythe diseases and suggest the fertilizer needed for the plant.
3.	Novelty / Uniqueness	Efficient approach for controlling the overuse ofinsecticides and fertilizer in farming. Time efficient aperture compared toKNN,SVM,and ANN. It can suggest and predict best and correct fertilizer for disease in the plant. It does not require user to consult any specialistfor identificationfor disease that affected the leaves and the fertilizer that is required for the same. It detects plant disease at a early stage.
4.	Social Impact / Customer Satisfaction	Yield right crop at the right time , balancing the crop production,control plant disease,economicgrowth and planning to reduce the crop scarcity. Hence to detect and recognize the plant diseaseand to recommend fertilizer it is necessary to provide symptoms in identifying the disease at its earliest. Hence implemented new fertilizer recommendation system for crop diseaseprediction. The whole process of identifying disease andrecommendation of fertilizer happens just byuploading image so it is user friendly. It helps farmers to get good yield out of thecrop. People will get good quality food products.
5.	Business Model (Revenue Model)	Typically dedicate 10% of their AI investment toalgorithms,20% to technoligies and 70% to embedding AI into business processes and agilewaysof working. In other words, companies invest twice as muchin people and processes as they do in technologies. Social media is the best way to spread the wordabout our application. And with influencers we can reach out to people.

		Clustering and targeting the farmers for identifying disease on their plants and recommending them fertilizers for the same.
Cultivators Gardeners Plant	Scalability of the Solution	This can be improved by introducing online purchases crops,fertilizers , etc..,easily. It can be used in research areas to study about the diseases in plants and the best fertilizer thatcan be recommended for it among the list of fertilizer available. It can be used by anyone in the world.

The farmers may not be aware of the infections or diseases that affected their plants.
Even if they did, the nutrients required to cure may not be known. Identification of the right fertilizers and the quantity

Image acquisition is followed by pre processing and segmentation. Leaves are classified using Support Vector Machines algorithm. Fertilizers for affected leaves is

Project Title:

Project Design Phase-I - Solution Fit Template

PNT2022TMIDxxxxxx

Team ID:

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Explore AS, differentiate

Fertilizers contain specific nutrients that are required for the proper development of the plant body.Some fertilizers soil fit plants indirectly by increasing water retention

An system that takes the image of leaves as input and identifies the different symptoms to decide on the disease that affects plant.
This will be done usinf Doop learning techniques.

Online portal is for accepting the input image and displaying the recommended fertilizers.

Soil may not have adequate quantities of all nutrients.Rate of replenish of soil nutrients is much slower than the rate pf consumption.Hence fertilizers are required to balance these rates by providing enough nutrients to the soil and plants

While offline, the image pre processing, segmentation, disease prediction etc are done

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

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits
indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Identify strong TR & EM

3. TRIGGERS

TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.

If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

8. CHANNELS of BEHAVIOUR

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

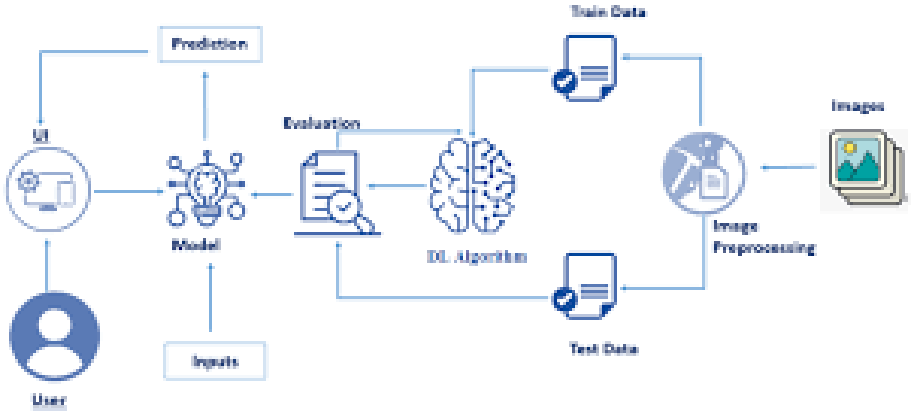
Project Design Phase-II
Data Flow Diagram & User Stories

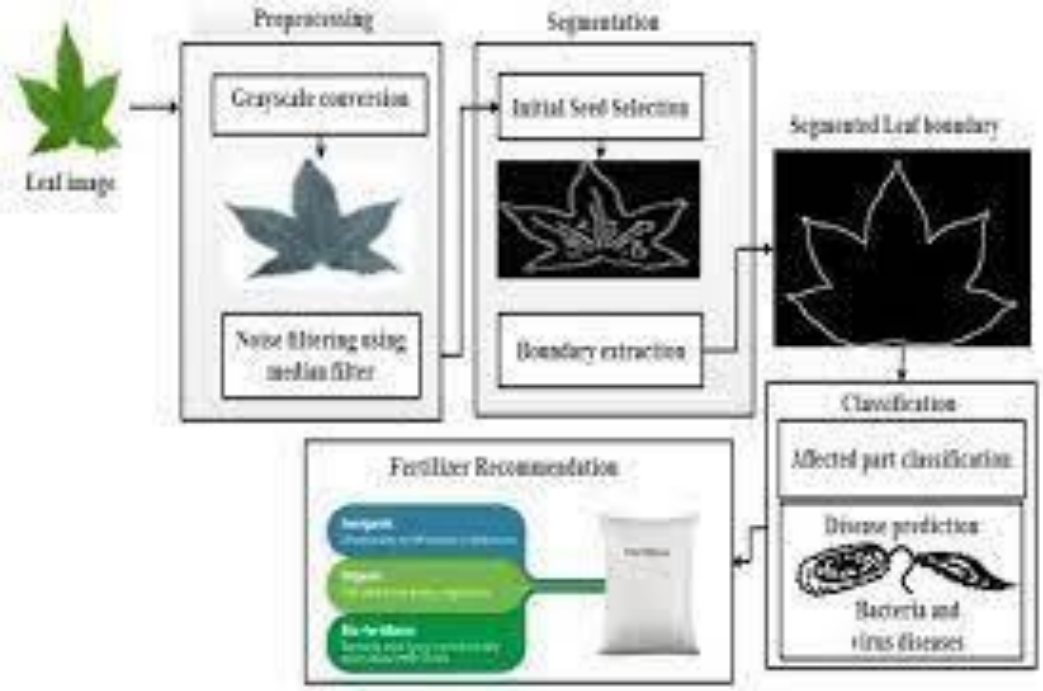
Date	20 October 2022
Team ID	PNT2022TMID49029
Project Name	Project – Fertilizer recommendation system for disease prediction
Maximum Marks	4 Marks

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example:


Train Data
Prediction
Images
UI
Evaluation
Model
DL Algorithm
Image
Preprocessing
Test Data
Inputs
User


Leaf image
Preprocessing
Grayscale conversion
Noise filtering using median filter
Segmentation
Initial Seed Selection
Boundary extraction
Separated Leaf boundary
Classification
Affected part classification
Disease prediction
Bacteria and virus diseases
Fertilizer Recommendation

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	Acceptan
Developer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access dashb
Assistant developer	Login	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive email & clic
Customer Care Executive	Worker	USN-3	As a user, I can register for the application through Facebook	I can register dashboard wi Lo
Administrator	Worker	USN-4	As a user, I can register for the application through Gmail	Data is p
Customer web user	Login	USN-5	As a user, I can log into the application by entering email & password	I get the re opened throu

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	20 October 2022
Team ID	PNT2022TMID49029
Project Name	Project – Fertilizer recommendation system for disease prediction
Maximum Marks	4 Marks

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	NAME: Enter Name EMAIL: Enter Mail PASSWORD: Enter Password PHONE: Enter Phone number (Registration through form Registration through gmail Registration through linked IN)
FR-2	User Confirmation	Thank you for your email We have received a request from your mail. Please confirm to proceed further. If any queries please contact our help centre to help get you an instant answer to your question. (Confirmation via Email Confirmation via OTP)

FR-3	Product Features	It provides data of the fertilizer to full fill the user's demands. Reading soil and plants characteristics by sensors.
FR-4	Testing Features	This estimation of nutrient in soil is done using an NPKmonitoring unit with Arduino UNO as the microcontroller to read the values from it. Convolutional Neural Networks (CNN) algorithms recommend appropriate fertilizers that can be used toprevent damage to plants from pathogenic viruses. The fertilizer data is collected from various markets about the brand name and NPK ratio of the fertilizer iscollected.
FR-5	Objective	Smart farming and precession farming can be advanced by calculating NPK value for more accurate values.

		Analyzing the soil condition of any region and the requirements of the farmer to maximize the soil production.

Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is very easily usable for the customer. The customer gets a notification whenever the insufficient nutrient or disease is detected in the plant. Help the farmer to identify the disease.
NFR-2	**Security**	Security is very much concerned regarding the datacollected and customer details. These securities are mainly related to the cloud services, they have strict security across the network. The proposed method combines the two major aspects in farming,pest identification and insecticides recommendation.
NFR-3	**Reliability**	The use of artificial intelligence gives appropriate result. The CNN algorithm model has 95% accuracy. The reliability is more for the customers. It is easy to use so that health issues can beavoided.
NFR-4	**Performance**	The app runs on a mobile device under various loadsand circumstances. Precision fertilizer and precision crops is mostly used.

NFR-5	**Availability**	There is a high availability for user's access. Anyonecan make use of it. Reduces the losses as ammonia,nitrate leaching,apply the right rate and accurately.
NFR-6	**Scalability**	It is an effective way to minimize the damages for aplant by early detection of disease and recommending suitable fertilizers. If the soilis not replenished with nutrients through fertilizer,crop yield will deteriote over time.

Project design phase-II Technology stack(Architecture & stack)

Date	20 October 2022
Team id	PNT2022TMID49029
Project name	Fertilizer recommendation system for disease prediction
Maximum number	4 marks

Technical Architecture:

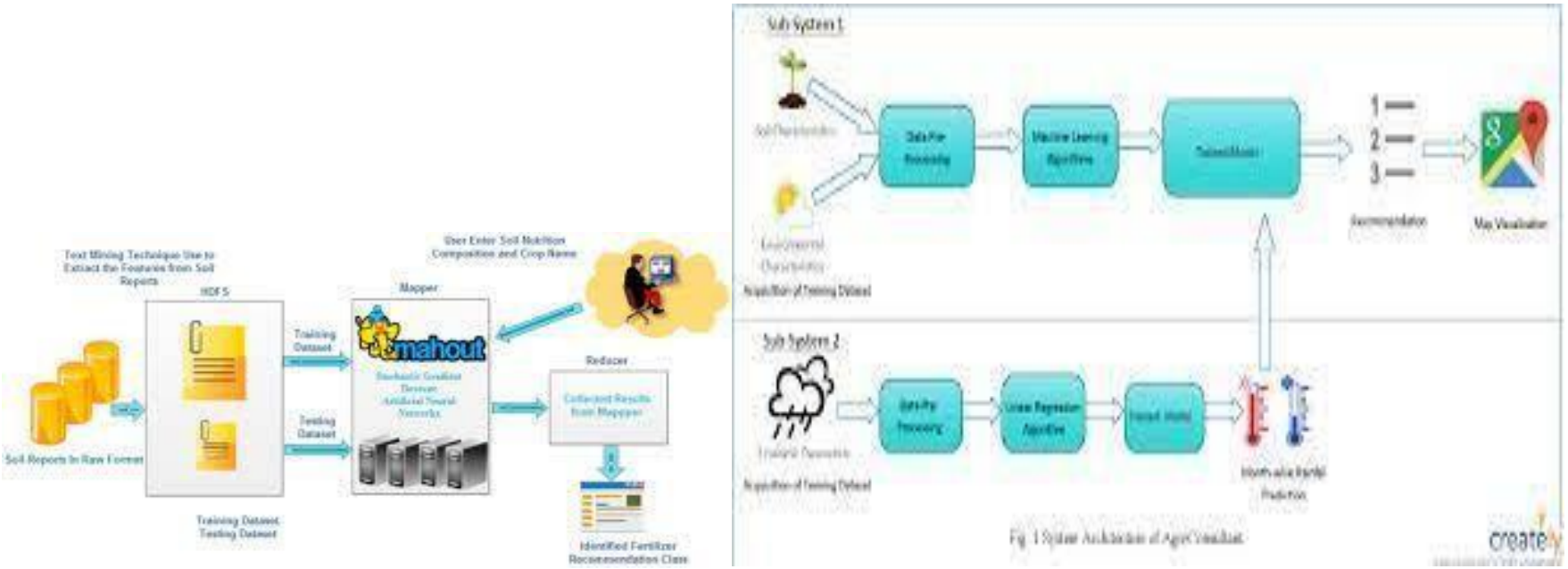

Sub System 1
Sub System 2
1
2
3
Recommendation
Map Visualization
Mapper
Reducer
Training Dataset
Testing Dataset
Soil Reports In Raw Format
Training Dataset
Testing Dataset
Identified Fertilizer
Recommendation Class
createx

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Anaconda Navigator, Tensor flow, Keras, Flask
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Response time, Throughput, CPU and network usages, etc.
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	All kind of users.
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Predicting disease, image processing, Visual similarity, rules, machine learning techniques, etc**.**

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	18 October 2022
Team ID	PNT2022TMID49029
Project Name	Project - Fertilizers Recommendation system for disease prediction
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can sign up and register respective sites to access the required details and data. And import the required libraries for the processes.	2	High	P.Nagajothi M.Bharathi priya T.Divyadharshini S.Yuvashree
Sprint-2	Login	USN-2	As a user, I will access the page and test and train the CNN model to predict or detect the plant disease.	2	High	P.Nagajothi M.Bharathi priya T.Divyadharshini S.Yuvashree
Sprint-3	Customer Service	USN-3	As a customer care executive , I am available to the customers . so if the customers have any issues or in need of any assistance they will get help and solve them.	1	Medium	P.Nagajothi M.Bharathi priya T.Divyadharshini S.Yuvashree
Sprint-4	Dashboard	USN-4	As a user, I will have the access to know about the activities in the plant.	2	High	P.Nagajothi M.Bharathi priya T.Divyadharshini S.Yuvashree

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	04 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		09 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		12 Nov 2022

Velocity:

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

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

AV:

Sprint 1 = 20/6= 3.33,

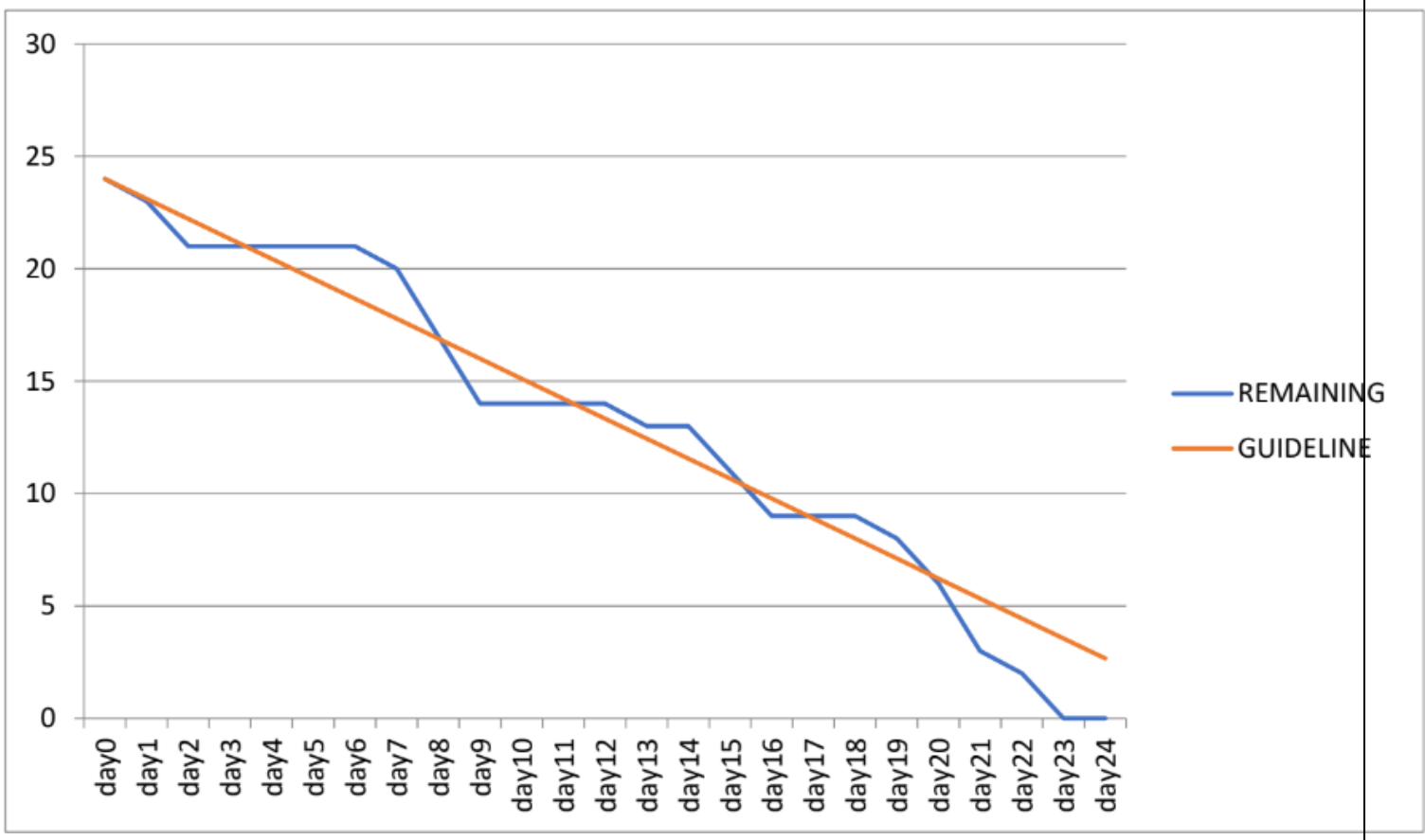
Sprint 2 = 20/6= 3.33,

Sprint 3 = 20/6= 3.33,

Sprint 4 = 20/6= 3.33.

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



Fertilizers Recommendation System for Disease Prediction

PROJECT REPORT

Submitted by

Team ID: PNT2022TMID49029

P.NAGAJOTHI

M.BHARATHIPRIYA

T.DIVYADHARSHINI

S.YUVASHREE

In partial fulfilment for the award of the

degreeOf

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

SBM COLLEGE OF ENGINEERING AND TECHNOLOGY,DINDIGUL.

1. INTRODUCTION

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

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

2. LITERATURE SURVEY

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

2.2 Proposed solution In this project work, a deep learning based neural network is used to train the collected datasets and test the same. The deep learning based

neuralnetwork is CNN which gives more than 90% classification accuracies. By increasing

the more number of dense layers and by modifying hyperparameters such as numberof epochs, batch size, the accuracy rate can be increased to 95% to 98%.

3. THEORITICAL ANALYSIS

3.1 Block diagram

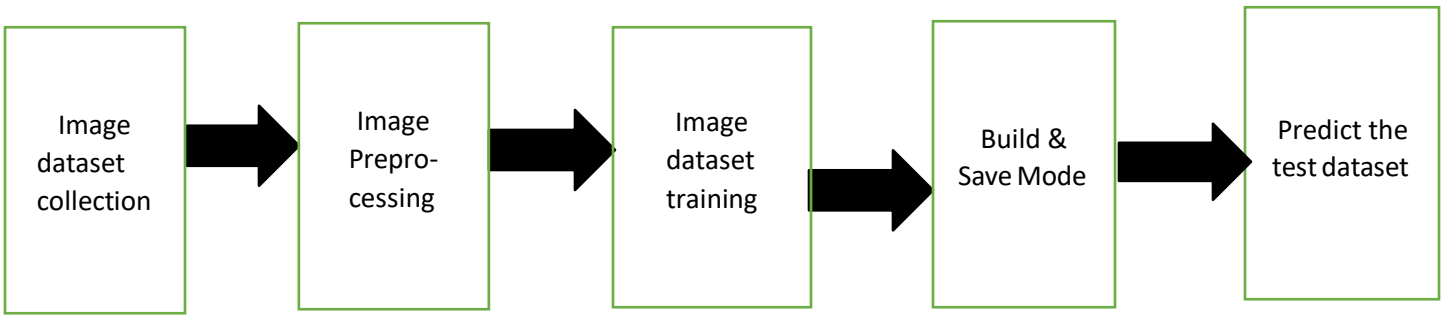


Figure.3.1. Block Diagram of the project

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

3.2 Hardware/Software designing

The software used for training and testing the dataset is Python. The Jupyter notebook (Notebook of IBM cloud also) is used for python programming. The neural network used for training and testing the model is Convolutional Neural Network (CNN).

The CNN has following layers:

- Convolutional layer (32x32 kernal (3x3))

- Max-pool layer (kernel(2x2))
- Flatten layer
- Dense layer (different layers with different size)
- Drop out layer (optional)
- Final output dense layer(size 6x1 for fruit dataset and 9x1 for Vegetable dataset)

In the preprocessing step, images are normalized to 1 and then resized to 128x128. The images are arranged in different batch sizes. Then train set and test set are formed from the collected datasets. In order to do the above steps in Python, the following Python libraries must be imported before starting the process:

- NumPy
- TensorFlow
- Keras
- Matplotlib (optional for data visualization)

The following activation functions used in the CNN training:

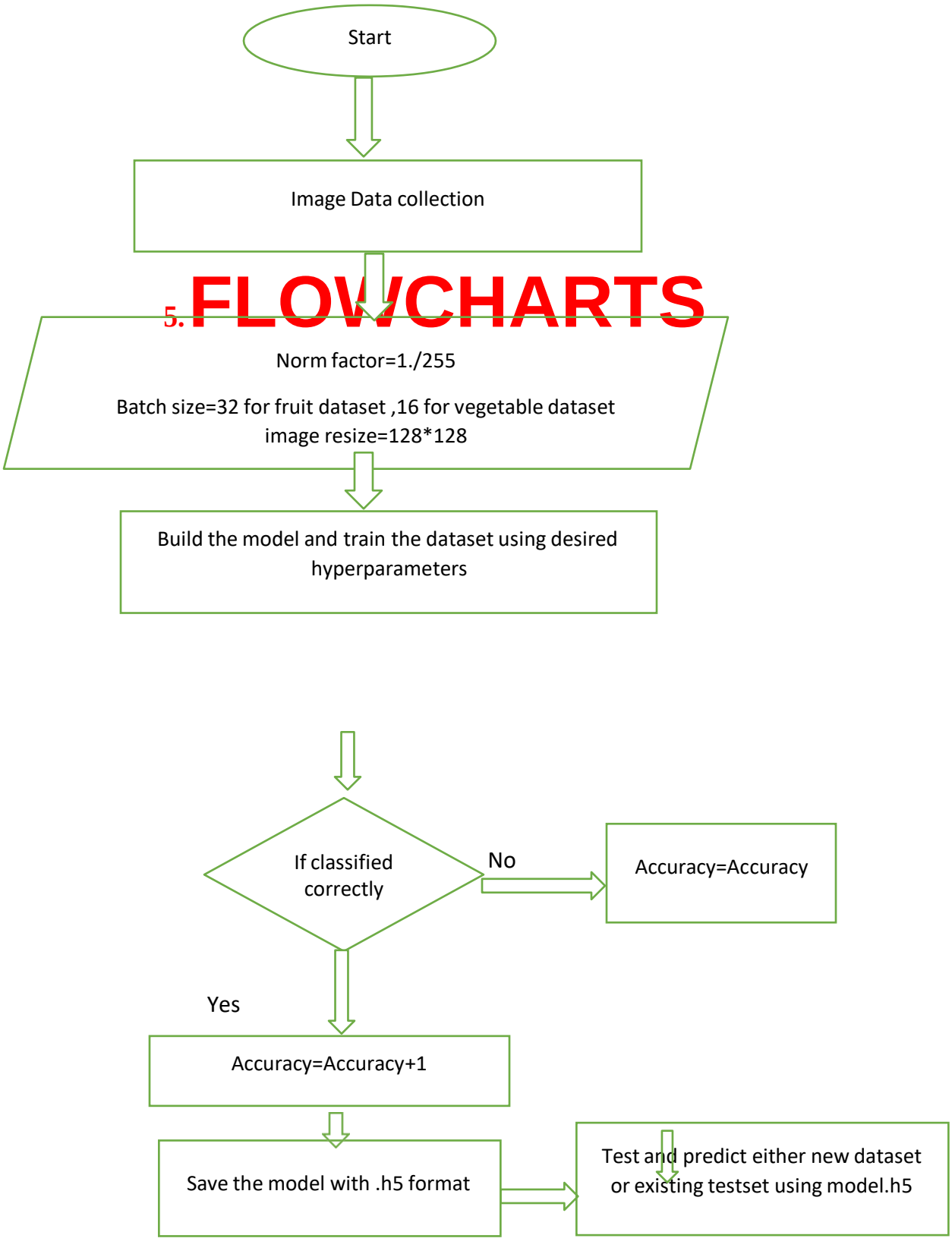
- RELU at the end of convolution layer and Max Pool layer
- SoftMax at the end of output dense layer
- For testing the dataset argmax is used, its an optional

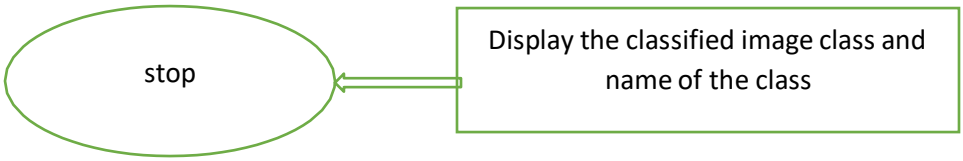
4. EXPERIMENTAL INVESTIGATIONS

Analysis made while working on the solution The batch sizes are varied and tested.For different batch sizes, the CNN gives different accuracies. The batch size

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

5. FLOWCHARTS




stop
Display the classified image class and name of the class

6. RESULTS

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

Figure.6.2 Test the Fruit dataset

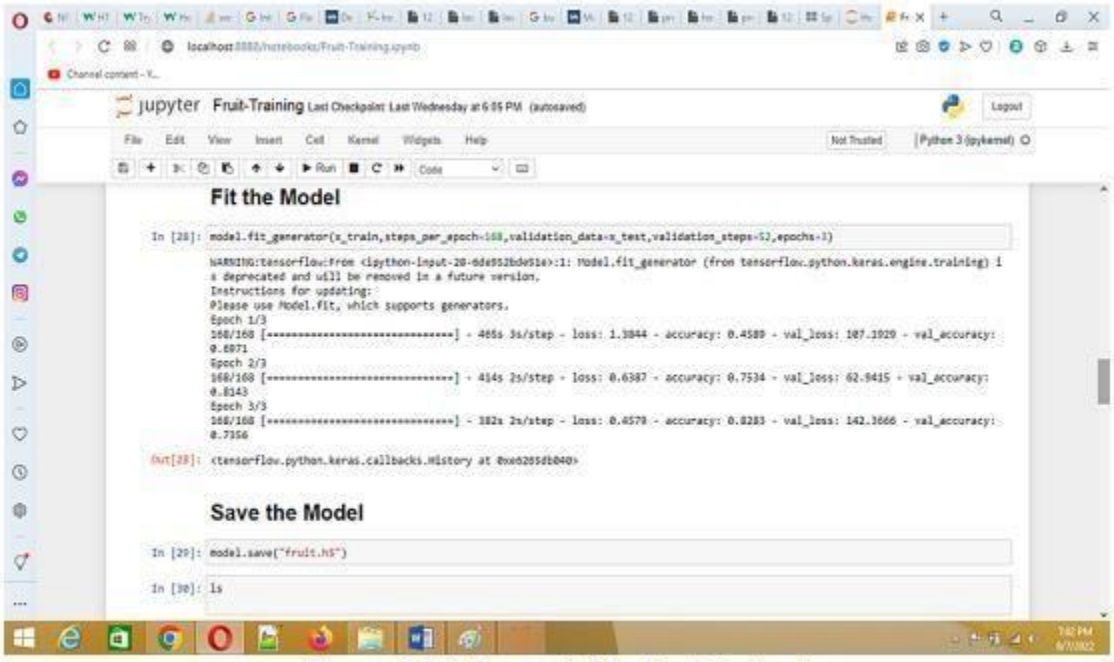


Figure.6.1. Fit a model for Fruit dataset

Train and Test Vegetable dataset

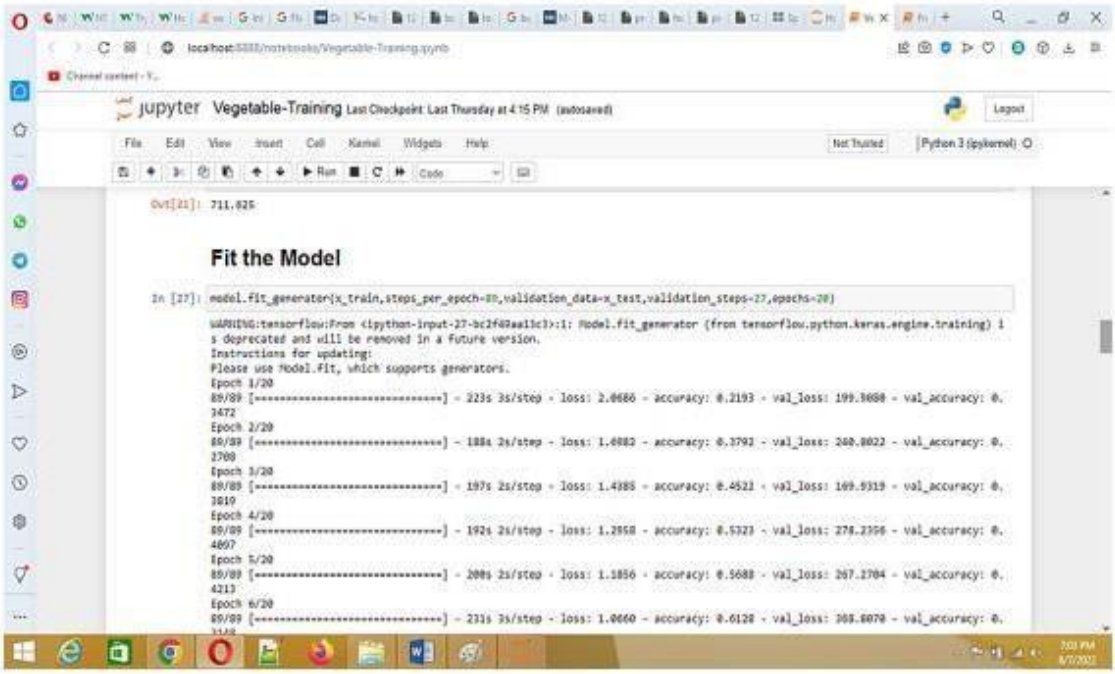


Figure.6.3. Train the Vegetable dataset

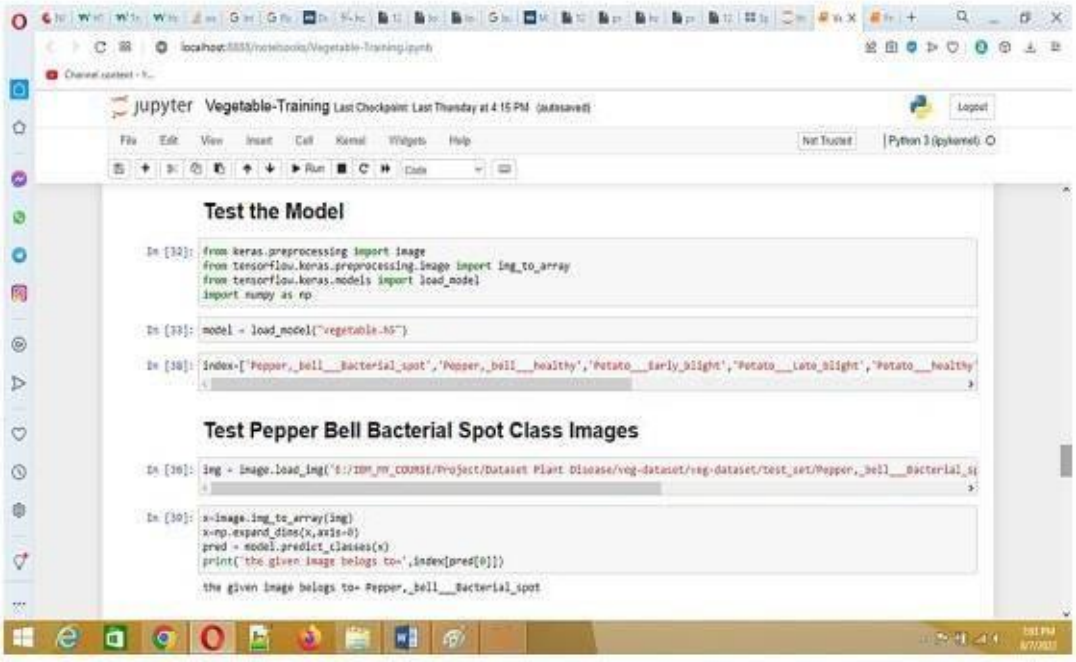


Figure.6.4. Test the Vegetable dataset

Train and Test Vegetable dataset IBM Cloud

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

(i). Fruit dataset

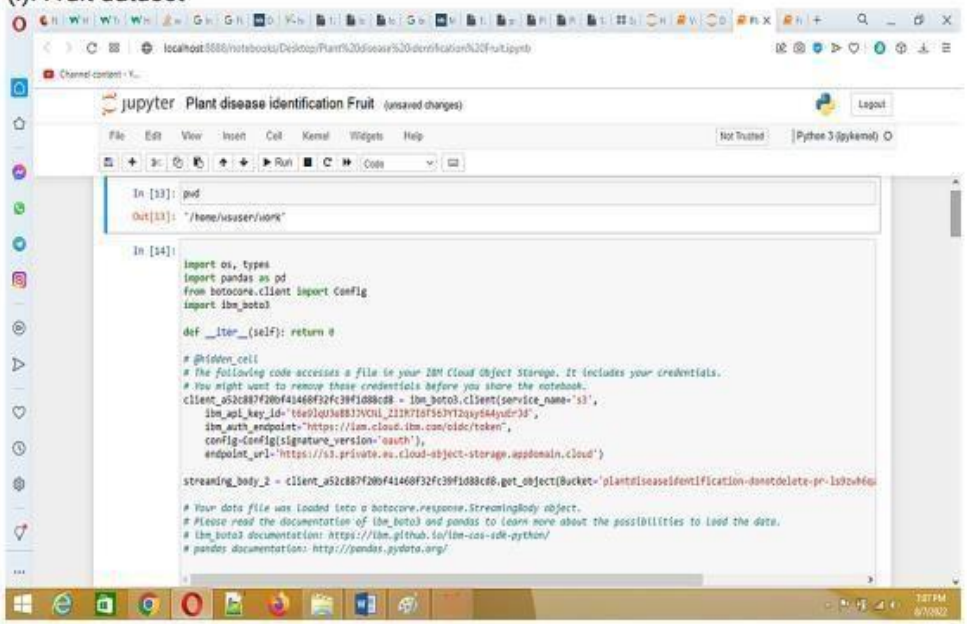


Figure.6.5. Training Fruit Dataset in IBM Cloud

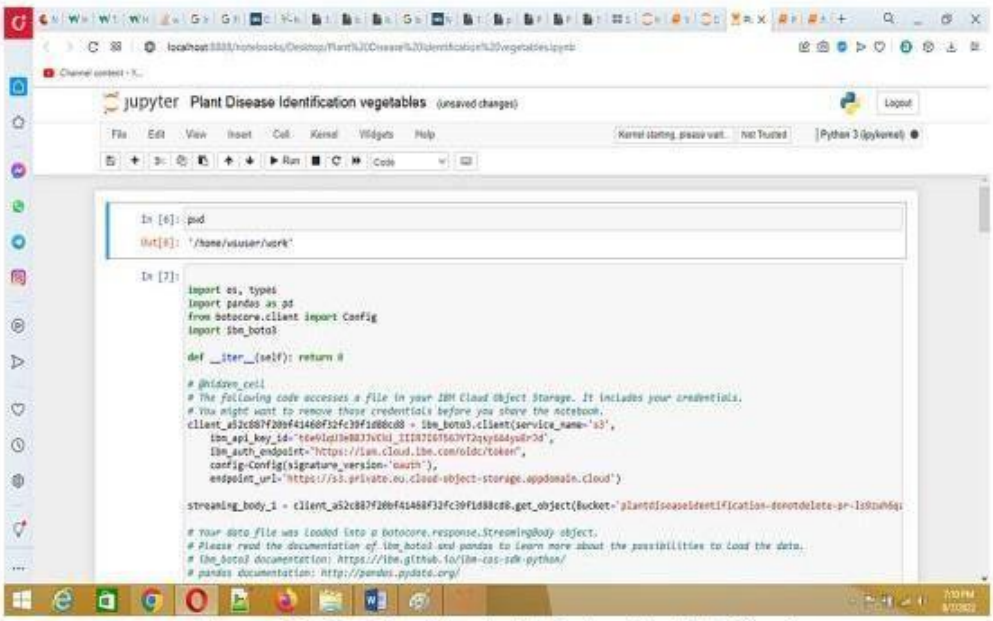
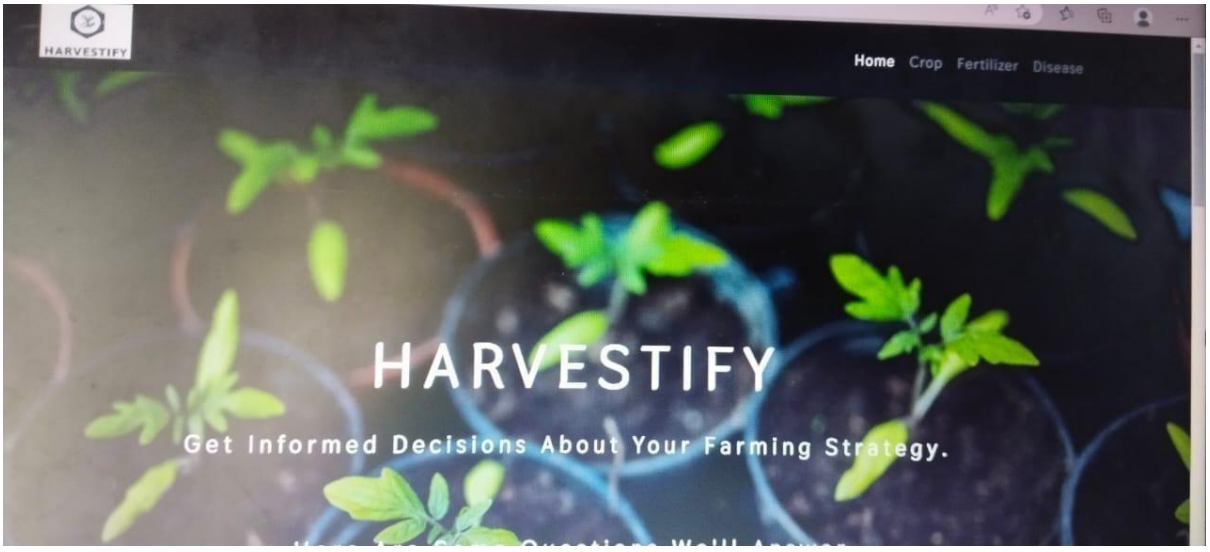
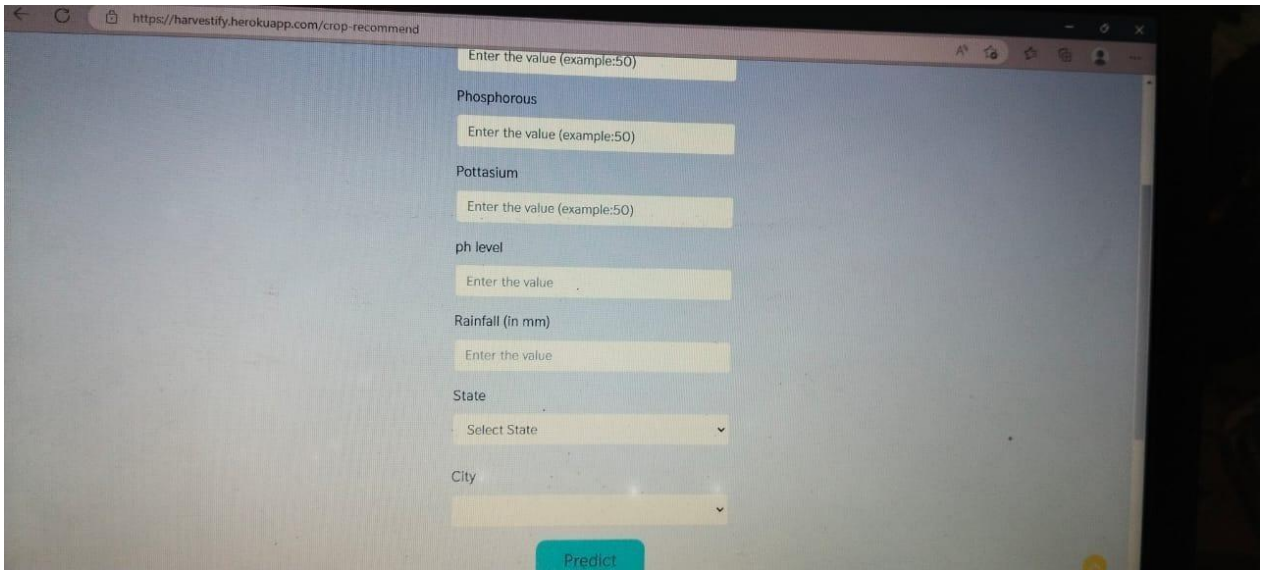
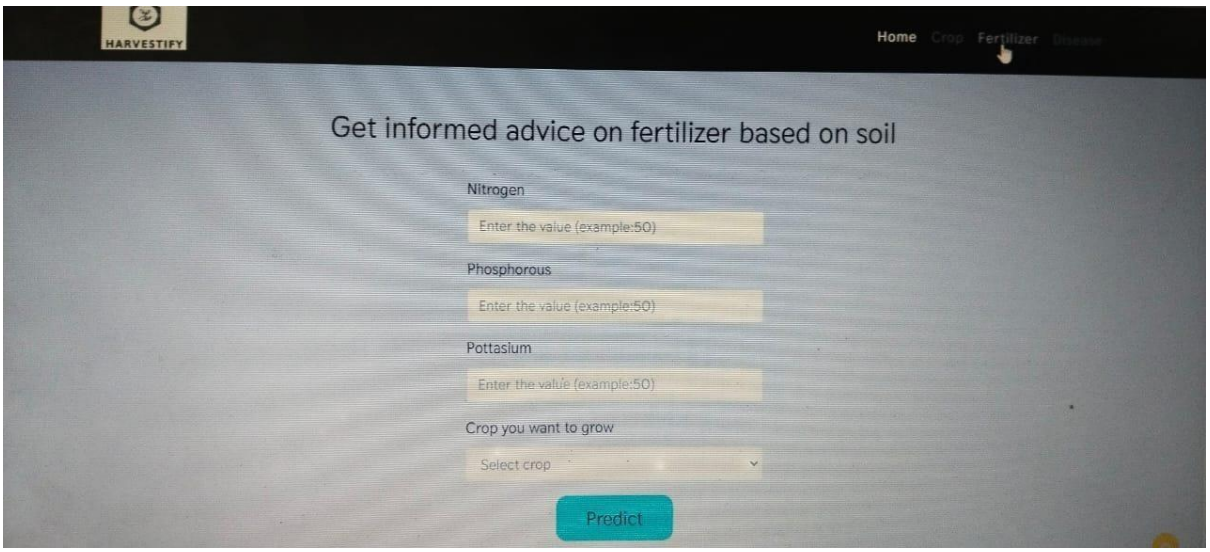


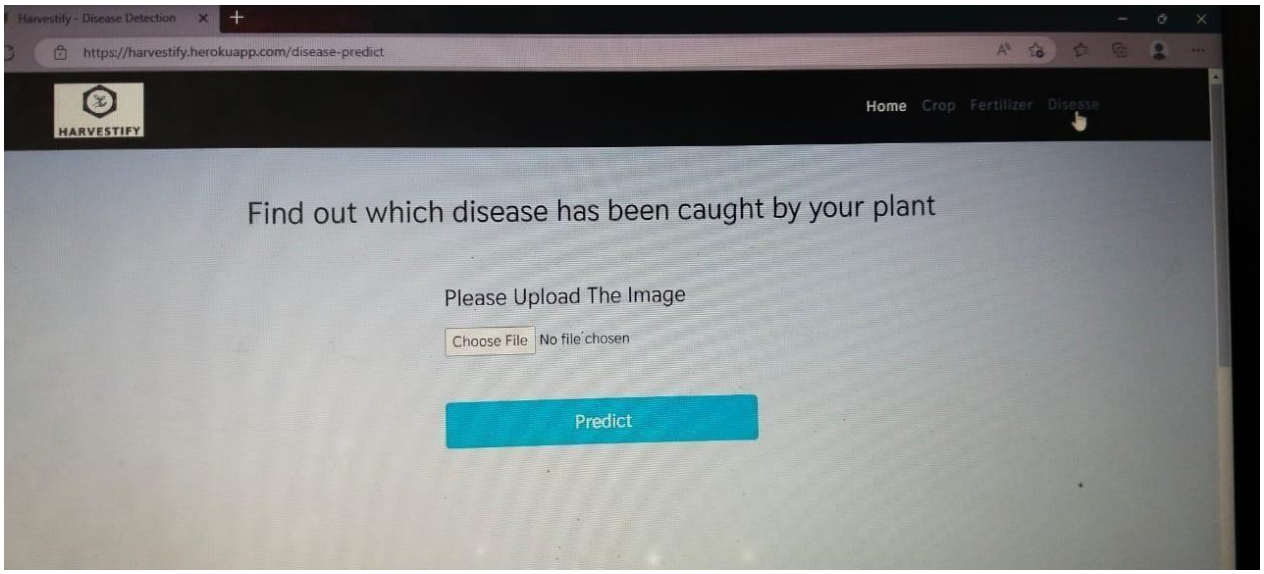
Figure.6.6. Training Vegetable Dataset in IBM Cloud

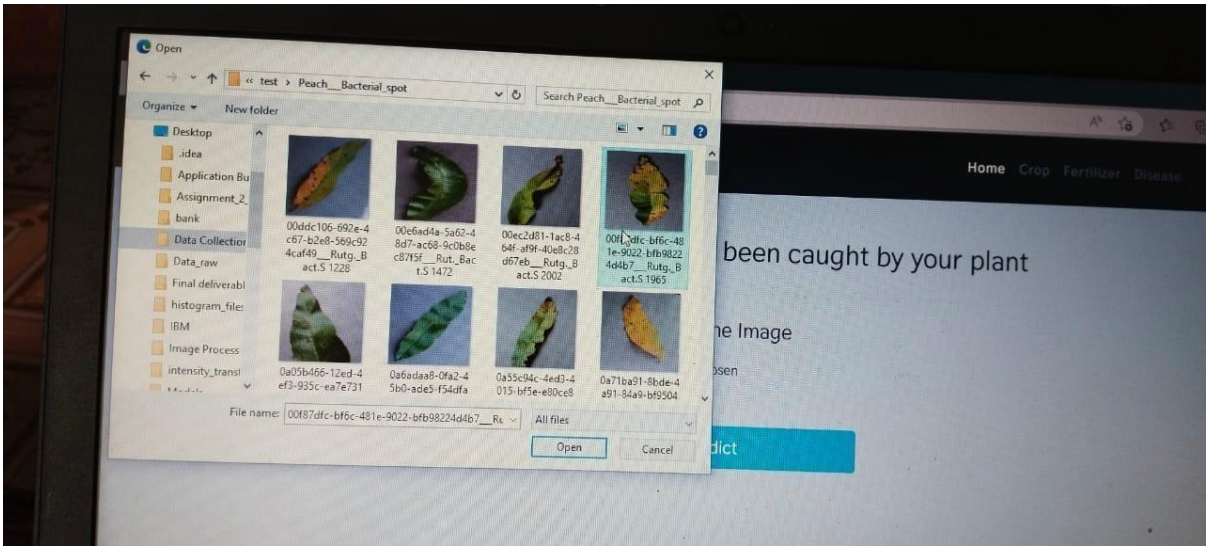
Out put

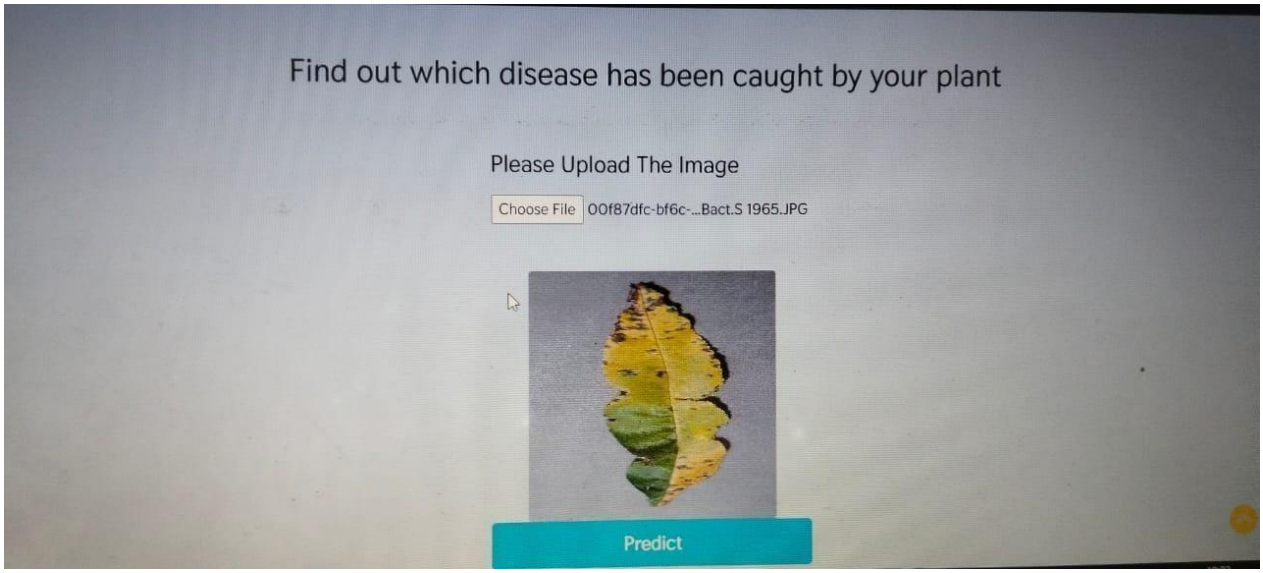


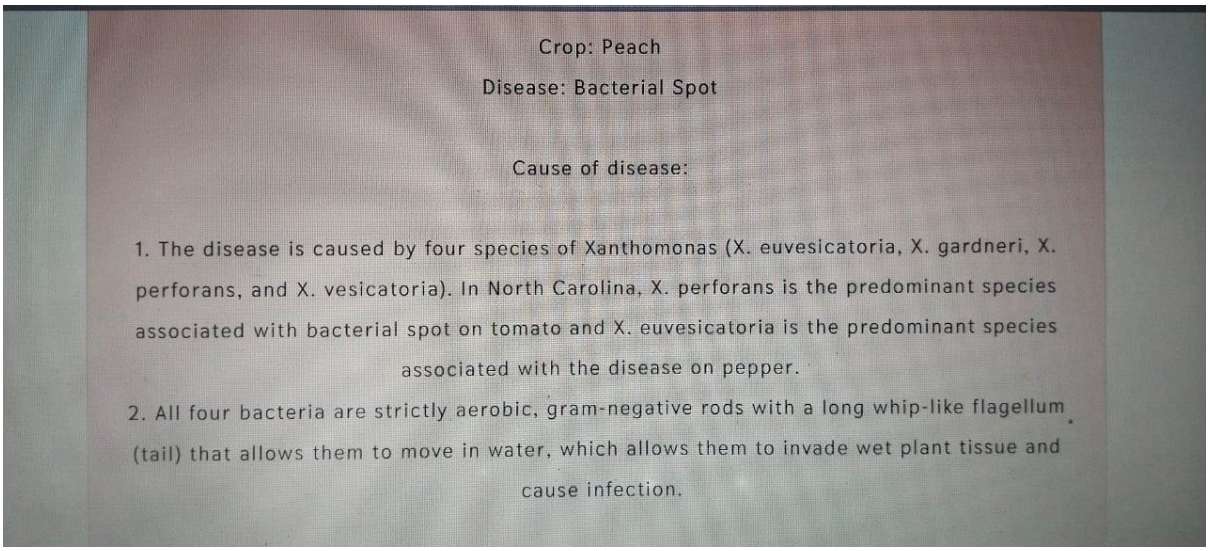

https://harvestify.herokuapp.com/crop-recommend
Enter the value (example:50)
Phosphorous
Enter the value (example:50)
Pottasium
Enter the value (example:50)
ph level
Enter the value
Rainfall (in mm)
Enter the value
State
Select State
City
Predict

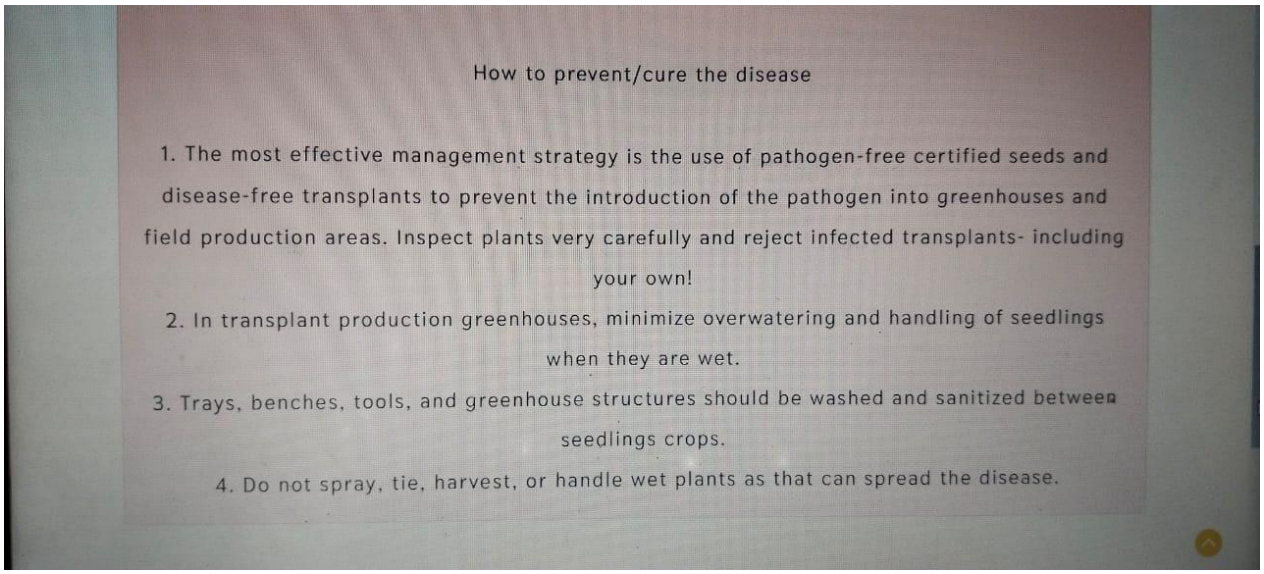

HARVESTIFY
Home
Crop
Fertilizer
Disease
Get informed advice on fertilizer based on soil
Nitrogen
Enter the value (example:50)
Phosphorous
Enter the value (example:50)
Pottasium
Enter the value (example:50)
Crop you want to grow
Select crop
Predict


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


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


Find out which disease has been caught by your plant
Please Upload The Image
Choose File
00f87dfc-bf6c-...Bact.S 1965.JPG
Predict


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



7. ADVANTAGES & DISADVANTAGES

List of advantages

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

List of disadvantages

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

8. APPLICATIONS

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

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

9. CONCLUSIONS

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

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

10. FUTURE SCOPE

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

11. BIBILOGRAPHY

[1]. R Indumathi Leaf Disease Detection and Fertilizer Suggestion", IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 29-30 March 2019, DOI: 10.1109/ICSCAN.2019.8878781.

[2]. P. Pandi Selvi, P. Poornima, "Soil Based Fertilizer Recommendation System forCrop Disease Prediction System", International Journal of Engineering Trends and Applications (IJETA) – Volume 8 Issue 2, Mar-Apr 2021.

[3]. H Shiva reddy, Ganesh hedge, Prof. DR Chinnaya3, "IoT based Leaf Disease Detection and Fertilizer Recommendation", International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 11, Nov 2019, e-ISSN: 2395- 0056.

APPENDIX

A. Source Code (Jupyter notebook python code)

fruit.ipynb (due to limited page size the code vegetable.ipynb uploaded in github)#!/usr/bin/env python

coding: utf-8# In[1]:

pwd

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

Apply ImageDataGenerator functionality to Train and Test set

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

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

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

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

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

Initializing the models 10# In[8]: model=Sequential() # # Add CNN Layers

In[9]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu')) # In[10]: x_train.class_indices

Add Pooling layer

In[11]: model.add(MaxPool2D(pool_size=(2,2))) # # Add Flatten layer # In[12]: model.add(Flatten())# # Add Dense Layer

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

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

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

In[27]: 5384/32

Fit the Model

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

Save the Model

In[29]: model.save("fruit.h5")# In[30]: ls

Test the Model

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

In[33]: model = load_model("fruit.h5")# # Test Apple_Healthy Class images

In[37]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset PlantDisease/fruitdataset/fruit-dataset/test/Apple _____healthy/00fca0da-2db3-481b- b98a9b67bb7b105c_____RS_HL 7708.JPG',target_size=(128,128)) 11

In[39]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0)# In[40]: pred = model.predict_classes(x)

In[41]: pred

In[45]: index =['Apple_____Black_rot','Apple_____healthy','Corn_(maize)_____Northern_Leaf_Blight','Corn_(maize)_____healthy','Peach_____Bacterial_spot','Peach_____healthy']

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

Test Apple Black Rot class images # In[54]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset PlantDisease/fruitdataset/fruit-dataset/test/Apple____Black_rot/0f3d45f4-e121-42cd- a5b6- be2f866a0574_____JR_FrgE.S 2870.JPG',target_size=(128,128))

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

Test Corn Northern leaf Blight class images

In[56]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset PlantDisease/fruitdataset/fruit-dataset/test/Corn_(maize)_Northern_Leaf_Blight/00a14441-7a62- 4034-bc40-b196aeab2785_RS_NLB 3932.JPG',target_size=(128,128))

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

Test Corn Healthy class images

In[58]: img =image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Corn_(maize)____healthy/0a68ef5a-027c- 41ae-b227-159dae77d3dd_____R.S_HL 7969 copy.jpg',target_size=(128,128))

In[59]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]]) # # Test Peach Bacterial spot class images # In[60]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Peach_____Bacterial_spot/00ddc106-692e- 4c67-b2e8- 569c924caf49_____Rutg._Bact.S 1228.JPG',target_size=(128,128)) 12# In[61]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred = model.predict_classes(x) print('the given image belogs to=',index[pred[0]])

Test Peach Healthy class images

In[62]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruitdataset/fruit-dataset/test/Peach_____healthy/1a07ce54-f4fd-41cf- b088- 144f6bf71859_____Rutg._HL 3543.JPG',target_size=(128,128))

In[63]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred =model.predict_classes(x) print('the given image belogs to=',index[pred[0]])

FINAL CODE:

Index.html:

```
<!DOCTYPE html>

<html lang="en">
    <head>

        <!-- basic -->
        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <!-- mobile metas -->

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

        <meta name="viewport" content="initial-scale=1, maximum-scale=1">
        <!-- site metas -->

        <title>IBM</title>
        <meta name="keywords" content="">

        <meta name="description" content="">
        <meta name="author" content="">

        <!-- fevicon -->

<link rel="icon" href="C:\Users\uma25\project\flask\static\images\fevicon.png"type="image/gif" />

        <!-- bootstrap css -->
        <link rel="stylesheet" href="C:\Users\uma25\project\flask\static\css\bootstrap.min.css">

        <!-- style css -->

        <link rel="stylesheet" href="C:\Users\uma25\project\flask\static\css\style.css">

        <!-- Responsive-->
        <link rel="stylesheet" href="C:\Users\uma25\project\flask\static\css\responsive.css">
```

```
<!-- Scrollbar Custom CSS -->
<link rel="stylesheet" href="C:\Users\uma25\project\flask\static\css\jquery.mCustomScrollbar.min.css ">

<!-- Tweaks for older IEs-->

<link rel="stylesheet" href="https://netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-
awesome.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.mi n.css"
media="screen">

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script><![endif]-
->
</head>

<!-- body -->
<body class="main-layout">

<!-- loader  -->

<div class="loader_bg">
<div class="loader"><img src="C:\Users\uma25\project\flask\static\images\loading.gif" alt="#" /></div>

</div>

<!-- end loader -->

<!-- header -->

<header>

<!-- header inner -->

<div class="header-top">
<div class="header">
```

```
<div class="container-fluid">

<div class="row">
<div class="col-xl-2 col-lg-4 col-md-4 col-sm-3 col logo_section">

<div class="full">
<div class="center-desk">

<div class="logo">

<a href="index.html"><img src="C:\Users\uma25\project\flask\static\images\logo.png" alt="#" /></a>

</div>

</div>

</div>

</div>
<div class="col-xl-10 col-lg-8 col-md-8 col-sm-9">

<div class="menu-area">
<div class="limit-box">

<nav class="main-menu ">

<ul class="menu-area-main">
<li class="active"> <a href="index.html">Home</a> </li>

<li> <a href="#about">About</a> </li>

<li> <a href="predict.html">Predict</a> </li>

</ul>
</nav>

</div>

</div>

</div>

</div>
</div>

</div>
```

```
</div>

<!-- end header inner -->
<!-- end header -->

<section class="slider_section">
<div id="myCarousel" class="carousel slide" data-ride="carousel">

<ol class="carousel-indicators">
<li data-target="#myCarousel" data-slide-to="0" class="active"></li>

<li data-target="#myCarousel" data-slide-to="1"></li>
</ol>

<div class="carousel-inner">

<div class="carousel-item active">
<div class="container-fluid padding_dd">

<div class="carousel-caption">
<div class="row">

<div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
<div class="text-bg">
<h1 style="font-size: 50px;">Fertilizers Recommendation SystemFor Disease Prediction In Plants</h1>

<form class="Vegetable">

</div>

</div>
<div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">

<div class="images_box">

<figure><img src="C:\Users\uma25\project\flask\static\images\img2.png"></figure>

</div>

</div>
</div>
```

```
</div>

</div>
</div>

<div class="carousel-item">
<div class="container-fluid padding_dd">

<div class="carousel-caption">
<div class="row">

<div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
<div class="text-bg">
<h2 style="font-size: 35px;">It is used to predict diseases in plants and to suggest fertilizers which would be required to eradicate the diseaseand to fight off the disease in the future too.</h2>

</div>

</div>
<div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">

<div class="images_box">

<figure><img src="C:\Users\uma25\project\flask\static\images\img4.gif"></figure>

</div>

</div>
</div>

</div>
</div>

</div>

</div>

</div>

<a class="carousel-control-prev" href="#myCarousel" role="button" data-slide="prev">
```

Previous

Next

</section>

</header>
<!-- about -->

<h2>About us</h2>

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 toensure high quantity and best quality. In recent years, the number of diseases onplants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

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

```
</div>

</div>
<div class="col-xl-6 col-lg-6 col-md-6 col-sm-12 padding_rl">

<div class="about-box_img">

<figure><img src="C:\Users\uma25\project\flask\static\images\about.jpg"alt="#" /></figure>

</div>

</div>
</div>

</div>

</div>

<!-- end abouts -->
<!-- end vegetable -->

<!-- Javascript files-->

<script src="C:\Users\uma25\project\flask\static\js\jquery.min.js"></script>

<script src="C:\Users\uma25\project\flask\static\js\popper.min.js"></script>

<script src="C:\Users\uma25\project\flask\static\js\bootstrap.bundle.min.js"></script>

<script src="C:\Users\uma25\project\flask\static\js\jquery-3.0.0.min.js"></script>

<script src="C:\Users\uma25\project\flask\static\js\plugin.js"></script>
<!-- sidebar -->
<script src="C:\Users\uma25\project\flask\static\js\jquery.mCustomScrollbar.concat.mi n.js"></script>

<script src="C:\Users\uma25\project\flask\static\js\custom.js"></script>
```

```
<script src="https:cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.js "></script>

</body>

</html>
```

Predict.html:

```
<!DOCTYPE html>

<html lang="en">


<head>

<!-- basic -->

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">
<!-- mobile metas -->

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

<!-- site metas -->
<title>IBM</title>

<meta name="keywords" content="">
<meta name="description" content="">

<meta name="author" content="">

<!-- fevicon -->

<link rel="icon" href="C:\Users\uma25\project\flask\static\images\fevicon.png"type="image/gif" />

<!-- bootstrap css -->
<link rel="stylesheet" href="C:\Users\uma25\project\flask\static\css\bootstrap.min.css">
```

```
<!-- style css -->
<link rel="stylesheet" href="C:\Users\uma25\project\flask\static\css\style.css">

<!-- Responsive-->
<link rel="stylesheet" href="C:\Users\uma25\project\flask\static\css\responsive.css">

<!-- Scrollbar Custom CSS -->

<link rel="stylesheet" href="C:\Users\uma25\project\flask\static\css\jquery.mCustomScrollbar.min.css ">

<!-- Tweaks for older IEs-->
<link rel="stylesheet" href="https://netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-
awesome.css">

<link href="{{ url_for('static', filename='css/final.css') }}" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.mi n.css"
media="screen">

<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script><![endif]-
->
</head>

<body>

<!-- predict -->
<div id="predict" class="vegetable">

<div class="container">
<div class="row">

<div class="col-md-12">
<div  class="titlepage">
```

```
<h2> Plant Disease <strong class="llow">Prediction</strong> </h2>

</div>
</div>

</div>
<div class="row">

<div class="col-xl-5 col-lg-5 col-md-5 col-sm-12 ">

<h4 style="color: green;"><b>Drop the image to get theprediction!!!...</b></h4>

<div class="vegetable_shop">
<form action="" id="upload-file" method="post"enctype="multipart/form-
data">

<h5 style="color: red;"><b>Choose Category</b></h5>

<select name="plant">

<option value="fruit" name="fruit">Fruits</option>
<option value="vegetable" name="vegetable">Vegetables</option>

</select>
<br><br>
<input id="imageUpload" type="file" name="image" value="CHOOSE"accept="image/jpeg, image/jpg,
image/png" onchange="document.getElementById('output').src=window.URL.createObject URL(this.files[0])">

<br><br>

<img src="" id="output">

<br><br>

<button type="button" class="btn btn-info btn-lg " id="btn-predict"style="background:
#28272c;">Predict</button>

</form>
<!--drop down,file upload,button-->

</div>
```

```
</div>

<div class="loader" style="display:none;"></div>
                                        <h3>

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

<div class="col-xl-7 col-lg-7 col-md-7 col-sm-12 ">
                                    <div class="vegetable_img">

                                        <right>

<figure><img src="C:\Users\uma25\project\flask\static\images\v1.jpg"alt="#"/></figure>

                                        </right>

                                         </div>
                                        </div>

                                        </div>
                                        </div>

                                        <!-- Javascript files-->

                                    <script src="C:\Users\uma25\project\flask\static\js\jquery.min.js"></script>
                                    <script src="C:\Users\uma25\project\flask\static\js\popper.min.js"></script>
                                <script src="C:\Users\uma25\project\flask\static\js\bootstrap.bundle.min.js"></script>

<script src="C:\Users\uma25\project\flask\static\js\jquery-3.0.0.min.js"></script>

                                    <script src="C:\Users\uma25\project\flask\static\js\plugin.js"></script>

                                    <script src="C:\Users\uma25\project\flask\static\js\main.js"></script>
                                        <!-- sidebar -->

                                <script src="C:\Users\uma25\project\flask\static\js\jquery.mCustomScrollbar.concat.mi n.js"></script>

                                    <script src="C:\Users\uma25\project\flask\static\js\custom.js"></script>
```

<script src="https:cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.js "></script>

</body>

</html>

App.py :

```
import requests
from tensorflow.keras.preprocessing import imagefrom
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_forimport os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session


app=Flask(__name__)


model=load_model('vegetable.h5')model1=load_model('fruit.h5')


#homepage
@app.route('/')def
home():
return render_template('index.html')
```

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


@app.route('/predict',methods=['POST'])def predict():
     if request.method=='POST': f=request.files['image']
     basepath=os.path.dirname(_____file_____)
                    file_path=os.path.join(basepath,'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)

Fertilizers Recommendation System For Disease Prediction

Team ID : PNT2022TMID49029

Project Objectives:

1. Preprocess the images.
2. Applying the CNN algorithm to the dataset.
3. How deep neural networks detect the disease.
4. You will be able to know how to find the accuracy of the model.
5. You will be able to build web applications using the Flask framework.

Project Flow :

- A web Application is built where;
  - Farmers can interact with the portal build
  - Interacts with the user interface to upload images of diseased leaf
  - Our model built analyses the Disease and suggests the farmer with fertilizers areto be used
- To accomplish the above task you must complete the below activities and tasks;

- Download the dataset.
- Classify the dataset into train and test sets.
- Add the neural network layers.
- Load the trained images and fit the model.
- Test the model.
- Save the model and its dependencies.
- Build a Web application using a flask that integrates with the model built.

Prior Knowledge :

- **Supervised and Unsupervised learning:**
  - Supervised learning, as the name indicates, has the presence of a supervisor as a teacher. Basically supervised learning is when we teach or train the machine using data that is welllabelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of data.so that the supervised learning algorithm analyses the training data and produces a correct outcome from labelled data.
  - Unsupervised learning is the training of a machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.
- **Regression Classification and Clustering:**
  - Clustering is an unsupervised technique. With clustering, the algorithm tries to find a pattern in data sets without labels associated with it. This could be a clustering of buying behaviour of customers. Features for this would be the household income, age, … and clusters of different consumers could then be built.
  - Classification algorithms look at existing data and predicts what a new data belongs to. Classification is used for spam for years now and these algorithms are more or less maturein classifying something as spam or not. With machine data, it could be used to predict a material quality by several known parameters (e.g. humidity, strength, color, …).
- **Artificial Neural Networks :**
  - To understand the concept of the architecture of an artificial neural network, we have to understand what a neural network consists of. In order to define a neural network that consists of a large number of artificial neurons, which are termed unitsarranged in a sequence of layers.Lets us look at various types of layers available inan artificial neural network.
    - Input Layer
    - Hidden Layer
    - Output layer

- **Convolution Neural Networks :**
  - A convolutional neural network, or CNN, is a deep learning neural networksketchedfor processing structured arrays of data such as portrayals.
  - CNN are very satisfactory at picking up on design in the input image, such aslines, gradients, circles, or even eyes and faces.
  - CNN can run directly on a underdone image and do not need anypreprocessing.
- **Flask:**
  - To put it to use in order to predict the new data, we have to deploy it over the internet so that the outside world can use it. In this article, we will talk about how we have trained a machine learning model and created a web applicationon it using Flask.

Prerequisites :

Anaconda Navigator :

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

To build Deep learning models you must require the following packages :

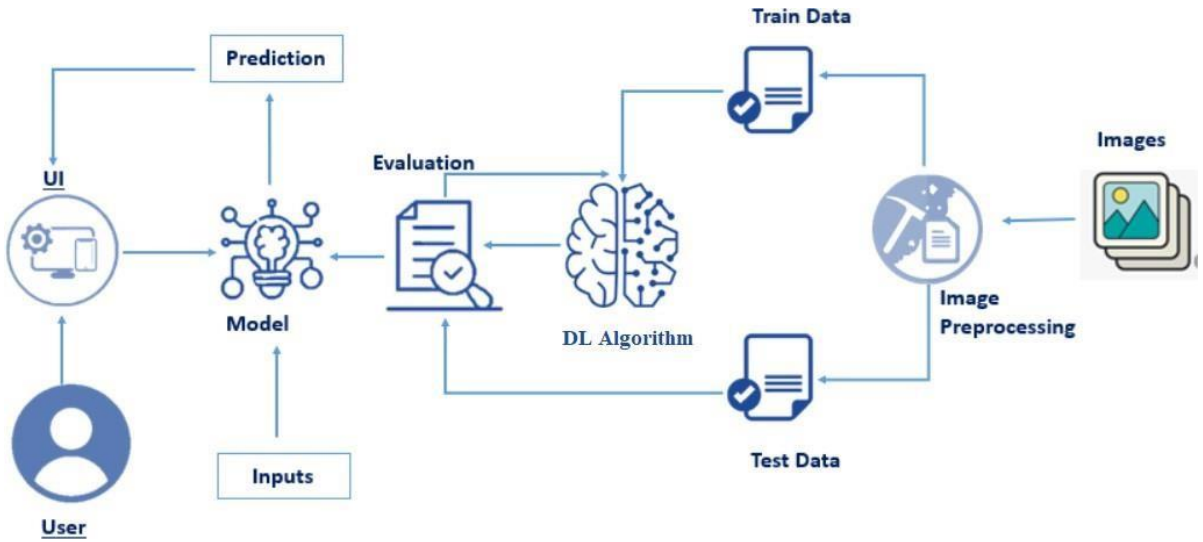
- **Tensor flow:** TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researcherspush the state-of-the-art in ML and developers can easily build anddeploy ML powered applications.

- **Keras :** Keras leverages various optimization techniques to make high level neuralnetworkAPI easier and more performant. It supports the following features.

1. Consistent, simple and extensible API.
2. Minimal structure - easy to achieve the result without any frills.
3. It supports multiple platforms and backend.
4. It is user-friendly framework that runs on both CPU and GPU.
5. Highly scalability of computation.

- **Flask:** Web framework used for building Web applications.

Technical Architecture :



DATASET

DATE	10 November 2022

TEAM ID	PNT2022TMID49029
PROJECT NAME	Fertilizer Recommendation System ForDisease Prediction
MAXIMUM MARK	4 Marks

Dataset:

https://drive.google.com/file/d/1fxs7ptI6zh7NTbCOZARKZ7AmYKjnprrY/view?usp=sharing

IBM PROJECT

Fertilizers Recommendation System For Disease Prediction

ABSTRACT

✓Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. 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.

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

EXISTING PROBLEM

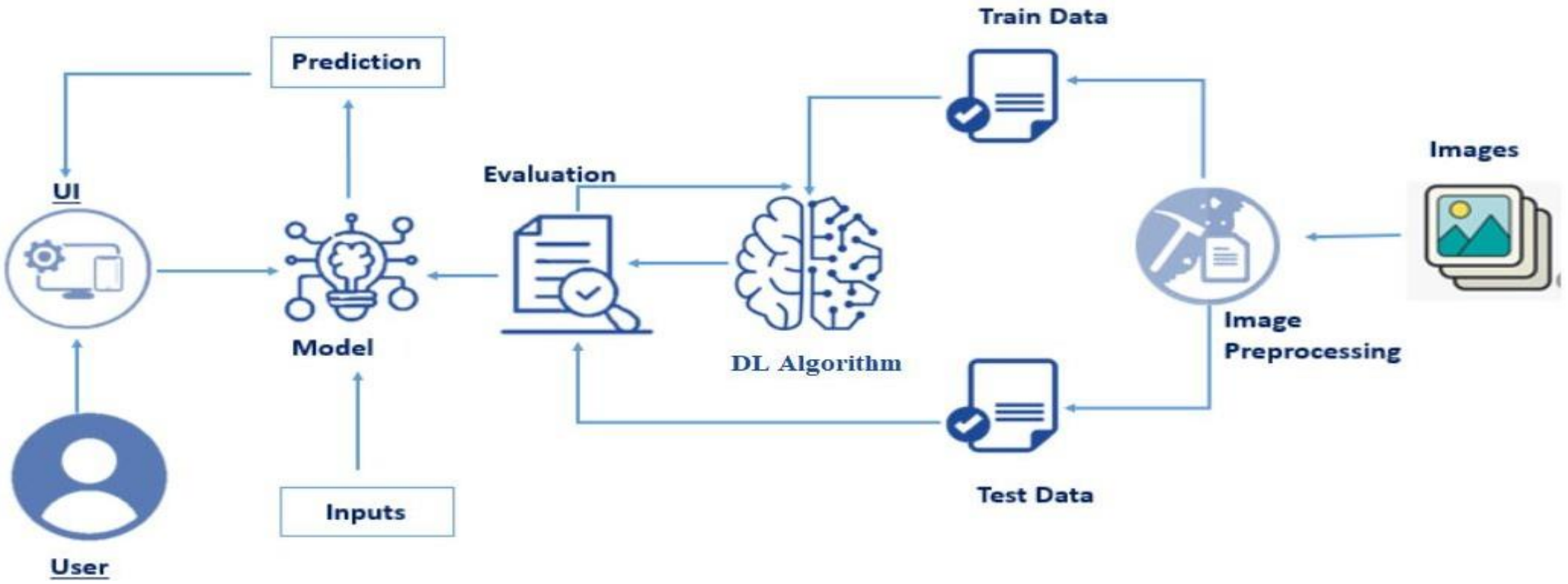
- ✓Adequate mineral nutrition is central to crop production. However, it can also exert considerable Influence on disease development. Fertilizer application can increase or decrease development of diseases caused by different pathogens, and the mechanisms responsible are complex, including effects of nutrients on plant growth, plant resistance mechanisms and direct effects on the pathogen. The effects of mineral nutrition on plant disease and the mechanisms responsible for those effects have been dealt with comprehensively elsewhere. In India, around 40% of land is kept and grown using reliable irrigation technologies, while the rest relies on the monsoon environment for water. Irrigation decreases reliance on the monsoon, increases food security, and boosts agricultural production.
- ✓Most research articles use humidity, moisture, and temperature sensors near the plant's root, with an external device handling all of the data provided by the sensors and transmitting it directly to an external display or an Android application. The application was created to measure the approximate values of temperature, humidity and moisture sensors that were programmed into a microcontroller to manage the amount of water.

PROPOSED SYSTEM

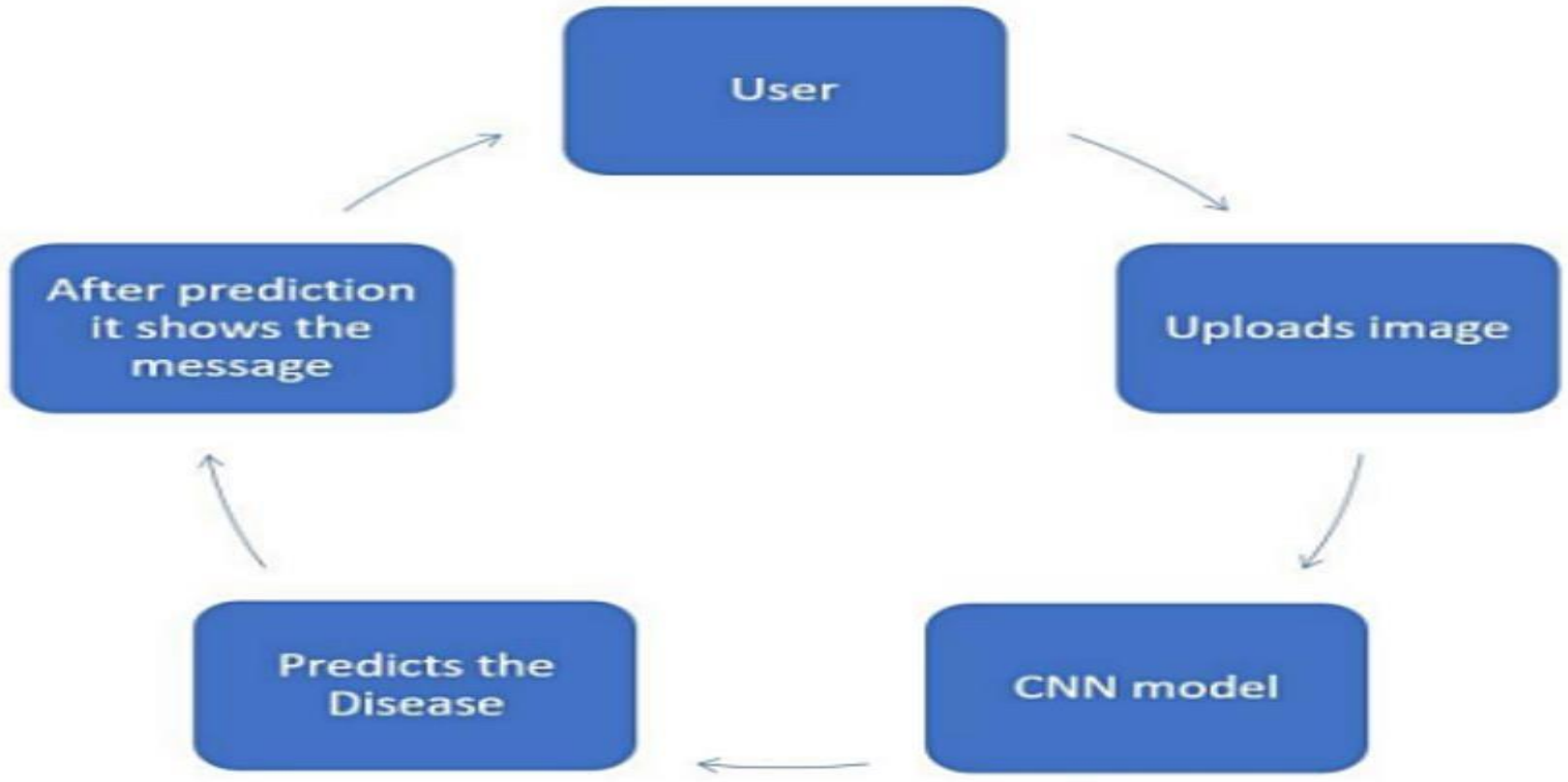
So we have built Web Application where:

- Farmers can easily interact with the application that we have built.
- This web application provides the user interface to upload images of diseased crop or leaf.
- Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest.
- Our web application analysis the disease of the crop or leaf and suggests the farmer with the fertilizers are to be used.
- It recommends the fertilizer for affected leaves based on severity level.
- This web application makes the farmers to take right decision in selecting the fertilizer for crop disease such that agricultural sector will be developed by innovative idea.

TECHNICAL ARCHITECTURE



FLOW CHART


User
Uploads image
CNN model
Predicts the Disease
After prediction it shows the message

HARDWARE & SOFTWARE REQUIREMENT

✓ To complete this project, you should have the following software and packages.

Software's:

➢ Anaconda Navigator

➢ py charm

➢ Visual studio code

➢ Jupiter notebook

➢ IBM Watson studio

Packages:

➢ Tensor flow

➢ Keras

➢ Flask

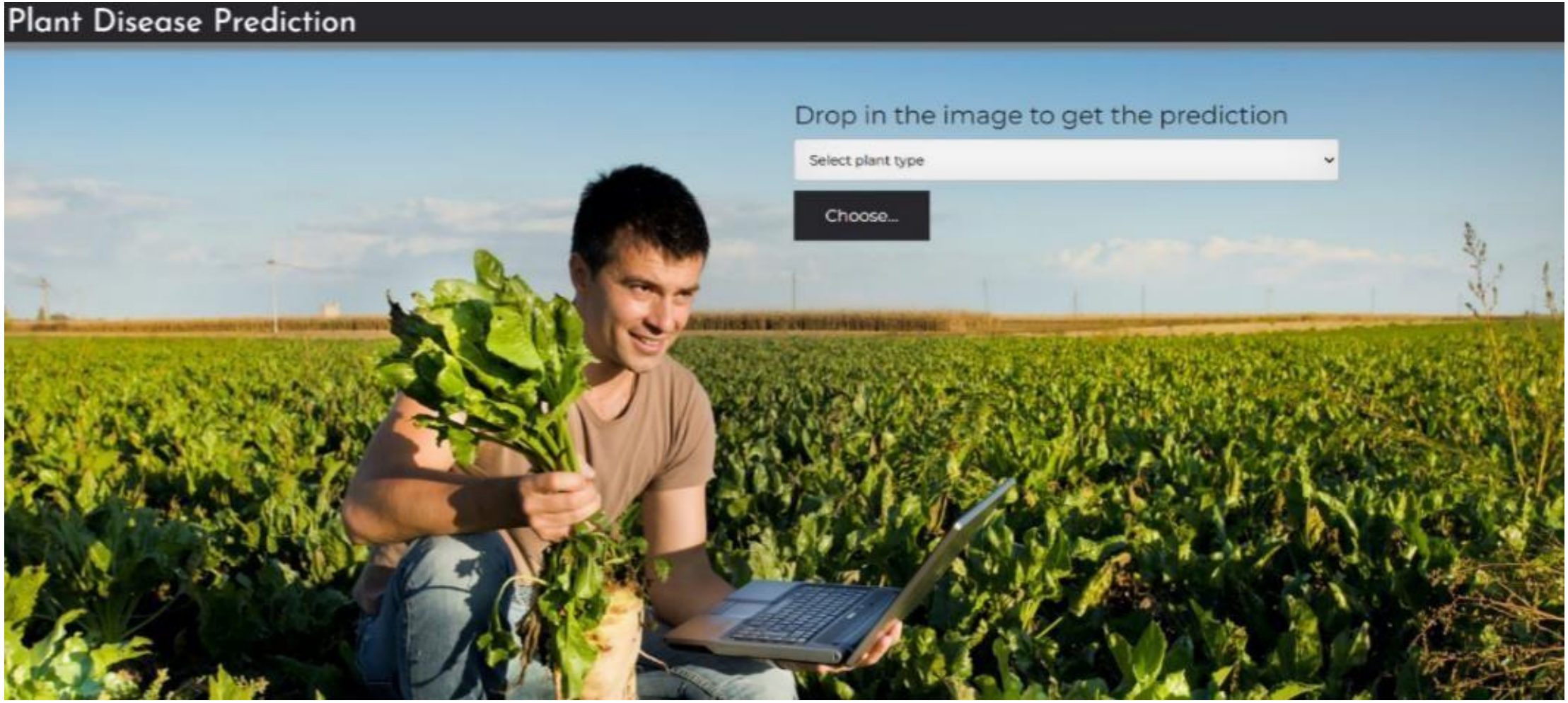
➢ numpy

➢ Pandas

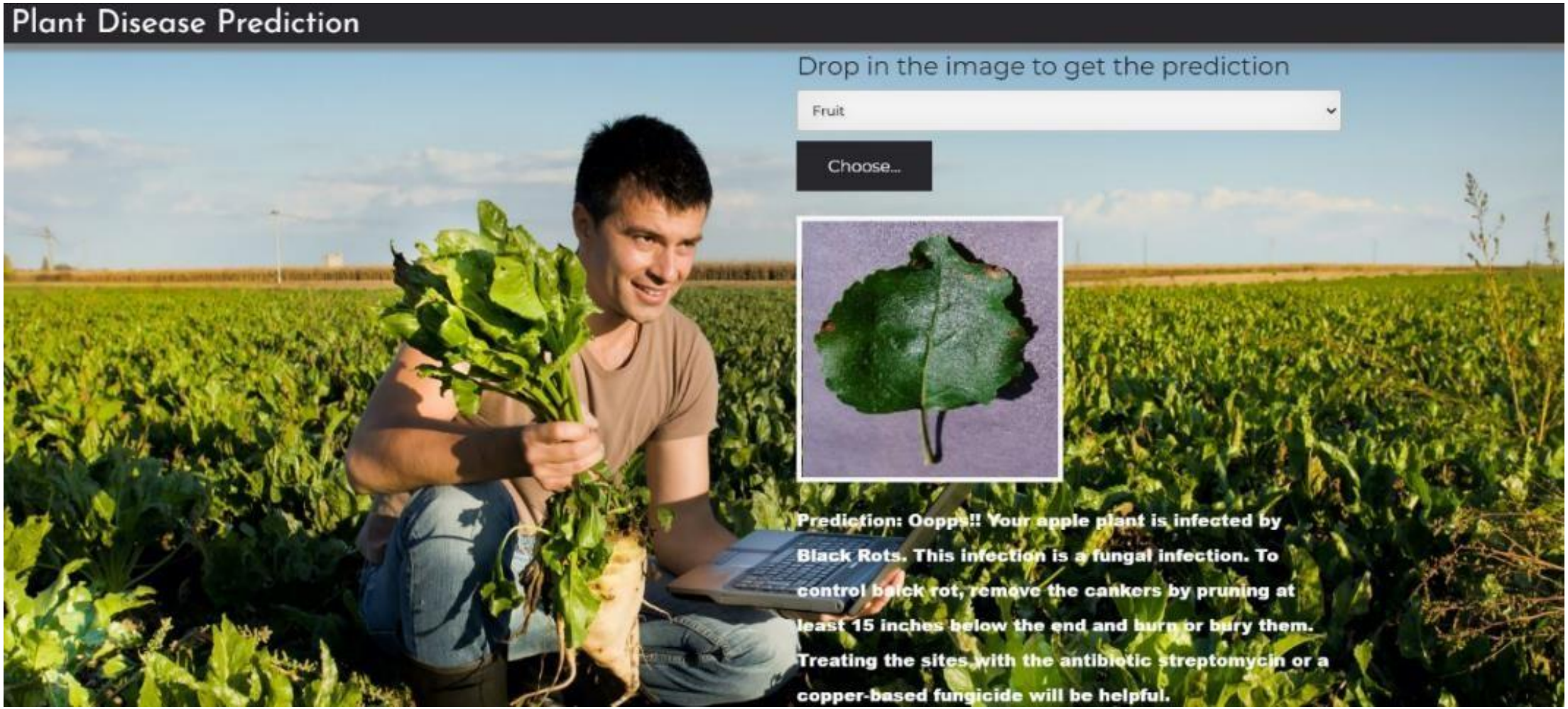
OUTPUT


Plant Disease Prediction
Home
Predict
Fertilizers Recommendation System
For Disease Prediction!!
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 labour costs.

OUTPUT


Plant Disease Prediction
Drop in the image to get the prediction
Select plant type
Choose...

OUTPUT


Plant Disease Prediction
Drop in the image to get the prediction
Fruit
Choose...
Prediction: Oopps!! Your apple plant is infected by Black Rots. This infection is a fungal infection. To control balck rot, remove the cankers by pruning at least 15 inches below the end and burn or bury them. Treating the sites with the antibiotic streptomycin or a copper-based fungicide will be helpful.

APPLICATION

- This web application can be used by farmers or users to check whether their plant is infected or not and can also show the remedy so that the user can take necessary precautions.
- These kind of web applications can be used in the agricultural sector as well as for small house hold plants as well.

Thank you

Presentation title

PROJECT REPORT

Fertilizers Recommendation System for DiseasePrediction

Team ID: PNT2022TMID49029

TEAM MEMBERS:

P.NAGAJOTHI (921619106041)
M.BHARATHI PRIYA (921619106009)
T.DIVYADHARSHINI(921619106018)
S.YUVASHREE(921619106067)

ABSTRACT:

✓ 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.

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

Presentation title

Purpose:

✓ To Detect and recognize the plant diseases and to recommend fertilizer, it is necessary to provide symptoms in identifying the disease at its earliest.

Hence the authors proposed and implemented new fertilizers Recommendation System for crop disease prediction.

Presentation title

LITERATURE SURVEY:

Existing Problem:

- ✓ Adequate mineral nutrition is central to crop production. However, it can also exert considerable Influence on disease development. Fertilizer application can increase or decrease development of diseases caused by different pathogens, and the mechanisms responsible are complex, including effects of nutrients on plant growth, plant resistance mechanisms and direct effects on the pathogen. The effects of mineral nutrition on plant disease and the mechanisms responsible for those effects have been dealt with comprehensively elsewhere. In India, around 40% of land is kept and grown using reliable irrigation technologies, while the rest relies on the monsoon environment for water. Irrigation decreases reliance on the monsoon, increases food security, and boosts agricultural production.
- ✓ Most research articles use humidity, moisture, and temperature sensors near the plant's root, with an external device handling all of the data provided by the sensors and transmitting it directly to an external display or an Android application. The application was created to measure the approximate values of temperature, humidity and moisture sensors that were programmed into a microcontroller to manage the amount of water.

Presentation title

Proposed solution:

So, we have built Web Application where:

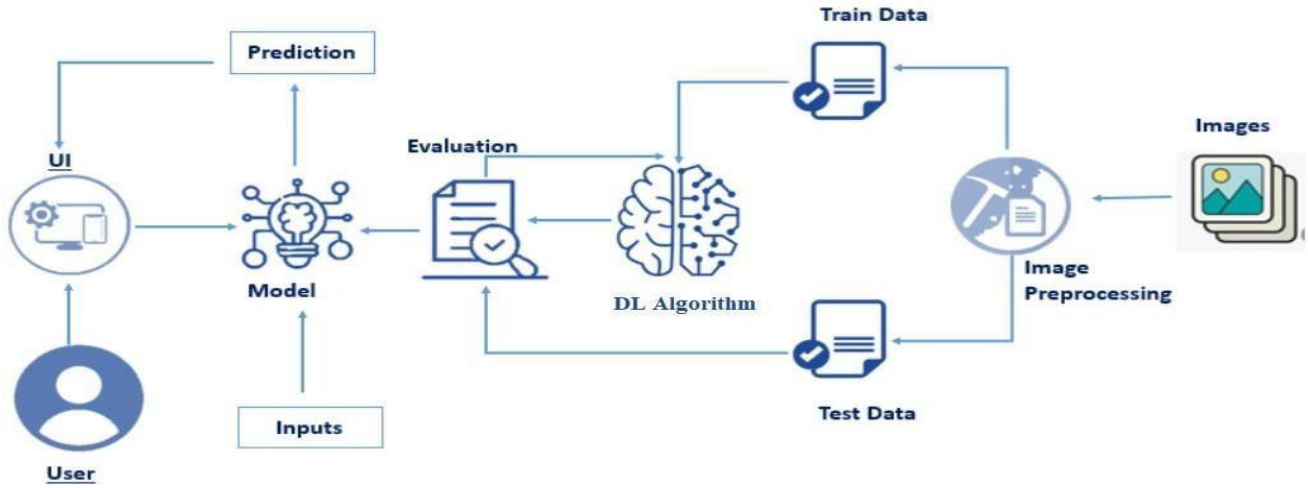
1. Farmers interact with the portal build
2.Interacts with the user interface to upload images of diseased leaf
3.Our model built analyses the Disease and suggests the farmer with fertilizers are to be used

- ➢ Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest.
  - ➢ It recommends the fertilizer for affected leaves based on severity level.
- ➢ This web application makes the farmers to take right decision in selecting

the fertilizer for crop disease such that agricultural sector will be developed by innovative idea.

Presentation title

TECHNICAL ARCHITECTURE:



HARDWARE / SOFTWARE REQUIREMENTS:

To complete this project, you should have the following software and packages.

Software's:

- Anaconda Navigator
- py charm
- Visual studio code
- Jupiter notebook
- IBM Watson studio

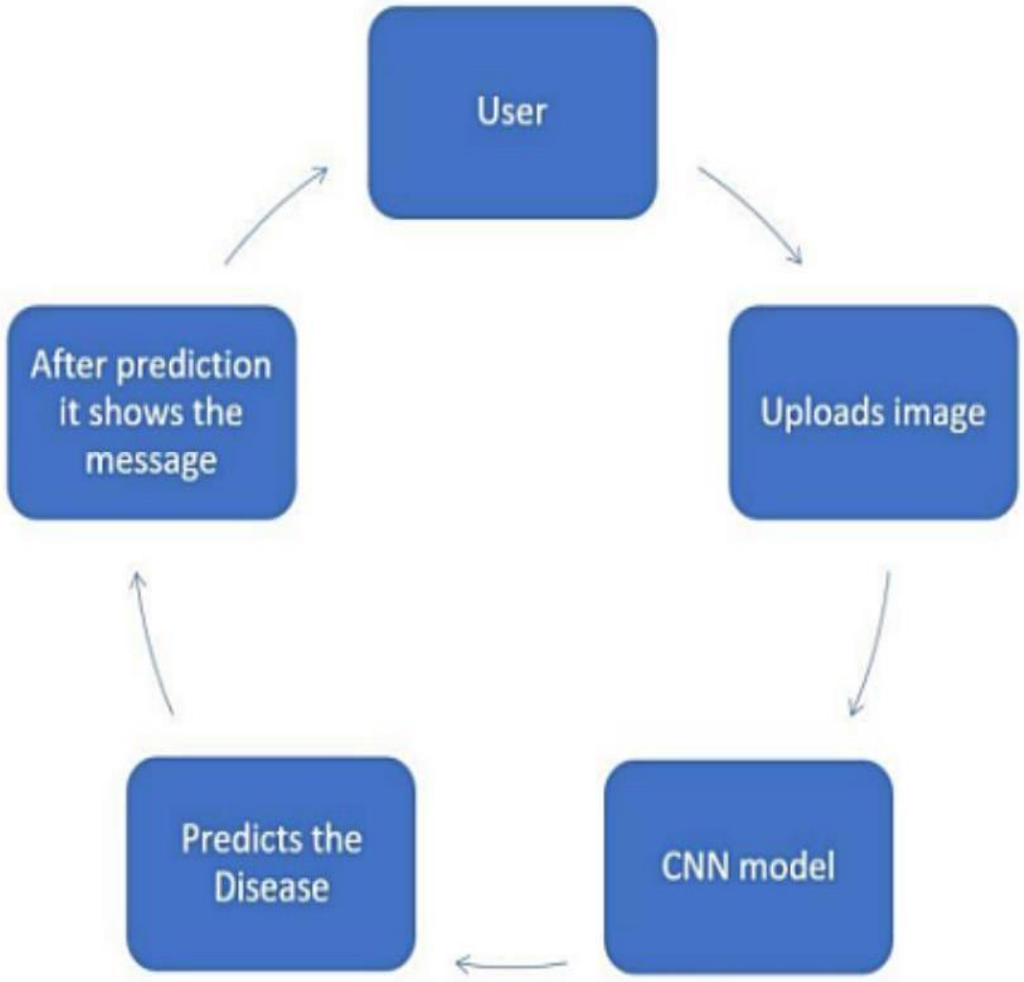
Presentation title

Packages:

- Tensor flow
- Keras
- Flask
- numpy
- Pandas

By using the above listed software's and packages, we built this application to take the input as image from the Farmer and detects whether theplant is infected or not. Here we use Deep learning techniques and give the output to the user as Farmer.

FLOWCHART:



To accomplish the above task, you must complete the below activities and tasks:

- Download the dataset.
- Classify the dataset into train and test sets.
- Add the neural network layers.
- Load the trained images and fit the model.

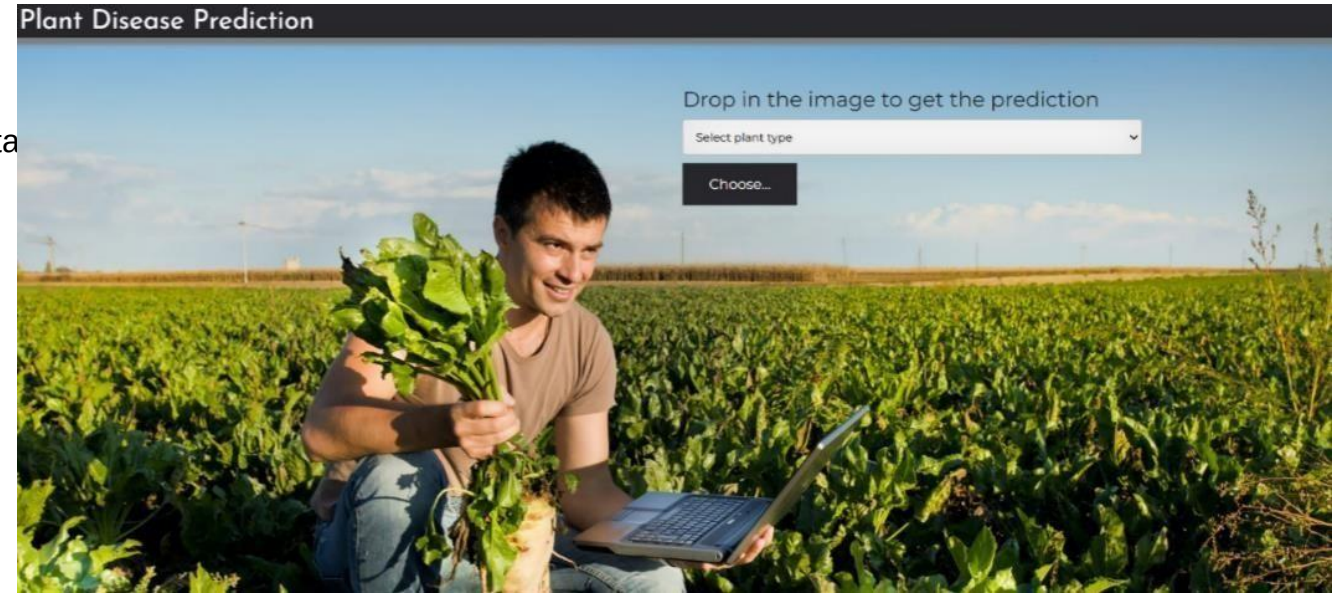
- Test the model.
- Save the model and its dependencies.
- Build a Web application using a flask that integrates with the model built.

OUTPUT:

Home Page:

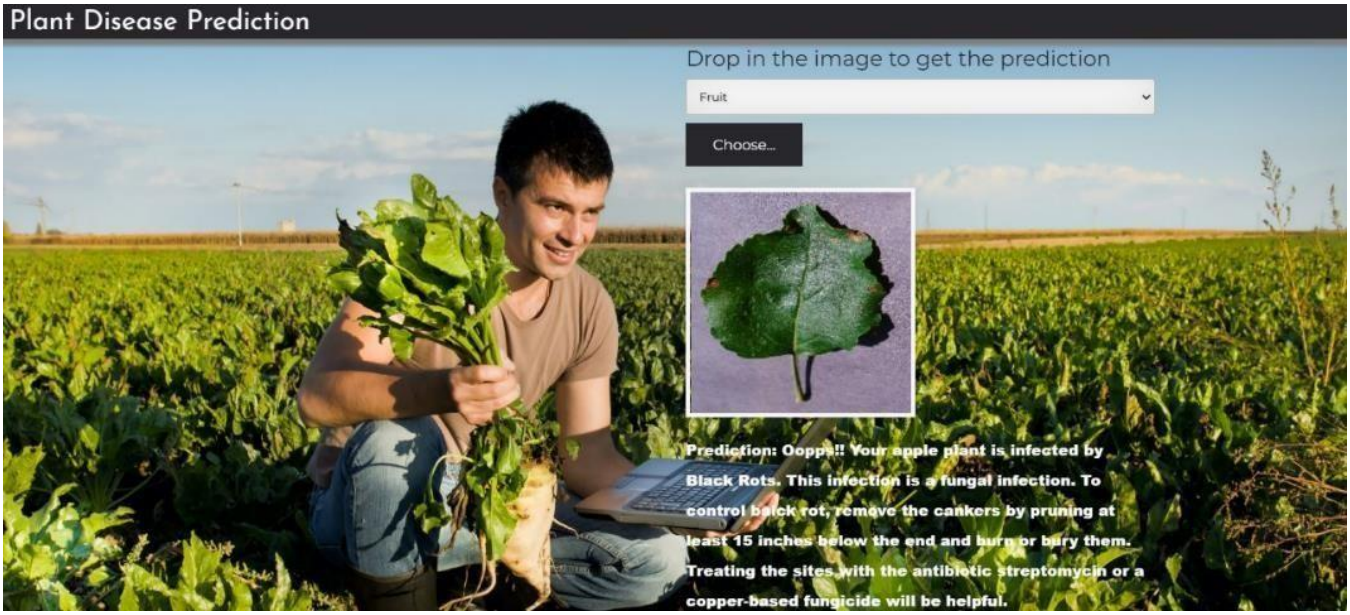
Prediction Page:

Presenta

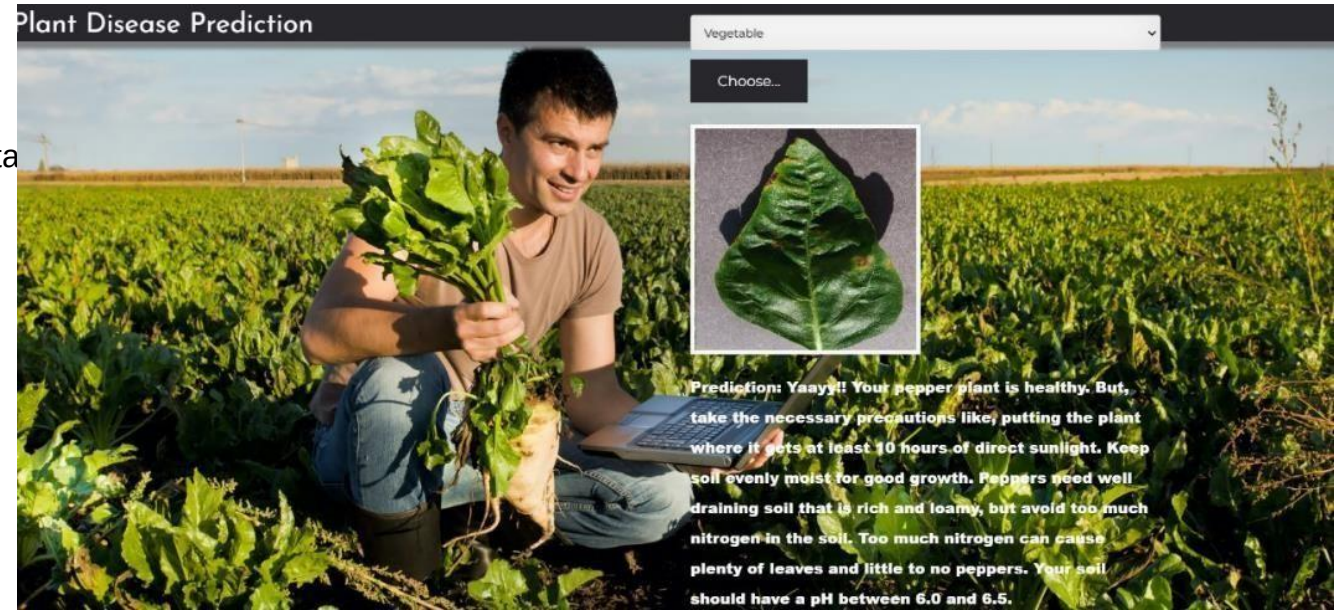


Result Page:

⇨ **Fruit:**



⇨ **Vegetable:**



Presenta

ADVANTAGES:

- The proposed model could predict the disease just from the image of a part icular plant.
- Easy to use UI.
- Model has some good accuracy in detecting the plant just by taking the input(leaf).

APPLICATIONS:

- This web application can be used by farmers or users to check whether their plant is infected or not and can also show the remedy so that the user can take necessary precautions.
- These kind of web applications can be used in the agricultural sector as well as for small house hold plants as well.

CONCLUSION:

- 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. Usage of such applications could help the farmers to necessary precautions so that they don't face any loss as such.

FUTURE SCOPE:

- As of now we have just built the web application which apparently takes the input as an image and then predict the out in the near future we can develop an application which computer vision and AI techniques to predict the

infection once you keep the camera near the plant or leaf this could make our project even more usable.

Presentation title

- This can be also done in Mobile applications like android, ios. It helps in many ways to improve the agriculture in cultivation of crops and predict the correct fertilizers to the crops.

APPENDIX:

Source Code:

```
In [1]: ls

In [2]: pwd
Out[2]: '/home/wsuser/work'

In [8]: !pip install keras==2.7.0
        !pip install tensorflow==2.5.0

Collecting keras==2.7.0
  Using cached keras-2.7.0-py2.py3-none-any.whl (1.3 MB)
Installing collected packages: keras
  Attempting uninstall: keras
    Found existing installation: Keras 2.2.4
    Uninstalling Keras-2.2.4:
      Successfully uninstalled Keras-2.2.4
Successfully installed keras-2.7.0
Requirement already satisfied: tensorflow==2.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (2.5.0)
Requirement already satisfied: protobuf>=3.9.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.
0) (3.19.1)
Requirement already satisfied: h5py~=3.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0)
(3.1.0)
Requirement already satisfied: astunparse~=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.
5.0) (1.6.3)
Requirement already satisfied: keras-nightly~=2.5.0.dev in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorf
low==2.5.0) (2.5.0.dev2021032900)
Requirement already satisfied: termcolor~=1.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.
5.0) (1.1.0)
Requirement already satisfied: flatbuffers~=1.12.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==
2.5.0) (1.12)
Requirement already satisfied: wrapt~=1.12.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0)
(1.12.1)
Requirement already satisfied: six~=1.15.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0)
```

Presenta

```
Requirement already satisfied: typing-extensions~=3.7.4 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorf
low==2.5.0) (3.7.4.3)
Requirement already satisfied: keras-preprocessing~=1.1.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tenso
rflow==2.5.0) (1.1.2)
Requirement already satisfied: absl-py~=0.10 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0)
(0.12.0)
Requirement already satisfied: grpcio~=1.34.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.
0) (1.34.1)
Requirement already satisfied: numpy~=1.19.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0)
(1.19.5)
Requirement already satisfied: google-pasta~=0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.
5.0) (0.2.0)
Requirement already satisfied: wheel~=0.35 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0)
(0.37.0)
Requirement already satisfied: opt-einsum~=3.3.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.
5.0) (3.3.0)
Requirement already satisfied: gast==0.4.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.5.0)
(0.4.0)
Requirement already satisfied: tensorboard~=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorflow==2.
5.0) (2.7.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboar
d~=2.5->tensorflow==2.5.0) (1.23.0)
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=2.5
->tensorflow==2.5.0) (3.3.3)
Requirement already satisfied: werkzeug>=0.11.15 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=
2.5->tensorflow==2.5.0) (2.0.2)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from tensorboard~=2.5->tensorflow==2.5.0) (0.6.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from te
nsorboard~=2.5->tensorflow==2.5.0) (1.6.0)
Requirement already satisfied: setuptools>=41.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~=
2.5->tensorflow==2.5.0) (58.0.4)
Requirement already satisfied: requests<3,>=2.21.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from tensorboard~
```

Image Augmentation

In [9]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [10]:
```
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
```

In [11]:
```
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [12]:
```
ls
```

In [13]:
```
pwd
```

Out[13]:
```
'/home/wsuser/work'
```

Presenta

In [14]:
```
import os, types
import pandas as pd
from botocore.client import Config
import 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='fru

# 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 possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [15]:
```
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_1.read()), "r")
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [16]:
```
pwd
```

Out[16]:
```
'/home/wsuser/work'
```

In [17]:
```
import os
filenames = os.listdir('/home/wsuser/work/fruit-dataset/train')
```

In [18]:
```
x_train=train_datagen.flow_from_directory("/home/wsuser/work/fruit-dataset/train",target_size=(128,128),class_mode='categorical',
```

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

In [ ]:
```

```

In [19]:
```
x_test=test_datagen.flow_from_directory(r"/home/wsuser/work/fruit-dataset/test",target_size=(128,128),
                                        class_mode='categorical',batch_size=24)
```

Presentatio

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

```
In [20]: x_train.class_indices
```

```
Out[20]: {'Apple___Black_rot': 0,
          'Apple___healthy': 1,
          'Corn_(maize)___Northern_Leaf_Blight': 2,
          'Corn_(maize)___healthy': 3,
          'Peach___Bacterial_spot': 4,
          'Peach___healthy': 5}
```

CNN

```
In [21]: from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
```

```
In [24]: model=Sequential()
```

```
In [25]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

```
In [26]: model.add(MaxPooling2D(pool_size=(2,2)))
```

```
In [27]: model.add(Flatten())
```

```
In [28]: 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
_________________________________________________________________
```

Presentati

```
In [29]: 32*(3*3*3+1)
```

```
Out[29]: 896
```

Hidden Layers

```
In [30]: model.add(Dense(300,activation='relu'))
         model.add(Dense(150,activation='relu'))
```

Output Layer

```
In [31]: model.add(Dense(6,activation='softmax'))
```

```
In [32]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
In [33]: len(x_train)
Out[33]: 225

In [34]: 1238/24
Out[34]: 51.583333333333336

In [35]: model.fit_generator(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_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 ver
sion. 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:
0.9620
Epoch 8/10
225/225 [==============================] - 116s 514ms/step - loss: 0.1282 - accuracy: 0.9597 - val_loss: 0.1194 - val_accuracy:
0.9508
Epoch 9/10
225/225 [==============================] - 116s 514ms/step - loss: 0.1141 - accuracy: 0.9616 - val_loss: 0.1478 - val_accuracy:
0.9751
Epoch 10/10
225/225 [==============================] - 116s 516ms/step - loss: 0.0927 - accuracy: 0.9695 - val_loss: 0.0772 - val_accuracy:
Out[35]: <keras.callbacks.History at 0x7f71e8184070>
```

Presentat

Saving Model

```
In [36]: ls
fruit-dataset/

In [37]: model.save('fruit.h5')

In [40]: !tar -zcvf Train-model_new.tgz fruit.h5
fruit.h5

In [39]: ls -1
fruit-dataset/
fruit.h5
Train-model_new.tgz
```

IBM Cloud Deployment Model

In [41]:
```
!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-cli
ent) (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-learn
ing-client) (2.11.0)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-c
lient) (1.3.4)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-c
lient) (0.3.3)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-cl
ient) (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->wa
tson-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 boto
core<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-mach
ine-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-mac
hine-learning-client) (1.19.5)
Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391
```

In [43]:
```
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)
```

In [44]:
```
client = APIClient(wml_credentials)
```

In [45]:
```
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'])
```

In [46]:
```
space_uid = guid_from_space_name(client, 'Trainmodel')
print("Space UID = " + space_uid)
```

```
Space UID = 616c7d74-e99b-4c09-9922-27394a62c2d0
```

In [47]:
```
client.set.default_space(space_uid)
```

Out[47]:
```
'SUCCESS'
```

In [48]:
```
client.software_specifications.list()
```

Presenta

```
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

------------------------------  ----------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

In [51]:
```
software_space_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid
```

Out[51]: '1eb25b84-d6ed-5dde-b6a5-3fbdf1665666'

In [54]:
```
ls
```

```
fruit-dataset/  fruit.h5  Train-model_new.tgz
```

Presentatio

In [56]:
```
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}
    )
```

In [57]:
```
model_id = client.repository.get_model_id(model_details)
```

In [58]:
```
model_id
```

Out[58]: 'd0aeb6a2-e89c-4f8d-bf2f-a28ca4ea3cca'

In [60]:
```
ls
```

```
fruit-dataset/  fruit.h5  Train-model_new.tgz
```

Test The Model

In [54]:
```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

In [55]:
```
model=load_model('fruit.h5')
```

In [68]:
```
img=image.load_img(r"C:\Users\Sree Ram\Desktop\ibm\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Apple___healthy\0adc1c5
```

In [69]:
```
img
```

Out[69]:

In [70]:
```
img=image.load_img(r"C:\Users\Sree Ram\Desktop\ibm\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Apple___healthy\0adc1c5
img
```

Out[70]:

Presenta In [71]:
```
x=image.img_to_array(img)
```

In [72]:
```
x
```

Out[72]:
```
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.]],
```

```
        ...,
        [ 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)
```

In [73]:
```
x=np.expand_dims(x,axis=0)
```

In [74]:
```
x
```

Out[74]:
```
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.]],

        ...,
```

Presenta

```
                ...,
                [187., 188., 206.],
                [185., 186., 204.],
                [186., 187., 205.]]], dtype=float32)
```

In [75]:
```
y=np.argmax(model.predict(x),axis=1)
```

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

In [76]:
```
x_train.class_indices
```

Out[76]:
```
{'Apple___Black_rot': 0,
 'Apple___healthy': 1,
 'Corn_(maize)___Northern_Leaf_Blight': 2,
 'Corn_(maize)___healthy': 3,
 'Peach___Bacterial_spot': 4,
 'Peach___healthy': 5}
```

In [77]:
```
index=['Apple___Black_rot','Apple___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_sp
```

In [78]:
```
index[y[0]]
```

Out[78]:
```
'Apple___healthy'
```

In [82]:
```
img=image.load_img(r"C:\Users\Sree Ram\Desktop\ibm\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Peach___healthy\0a2ed4(
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','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_sp
index[y[0]]
```

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

Out[82]:
```
'Corn_(maize)___healthy'
```

In [83]:
```
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from flask import Flask,render_template,request
```

In [61]:
```
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)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bact
        text="The Classified Fruit disease is : " +str(index[pred[0]])
    return text
if __name__=='__main__':
    app.run(debug=False)
```

Presenta

REFERENCE:

⇨ Fertilizers Recommendation System For Disease Prediction In Tree Leave | SemanticScholar

⇨ Soil Based Fertilizer Recommendation System for Crop Disease Prediction System(ijetajournal.org)

⇨ Leaf Disease Detection and Fertilizer Suggestion | IEEE Conference Publication | IEEEXplore

⇨ IRJET-V7I1004.pdf

⇨ A nutrient recommendation system for soil fertilization based on evolutionarycomputation - ScienceDirect

⇨ Fertilizers-Recommendation-System-For-Disease-Prediction-In-Tree-Leave.pdf (ijstr.org)

⇨ 2204.11340.pdf (arxiv.org)

⇨ 371-376,Tesma405,IJEAST.pdf

⇨ CROFED - Crop and Fertilizer Recommendation and Disease diagnosis system usingMachine Learning and Internet of Things. (ijirt.org)

⇨ Prediction of Crop, Fertilizer and Disease Detection for Precision Agriculture by IJRASET -Issuu

- Demo link: https://www.youtube.com/watch?v=mWRefknub_g

Presentation title

Demo link: https://www.youtube.com/watch?v=mWRefknub g

Demo link: https://www.youtube.com/watch?v=mWRefknub g

Presentation title