FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

Submitted by

Team ID: PNT2022TMID44619

Yogapriyan.M	732319104026
Kubenthiran.M	732319104011
Mesavaraj.D	732319104019
Mayilsamy.M	732319104017

Project Report Format

1.INTRODUCTION

Fertilizer Recommentation system for disease Prediction is a simple ML and DL based website which recommends the best crop to grow, fertilizers to use and the diseases caught by your crops.

1.1. Project Overview

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 codes are first written in Jupyter notebook supplied along with Anaconda Python and then the codes are tested in IBM cloud. Finally, a web-based framework is designed with help 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.

1.2. Purpose

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

Artificial intelligence (AI) is a rapidly evolving area that offers unparalleled opportunities of progress and applications in many Agriculture fields. This proposal is regarding automatic detection of diseases and pathological part present within the leaf pictures of plants and even within the agriculture Crop production it is through with advancement of technology that helps in farming to extend the production. Primarily there is downside of detection accuracy and in neural network approach support vector machine (SVM) is exist already.

Indian economy greatly depends on agriculture. Nowadays detection of crop disease is very important topic for analysis. It is one of the issues that cause reduction in quality and quantity of crop. So detection and classification of crop disease is necessary task to increase crop productivity and economic process.

The proposed research work is to analyze various machine learning and image processing techniques applied to detect crop disease. In this paper we will review, different machine learning techniques, such as Random Forest, Decision tree, Multilayer Regressor, Regression algorithms, image processing techniques, Extreme Machine Learning to get better accuracy for system. Here, crop leaf images are taken as input and after processing that image, it will detect whether there is any disease or not. If disease is detected, then it will tell what type of disease it is and will provide solutions such as pesticides or chemicals to cure that disease. It will increase the productivity and economic process.

With increase in population the need for food is on rise, in such circumstances, plant diseases prove to be a major threat to agricultural produce and result in disastrous consequences for farmers. Early detection of plant disease can help in ensuring food security and controlling financial losses. The images of diseased plants can be used to identify the diseases. Classification abilities of Convolutional Neural Networks are used to obtain reliable output.

inappropriate crop for their land based on a conventional or non-scientific approach. This is a challenging task for a country like India, where agriculture feeds approximately 42% of the population. And the outcomes for the farmer of choosing the wrong crop for land is moving towards metro city for livelihoods, suicide, quitting the *agriculture and give land on lease to industrialist or use for the non-agriculture purpose. The outcome of wrong crop selection is less yield and less profit.*

In the paper **—Deep learning for Image-Based Plant detection" [1]** the authors Prasanna Mohanty et al**.,** has proposed an approach to detect disease in plants by training a convolutional neural network. The CNN model is trained to identify healthy and diseased plants of 14 species. The model achieved an accuracy of 99.35% on test set data.
Malvika Ranjan et al. in the paper **—Detection and Classification of leaf disease using Artificial Neural Network"** proposed an approach to detect diseases in plant utilizing the captured image of the diseased leaf. Artificial Neural Network (ANN) is trained by properly choosing feature values to distinguish diseased plants and healthy samples. The ANN model achieves an accuracy of 80%.
According to paper **—Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features" [3]** by S. Arivazhagan, disease identification process includes four main steps as follows: first, a color transformation structure is taken for the input RGB image, and then by means of a specific threshold value, the green pixels are detected and uninvolved, which is followed by segmentation process, and for obtaining beneficial segments the texture statistics are computed. At last, classifier is used for the features that are extracted to classify the disease.
Kulkarni et al. in the paper **—Applying image processing technique to detect plant diseases" [4],** a methodology for early and accurately plant diseases detection, using artificial neural network (ANN) and diverse image processing techniques. As the proposed approach is based on ANN classifier for classification and Gabor filter for feature extraction, it gives better results with a recognition rate of up to 91%.
Azadbakht et al. [2019] analyzed the performance for identifying leaf rust of wheat at canopy scale and at a dierent level of leaf area index (LAI) - high, medium and low. Four methods had been used for analyzing the performance in identifying the rust detection namely Gaussian process regression, random forest regression, v-support vector regression and boosted regression tree. The severity level of disease is also predicted in this. For analyzing the performance the experiment was conducted at North West of Iran in 7000 hectares of wheat cultivation. Hyperspectral reectance data recorded at dierent environmental condition is used for it. Based on the results from the experiment hyperspectral images can be used for identifying the wheat rust. Comparison of spectral vegetation indices (SVIs) and ML shows that ML performs better than SVIs.

Ip et al. [2018] presented a review about crop protection using big data and how it helps in controlling the weed. Discussed mainly about invasive species detection, predicting and modeling of resistance in herbicide, support system used for the purpose of crop protection and robotic weed control system. The machine learning approaches used for the above-mentioned problem are discussed clearly.
Markov random _eld uses spatial component for analyzing is explored in it. Probabilistic graphical model, traditional neural networks and support vector machine are the generative approach used in it. For the purpose of modeling and prediction of resistance in herbicide rule-based approach and CART is used. For detecting species and weeds random forest classi_er is used. RIM is used to control the ryegrass in the _eld. AgBot is robotics used as autonomous weed control.
Markov Random Field helps
in content-based fresh fruit grading and it has the ability to handle spatial data with irregular spacing. For modeling, the crop disease in corn Bayesian network is used and SVM helps in the semi-autonomous estimation of vegetables.
Chlingaryan et al. [2018] nitrogen estimation is an important factor in precision agriculture. Remote sensing system there is a problem in processing huge amount of data from a di_erent platform. The machine learning techniques help in handling the non-linear task. The mainly hybrid system of integrating machine learning techniques and remote sensing technique is used in nitrogen estimation. For estimating the chemical concentration in dry leaf Continual Removal (CR) method is used. For identifying more informative feature (BPNNs) is used. For calculating the vegetation indices back propagation neural networks are used. Feature extraction can be done by combining a convolution neural network and Gaussian process. The Gaussian process also helps in identifying di_erent plant leaves characteristic. For the purpose multi-class crop prediction M5 prime regression tree is used. And for nitrogen estimation, least square support vector machine is used. Fuzzy Cognitive Map is used in the _eld of crop management.
Lu et al. [2017] presented a system based on the deep learning framework named in _eld automatic wheat diagnosis system (DNIL- WDDS). In wild condition, the image-level annotation is done for the wheat disease. VGG-FCN-S and VGG-FCN-VD16 are the two architecture used in it. The proposed system achieves the accuracy of 97.95% and 95.12% when compared to CNN architecture achieving the accuracy of 93.27% and 73.00 %. It has an added advantage of integrating the system with the mobile application. Nearly four types of deep learning method are used for analyzing the disease prediction accuracy in wheat. The DMIL-WDDS model shows improvement in accuracy when compared to other models.

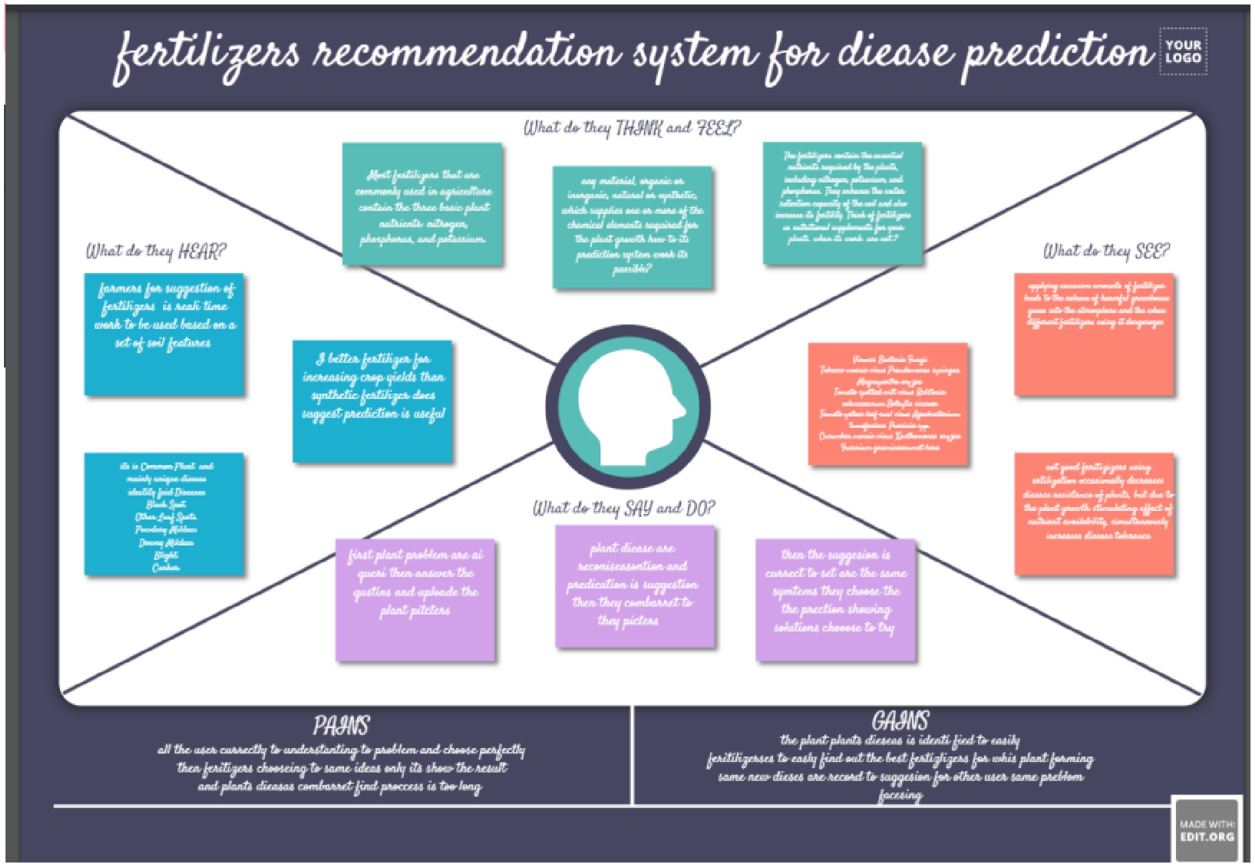
Tavakoli and Gebbers [2019] presented an analysis of winter wheat nitrogen and assessment of water in the _eld by using a camera. This experiment was conducted during a period of three years (2012, 2013, and 2014). Nitrogen fertilization and dierent level of water are applied in the _eld for the purpose of the experiment. Two machine learning algorithm was developed for the purpose of analysis namely Random Forest (RF) and Partial Least Square Regression (PLSR). Specter radiometer was used for radial measurement. Separately Vegetation Index (VI) is also calculated. For analyzing the nitrogen content R2 (RMSE) model is used and it is calculated separately for both data type. Random forest algorithm performs better in combined-date data. Nitrogen estimation calculation performs better while using the digital camera. It can also be integrated with the smartphone. It has a limitation of accessing only three spectral bands so that the analysis of plant status is also limited.
dos Santos Ferreira et al. [2017] proposed a method to identify unwanted weeds in the soybean _eld. Unwanted weed includes unwanted grasses and broadleaf. Convolution neural network technique is applied in the process of identifying the weeds in the soybean _eld. For the purpose of capturing the image, drones were used in it. The database used for analyzing purpose includes _fteen thousand pictures weeds, soil,
soybean, grass weed, and broadleaf. SafeNet architecture is used for training the neural network. The cafe software includes Alex Net in it. Pynovisao algorithm is used to build a robust image database. The results are compared with Support Vector Machine, Ada Boost, and Random Forest. The accuracy of 99% is achieved using the convolution neural network.
Barbedo [2018] the problems faced in the machine learning technique has been overcoming by the deep learning concepts such as Convolution Neural Networks (CNN). Large data sets are needed for processing this technique. This paper mainly focuses on how the size of data and its variety a_ects the performance of the deep learning concepts. 12 plant species with di_erent samples, di_erent disease, and di_erent character are taken into consideration. This analysis describes the di_erent CNN network used for disease classication along with where this large amount of data are collected for classi_cation. Accuracy is also calculated for each deep learning concepts. The number of correctly classi_ed sample divided by the total number of samples provides the accuracy value. List of di_erent plant species and its disease are listed in it. Removing background from image improves the prediction accuracy. This analysis was performed mainly using dataset obtained from di_erent sources.

Problem Statement Definition

In India, The Agriculture industry is extremely vital and crucial for economic and social development and jobs. In India, the agricultural sector provides a living for almost 48% of the population. As per the 2019-2020 economic survey, an Indian farmer's median wage in 16 states is Rupees 2500. Most of the Indian population depends on agriculture for their livelihood. Agriculture gives an opportunity of employment to the village people to develop a country like India on large scale and give a push in the economic sector. The majority of farmers face the problem of planting an

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas



fertilizers recommendation system for diease prediction

What do they THINK and FEEL?

Most fertilizers that are commonly used in agriculture contain the three basic plant nutrients: nitrogen, phosphorus, and potassium.

any material, organic or inorganic, natural or synthetic, which supplies one or more of the chemical elements required for the plant growth how to its prediction system work its possible?

The fertilizers contain the essential nutrients required by the plants, including nitrogen, potassium, and phosphorus. They enhance the water retention capacity of the soil and also increase its fertility. Think of fertilizers as nutritional supplements for your plants. when its work are not?

What do they HEAR?

farmers for suggestion of fertilizers is real time work to be used based on a set of soil features

I better fertilizer for increasing crop yields than synthetic fertilizer does suggest prediction is useful

its is Common Plant and mainly unique disease identify find Diseases
Black Spot
Other Leaf Spots
Powdery Mildew
Downy Mildew
Blight
Canker

What do they SEE?

applying excessive amounts of fertilizer leads to the release of harmful greenhouse gases into the atmosphere and the when different fertilizers using it dangerous

Viruses Bacteria Fungi
Tobacco mosaic virus Pseudomonas syringae Magnaporthe oryzae
Tomato spotted wilt virus Ralstonia solanacearum Botrytis cinerea
Tomato yellow leaf curl virus Agrobacterium tumefaciens Puccinia spp.
Cucumber mosaic virus Xanthomonas oryzae Fusarium graminearum it here

not good fertilizers using fertilization occasionally decreases disease resistance of plants, but due to the plant growth stimulating effect of nutrient availability, simultaneously increases disease tolerance

What do they SAY and DO?

first plant problem are ai queri then answer the qustins and uploade the plant pitcters

plant diease are reconiseasontion and predication is suggestion then they combarret to they picters

then the suggesion is currect to set are the same symtems they choose the the prection showing solutions choose to try

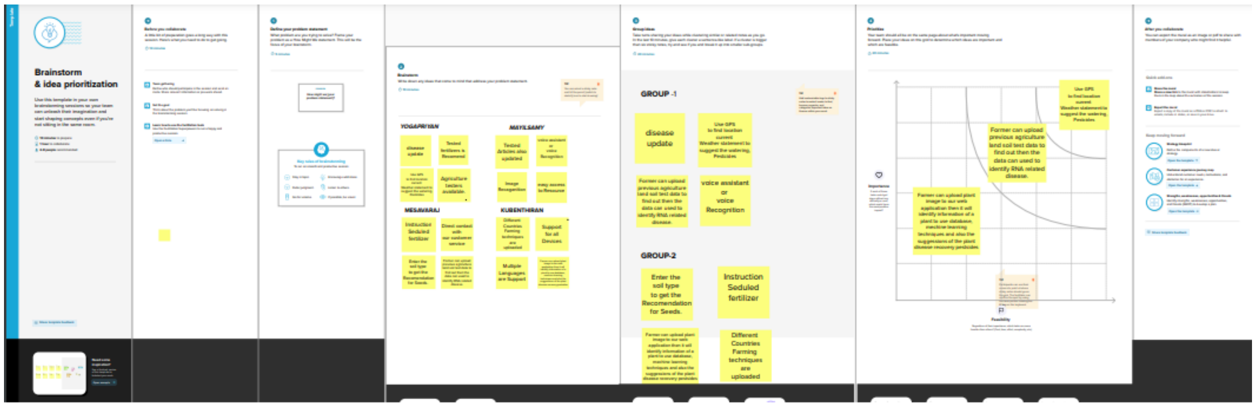
PAINS

all the user currectly to understanting to problem and choose perfectly then feritigers chooseing to same ideas only its show the result and plants dieasas combarret find proccess is too long

GAINS

the plant plants dieseas is identi fied to easily feritilizerses to easly find out the best fertigligers for whis plant forming same new dieses are record to suggesion for other user same prebllom facesing

3.2. Ideation & Brainstorming


Brainstorm
& idea prioritization
YOGAPRIYAN
MAYILSAMY
MESAVARAJ
KUBENTHIRAN
GROUP -1
disease update
voice assistant or voice Recognition
GROUP-2
Instruction Seduled fertilizer
Enter the soil type to get the Recomendation for Seeds.
Different Countries Farming techniques are uploaded
Former can upload previous agriculture land soil test data to find out then the data can used to identify RNA related disease.

3.3.Proposed Solution

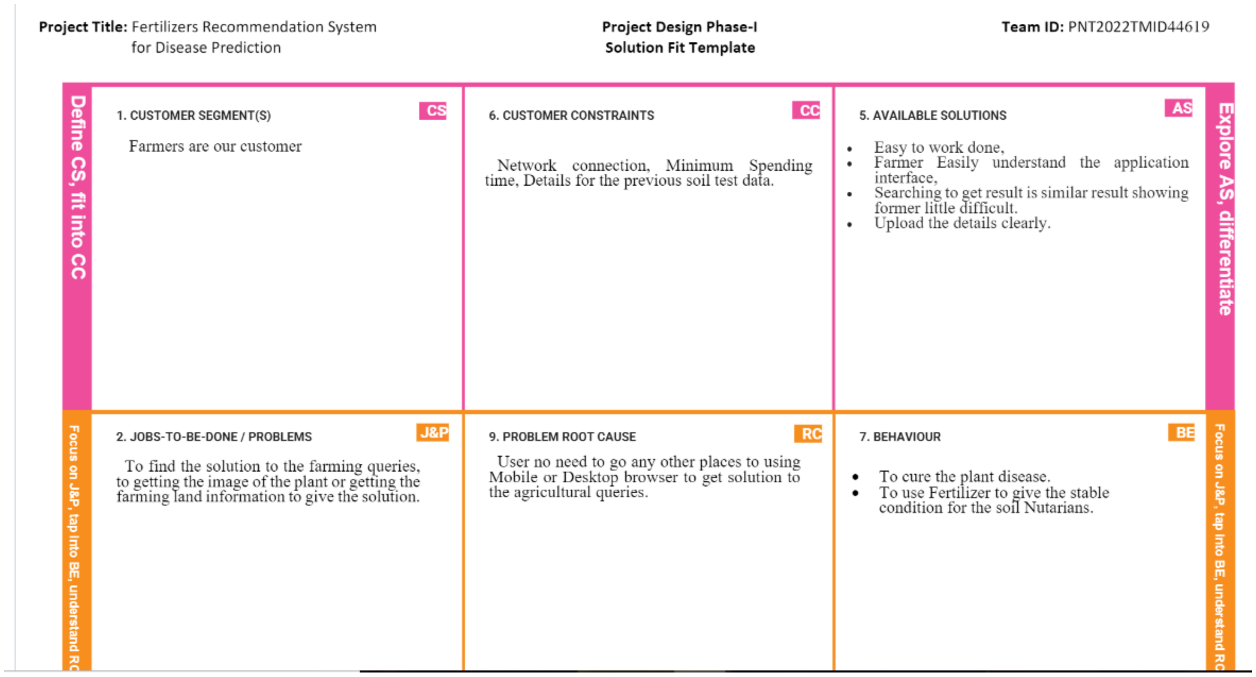
Project team shall fill the following information in proposed solution template.

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	Farmers' conventional methods of agricultural cultivation are ineffective. It does not make proper use of all available resources. Farmers are unable to detect crop diseases due to a lack of knowledge and old practices, which often result in soil nutrient deterioration and exhaustion. As a result, crop failure occurs. Growing only certain crops depletes the soil, and if the crops are harmed by illnesses, farmers are uninformed of how to recover such crops. Food needs cannot be met until and unless efficient resource management and use is implemented.
2.	Idea / Solution description	Efficient approach for controlling the overuse of insecticides and fertilizers in farming. Implementation of artificial intelligence for identification of pests and recommendation of insecticides using TPF-CNN.
3.	Novelty / Uniqueness	The proposed method uses SVM to classify tree leaves, identify the disease and suggest the fertilizer. The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result when compared to existing CNN.
4.	Social Impact / Customer Satisfaction	It also helps farmer to perform the activities like crop management including applications on yield prediction, disease detection, weed detection, crop quality, and growth prediction etc. This chapter describes the case study on "Crop Disease Detection and Yield prediction". The study includes identification of crop Agriculture is the mainstay of a rising economy in India. condition, disease detection, prediction about specific crop and recommendation using machine learning algorithms. It gives an idea about how recommender system is used in agriculture for disease detection and prediction.

5.	Business Model (Revenue Model)	Being an extremely vital industry as it manufactures some of the most important raw materials required for crop production, it is not wrong to say that the success of the agricultural sector in India is largely dependent on the fertilizer industry. Fertilizers are extensively being used to improve per hectare production of crops that can be used for food and industrial applications. If you like the idea of making a profit by helping people work with the soil, you might enjoy being a part of the fertilizer industry.
6.	Scalability of the Solution	Fertilizers replace the nutrients that crops remove from the soil. Without the addition of fertilizers, crop yields and agricultural productivity would be significantly reduced. That's why mineral fertilizers are used to supplement the soil's nutrient stocks with minerals that can be quickly absorbed and used by crops.

1.1

3.4Problem Solution fit



Project Title: Fertilizers Recommendation System for Disease Prediction

Project Design Phase-I
Solution Fit Template

Team ID: PNT2022TMID44619

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Farmers are our customer

6. CUSTOMER CONSTRAINTS CC

Network connection, Minimum Spending time, Details for the previous soil test data.

5. AVAILABLE SOLUTIONS AS

- Easy to work done,
- Farmer Easily understand the application interface,
- Searching to get result is similar result showing former little difficult.
- Upload the details clearly.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P

To find the solution to the farming queries, to getting the image of the plant or getting the farming land information to give the solution.

9. PROBLEM ROOT CAUSE RC

User no need to go any other places to using Mobile or Desktop browser to get solution to the agricultural queries.

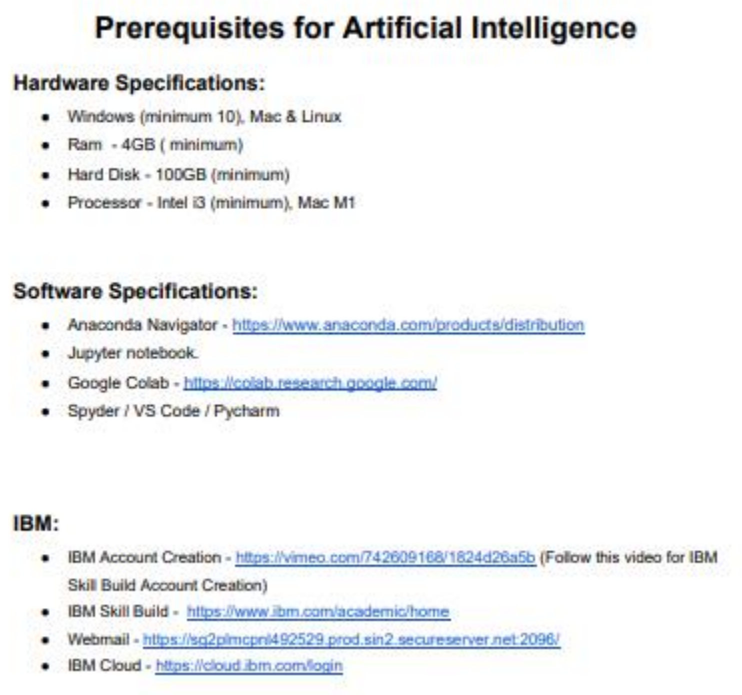
7. BEHAVIOUR BE

- To cure the plant disease.
- To use Fertilizer to give the stable condition for the soil Nutarians.

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

Identify strong TR & EM	3. TRIGGERS TR Anytime access, free of cost, work efficiency, find the disease. 4. EMOTIONS: BEFORE / AFTER EM Farmers need to deal with many problems cope with climate change, soil Erosion and biodiversity loss. Farmers didn't know all the information about farming techniques and disease. Get disease for plant to find the problem with solution to the disease Case.	10. YOUR SOLUTION SL To get the image and also agricultural land information to recommend the Fertilizers . Also give the disease prediction ideas . It fit the agricultural quires to improve the growth of the crops.	8. CHANNELS of BEHAVIOUR CH Online	Identify strong TR & EM

4.REQUIREMENT ANALYSIS



Prerequisites for Artificial Intelligence

Hardware Specifications:

- Windows (minimum 10), Mac & Linux
- Ram - 4GB (minimum)
- Hard Disk - 100GB (minimum)
- Processor - Intel i3 (minimum), Mac M1

Software Specifications:

- Anaconda Navigator - https://www.anaconda.com/products/distribution
- Jupyter notebook.
- Google Colab - https://colab.research.google.com/
- Spyder / VS Code / Pycharm

IBM:

- IBM Account Creation - https://vimeo.com/742609168/1824d26a5b (Follow this video for IBM Skill Build Account Creation)
- IBM Skill Build - https://www.ibm.com/academic/home
- Webmail - https://sg2plmcpnl492529.prod.sin2.secureserver.net:2096/
- IBM Cloud - https://cloud.ibm.com/login

4.1. Functional requirement & Non-Functional requirements

Ideation Phase

Title	Description	Status
Literature survey	Literature Survey on the selected projects & gathering information by referring the technical papers etc.	COMPLETED
Brainstorm and Idea prioritization	List the Idea's by organizing the brainstorming session & prioritize the top 4 Ideas based on the Feasibility & Importance	COMPLETED
Problem statement	List of problems in the project	COMPLETED
Prepare empathy map	Prepare Empathy Map Canvas to capture the User Pains & Gains, prepare list of problem statements	COMPLETED

Project Design Phase – 1

Title	Description	Status
Proposed Solution	Prepare the Proposed Solution Document, which includes the Novelty, Feasibility of Idea,Social impact, Scalability of solution, etc.	COMPLETED
Problem Solution Fit	Prepare Problem – Solution Fit Document	COMPLETED
Solution Architecture	Prepare the Technology (Solution) Architecture diagram	COMPLETED
Data Flow Diagram	Draw the data flow diagrams & submit for review	COMPLETED
Technology Architecture	Prepare the Technology Architecture Diagram	COMPLETED

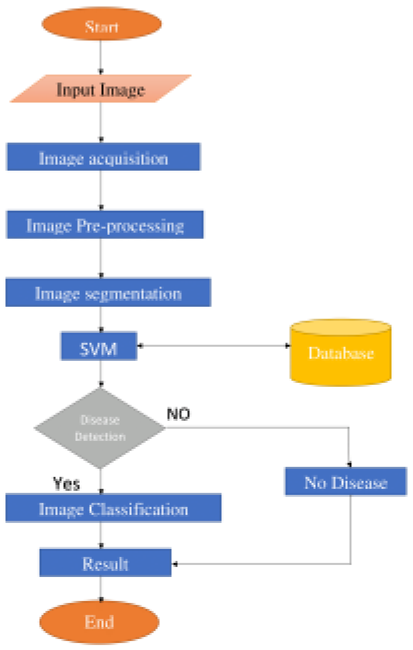
Project Planning Phase

Title	Description	Status
Prepare Project Planning & Sprint Delivery Plan	Prepare the Product Backlog, Sprint planning, stories & Story points	COMPLETED
Prepare Milestone & Activity List	Prepare the Milestones & activity list of the project	COMPLETED

5. PROJECT DESIGN

5.1. Data Flow Diagrams

Data Flow Diagrams:



5.2. Solution & Technical Architecture

Solution Architecture:

• Bridge between Farmer and Developer

. • Technical solution for Fertilizer recommendation system using medium for online.

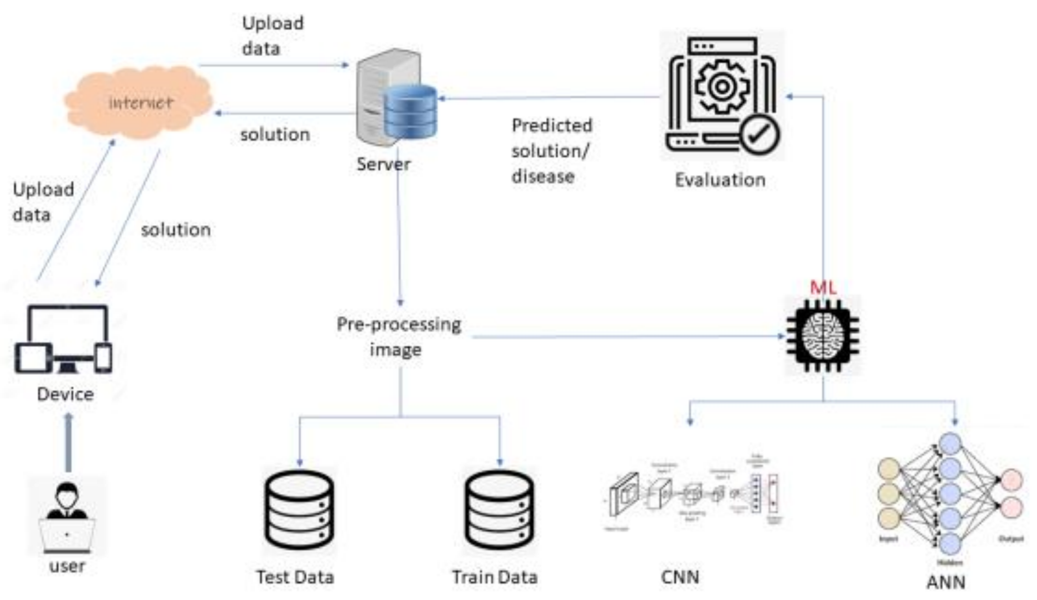
• Feature is to use of Machine Learning Algorithm, Deep learning algorithm (ANN, CNN) to image reorganization to give the solution.

• Solution Requirement want to get the image of the plant, also get the land or soil information of Agricultural land.

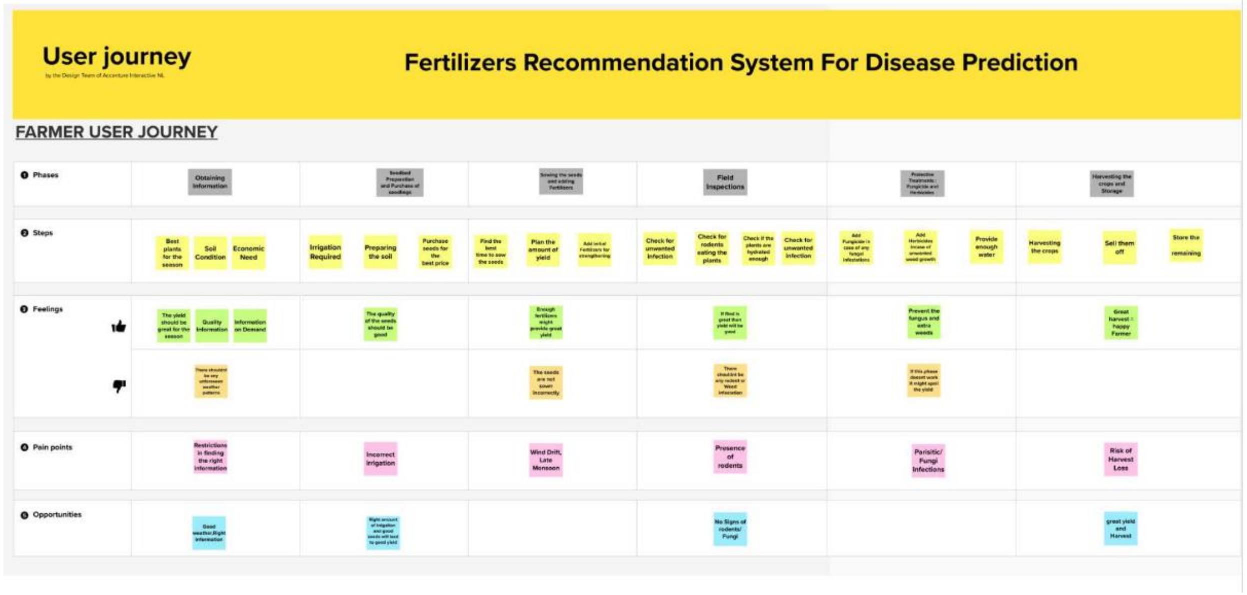
• Characteristics topic diversity, novelty, temporal awareness, risk awareness.

• Behaviour recommendation system based on user behaviour.

• Farmer aspects of the software user is a stack holder. Solution Architecture Diagram:



5.3.userstory



User journey

by the Design Team of Accenture Interactive NL

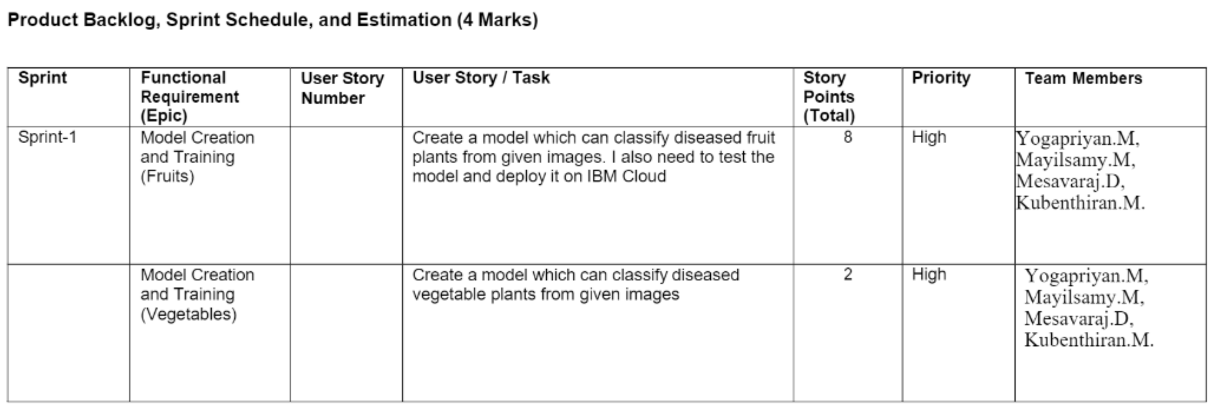
Fertilizers Recommendation System For Disease Prediction

FARMER USER JOURNEY

❶ Phases		Obtaining Information	Seedbed Preparation and Purchase of seedlings	Sowing the seeds and adding Fertilizers	Field Inspections	Protective Treatments : Fungicide and Herbicides	Harvesting the crops and Storage
❷ Steps		Best plants for the season; Soil Condition; Economic Need	Irrigation Required; Preparing the soil; Purchase seeds for the best price	Find the best time to sow the seeds; Plan the amount of yield; Add initial Fertilizers for strengthening	Check for unwanted infection; Check for rodents eating the plants; Check if the plants are hydrated enough; Check for unwanted infection	Add Fungicide in case of any fungal infestations; Add Herbicides incase of unwanted weed growth; Provide enough water	Harvesting the crops; Sell them off; Store the remaining
❸ Feelings	👍	The yield should be great for the season; Quality Information; Information on Demand	The quality of the seeds should be good	Enough fertilizers might provide great yield	If feed is great then yield will be good	Prevent the fungus and extra weeds	Great harvest = happy Farmer
	👎	There shouldn't be any unforeseen weather patterns		The seeds are not sown incorrectly	There shouldn't be any rodent or Weed infestation	If this phase doesn't work it might spoil the yield	
❹ Pain points		Restrictions in finding the right information	Incorrect irrigation	Wind Drift, Late Monsoon	Presence of rodents	Parisitic/ Fungi Infections	Risk of Harvest Loss
❺ Opportunities		Good weather,Right information	Right amount of irrigation and good seeds will lead to good yield		No Signs of rodents/ Fungi		great yield and Harvest

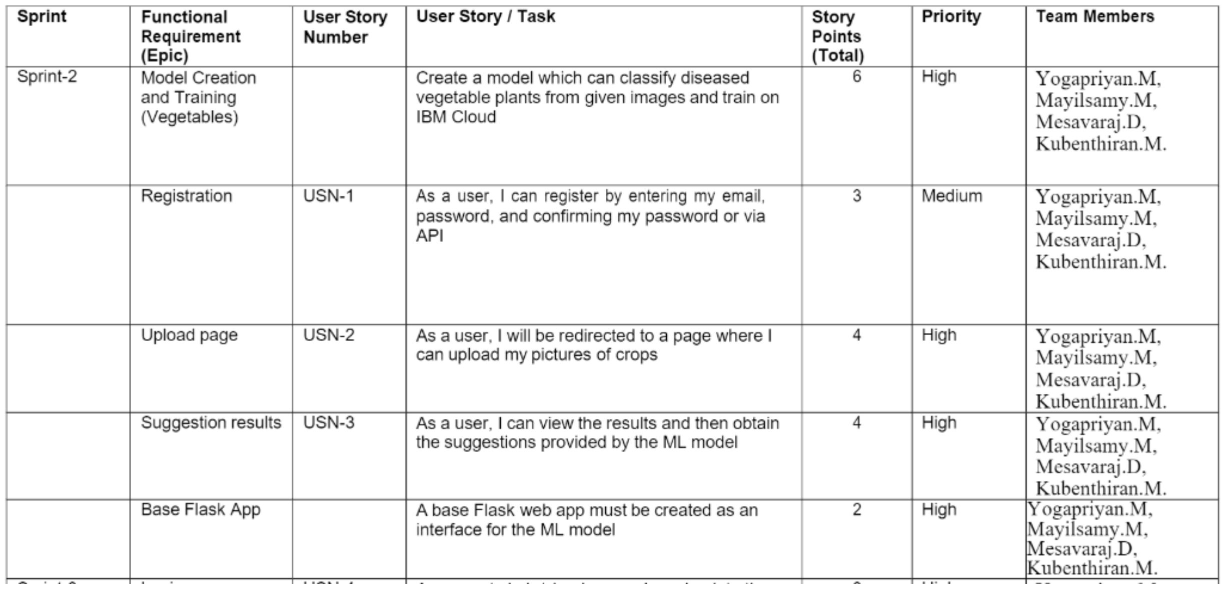
6. PROJECT PLANNING & SCHEDULING

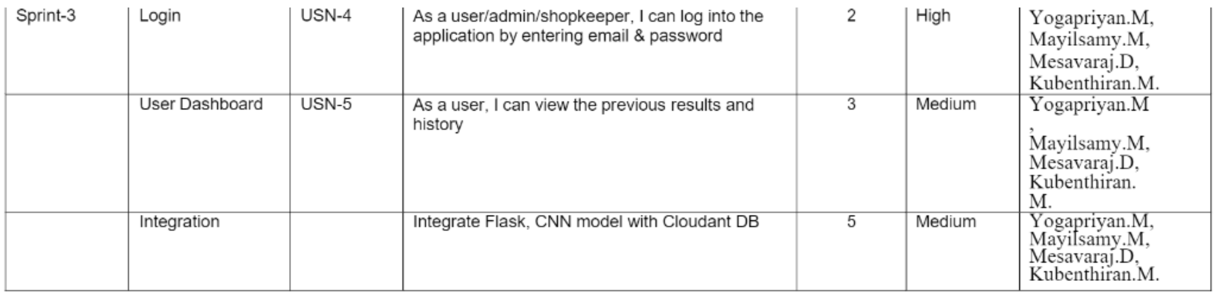
6.1Sprint Planning & Estimation



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

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





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

6.2. Sprint Delivery Schedule

import the libraries

```
In [1]: from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Dense
```

```
In [10]: #import the cnn layers
```

```
In [11]: from tensorflow.keras.layers import Convolution2D
         from tensorflow.keras.layers import MaxPooling2D
         from tensorflow.keras.layers import Flatten
```

```
In [12]: #image preprocessinh (or) image augmentation
         from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
In [13]: train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True,vertical_flip=True)
         #rescale => rescaling pixel value from 0 to 255 to 0 to 1
         #shear_range=> counter clock wise rotation(anti clock)
```

```
In [15]: test_datagen = ImageDataGenerator(rescale=1./255)
```

```
In [17]: x_train = train_datagen.flow_from_directory(r"C:\Users\ELCOT\fruit-dataset\fruit-dataset\train",target_size=(64,64),batch_size=3

         Found 5384 images belonging to 6 classes.
```

```
In [19]: x_test = test_datagen.flow_from_directory(r"C:\Users\ELCOT\fruit-dataset\fruit-dataset\test",target_size=(64,64),batch_size=32,c

         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}
```

```
In [21]: model = Sequential()
```

```
In [25]: # add convolution layer
```

```
In [22]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation="relu"))
         # 32 indicates => no of feature detectors
         #(3,3)=> kernel size (feature detector size)
```

```
In [15]: #add max pooling layer
```

```
In [23]: model.add(MaxPooling2D(pool_size=(2,2)))
```

```
In [17]: #add flatten layer => input to your ANN
```

```
In [24]: model.add(Flatten())
```

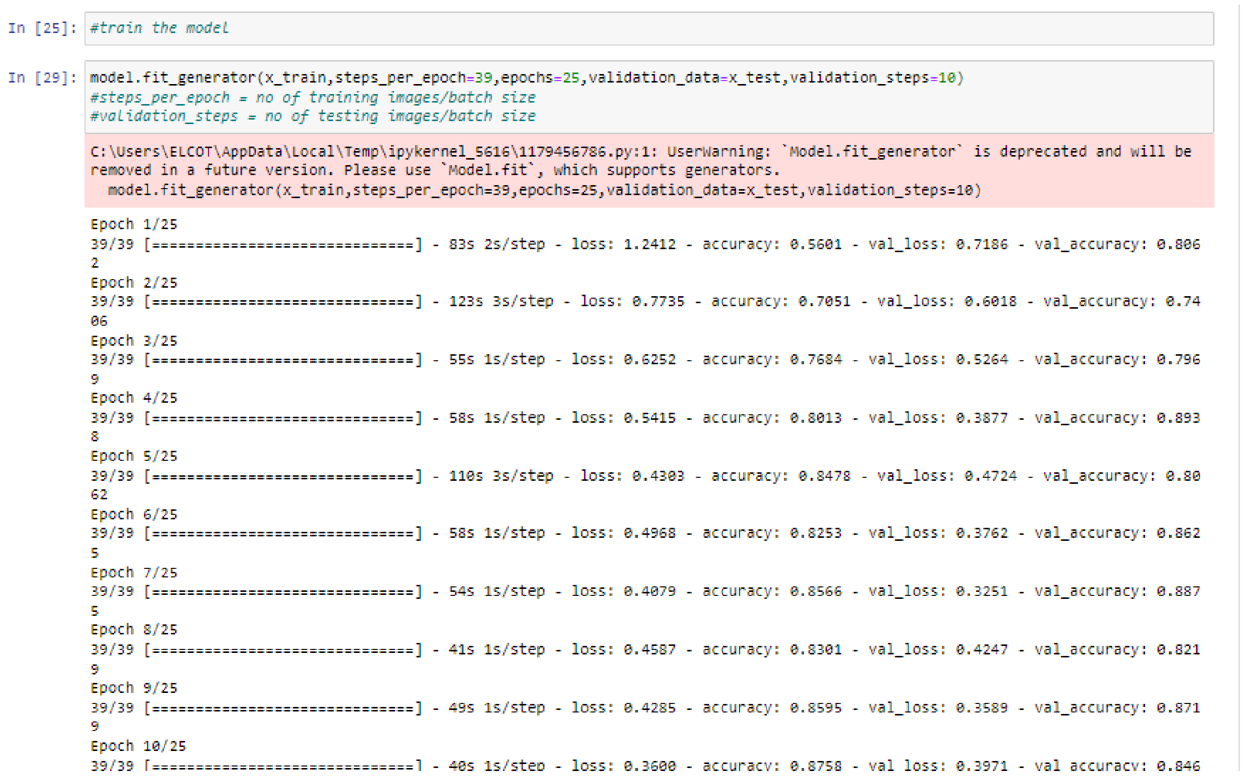
```
In [19]: #hidden layer
```

```
In [25]: model.add(Dense(units=300,kernel_initializer="random_uniform",activation="relu"))
```

```
In [26]: model.add(Dense(units=200,kernel_initializer="random_uniform",activation="relu"))
```

```
In [22]: #output layer
```

```
In [27]: model.add(Dense(units=6,kernel_initializer="random_uniform",activation="softmax"))
```



```
In [25]: #train the model

In [29]: model.fit_generator(x_train,steps_per_epoch=39,epochs=25,validation_data=x_test,validation_steps=10)
         #steps_per_epoch = no of training images/batch size
         #validation_steps = no of testing images/batch size

C:\Users\ELCOT\AppData\Local\Temp\ipykernel_5616\1179456786.py:1: UserWarning: `Model.fit_generator` is deprecated and will be
removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=39,epochs=25,validation_data=x_test,validation_steps=10)

Epoch 1/25
39/39 [==============================] - 83s 2s/step - loss: 1.2412 - accuracy: 0.5601 - val_loss: 0.7186 - val_accuracy: 0.806
2
Epoch 2/25
39/39 [==============================] - 123s 3s/step - loss: 0.7735 - accuracy: 0.7051 - val_loss: 0.6018 - val_accuracy: 0.74
06
Epoch 3/25
39/39 [==============================] - 55s 1s/step - loss: 0.6252 - accuracy: 0.7684 - val_loss: 0.5264 - val_accuracy: 0.796
9
Epoch 4/25
39/39 [==============================] - 58s 1s/step - loss: 0.5415 - accuracy: 0.8013 - val_loss: 0.3877 - val_accuracy: 0.893
8
Epoch 5/25
39/39 [==============================] - 110s 3s/step - loss: 0.4303 - accuracy: 0.8478 - val_loss: 0.4724 - val_accuracy: 0.80
62
Epoch 6/25
39/39 [==============================] - 58s 1s/step - loss: 0.4968 - accuracy: 0.8253 - val_loss: 0.3762 - val_accuracy: 0.862
5
Epoch 7/25
39/39 [==============================] - 54s 1s/step - loss: 0.4079 - accuracy: 0.8566 - val_loss: 0.3251 - val_accuracy: 0.887
5
Epoch 8/25
39/39 [==============================] - 41s 1s/step - loss: 0.4587 - accuracy: 0.8301 - val_loss: 0.4247 - val_accuracy: 0.821
9
Epoch 9/25
39/39 [==============================] - 49s 1s/step - loss: 0.4285 - accuracy: 0.8595 - val_loss: 0.3589 - val_accuracy: 0.871
9
Epoch 10/25
39/39 [==============================] - 40s 1s/step - loss: 0.3600 - accuracy: 0.8758 - val_loss: 0.3971 - val_accuracy: 0.846
```

Fertilizer Recommendation System For Disease Prediction

:- Prior Knowledge: Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at their earliest. Plant pathologists can analyze digital images using digital image processing to diagnose plant diseases. Application of computer vision and image processing strategies assists farmers in all agriculture regions. Generally, plant diseases are caused by the abnormal physiological functionalities of plants. Therefore, the characteristic symptoms are generated based on the differentiation between expected physiological functionalities and abnormal physiological functionalities of the plants.

Mainly, plant leaf diseases are caused by Pathogens positioned on the plants' stems. Different methods in image processing predict these different symptoms and diseases of leaves. These different methods include different fundamental processes like segmentation, feature extraction and classification and so on. Mainly, the prediction and diagnosis of leaf diseases depend on segmentation, such as segmenting the healthy tissues from diseased tissues of leaves.

1. Image Classification The proposed image classification technique is divided into the following steps. 2. Image acquisition The purpose of image preprocessing is improving image statistics so that undesired distortions are suppressed and image capabilities which are probably relevant for similar processing are emphasized. The preprocessing receives an image as input and generates an output image as a grayscale, an invert and a smoothed one. 3. Segmentation Implements Guided active contour method. Unconstrained active contours applied to the difficult natural images. Dealing with unsatisfying contours, which would try and make their way through every possible grab cut in the border of the leaf. The proposed solution is used the polygonal model obtained after the first step not only as an initial leaf contour but also as a shape prior that will guide its evolution towards the real leaf boundary 4. Disease Prediction Leaves are affected by bacteria, fungi, virus, and other insects. Support Vector Machine (SVM) algorithm classifies the leaf image as normal or affected. Vectors are constructed based on leaf features such as color, shape, textures. Then hyperplane constructed with conditions to categorize the preprocessed leaves and also implement multiclass classifier, to predict diseases in leaf image with improved accuracy 5. Fertilizer Recommendation Recommend the fertilizer for affected leaves based on severity level. Fertilizers may be organic or inorganic. Admin can store the fertilizers based on disease categorization with severity levels. The measurements of fertilizers suggested based on disease severity

7. CODING & SOLUTIONING

```
{% extends 'login new.html' %} {% block body %}


<div class="bg"></div>

<div class="bg bg2"></div>

<div class="bg bg3"></div>

<div class="content">

   <h1></h1>

</div>


<style>

  html body {

    background: rgb(2,0,36);

background: linear-gradient(11deg, rgba(2,0,36,1) 0%, rgba(15,121,9,1) 0%,

rgba(12,141,63,1) 42%, rgba(0,212,255,1) 100%);

  }

</style>

<style>

  label {


font-family: 'Raleway', sans-serif;

font-size: 1.1875rem;

font-weight: 400;

letter-spacing: 0.1rem;

font-style: normal;
text-transform: capitalize;

color: #FFFFFF;

background-color: #5ca8f0;

border-radius: 0.625rem;

-webkit-border-radius: 0.625rem;

-moz-border-radius: 0.625rem;

padding: 0.3rem 0.4rem;

border-style: solid;

border-width: 0rem;

border-color: #EF2D56;

-webkit-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);

-moz-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);

-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);


}

</style>

<br /><br />

 <h2 style="text-align: center; margin: 0px; color: black"><b>Wanna know which disease
Caught to your plant ?</b></h2>

 <br />

 <br>


<div style="

  width: 350px;

  height: 50rem;
```

```
    margin: 0px auto;

    color: black;

    border-radius: 25px;

    padding: 10px 10px;

    font-weight: bold;

  ">


<style>

  form {


font-family: 'Raleway', sans-serif;

font-size: 1.1875rem;

font-weight: 400;

letter-spacing: 0.1rem;

font-style: normal;

text-transform: capitalize;

color: #FFFFFF;

background-color: #5ca8f0;

border-radius: 0.625rem;

-webkit-border-radius: 0.625rem;

-moz-border-radius: 0.625rem;

padding: 0.3rem 0.4rem;

border-style: solid;

border-width: 0rem;

border-color: #EF2D56;
-webkit-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);
-moz-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);
-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);


}
</style>



  <form class="form-signin" method=post enctype=multipart/form-data>


    <h2 class="h4 mb-3 font-weight-normal">Upload Your Image</h2>
    <input type="file" name="file" class="form-control-file" id="inputfile"
onchange="preview_image(event)" style="font-weight: bold;">
    <br>
    <br>
    <img id="output-image" class="rounded mx-auto d-block" />
    <button class="btn btn-lg btn-primary btn-block" type="submit" style="font-weight:
bold;">Predict</button>



  </form>


</div>


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

```
 function preview_image(event) {
  var reader = new FileReader();
  reader.onload = function () {
   var output = document.getElementById('output-image')
   output.src = reader.result;
  }
  reader.readAsDataURL(event.target.files[0]);
 }
</script>


</div>
{% endblock %}
```

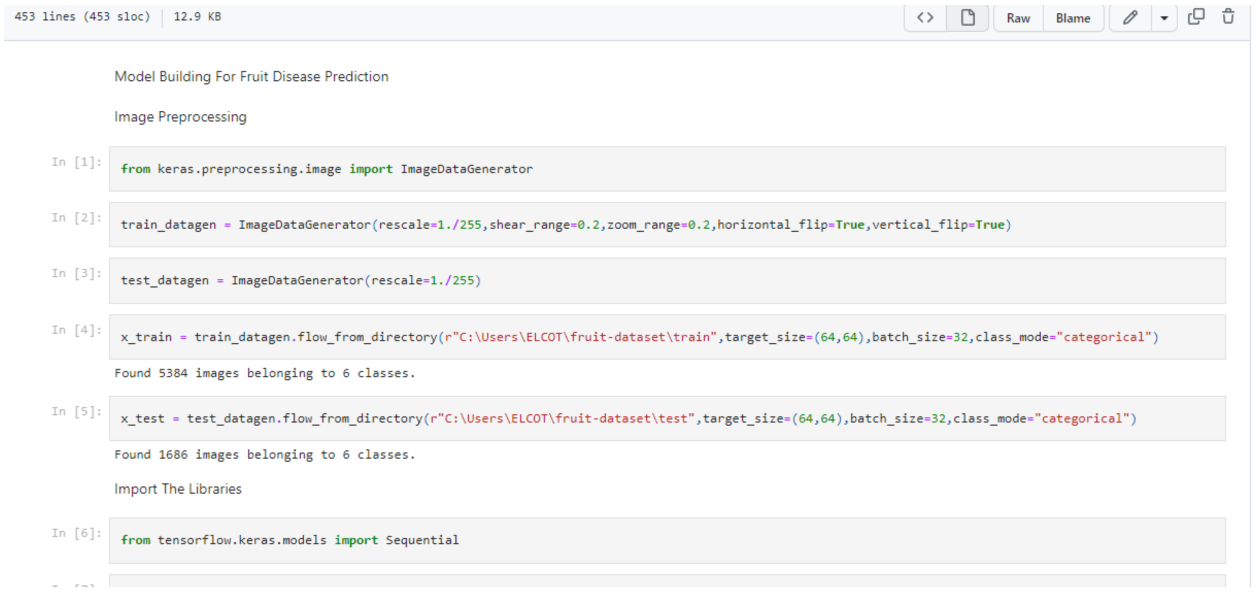
Feature 2

```
{% extends 'login new.html' %} {% block body %}


<div class="container py-2 mx-auto my-50 h-10 " style="margin: 9rem;">
 <div class="row">
   <div class="col-sm py-2 py-md-3">
     <div class="card card-body" style="justify-content: center;color: black; font-size: 15px;
font-family: Consolas, monaco, monospace; font-size: 24px; font-style: normal; font-variant:
normal; font-size: 1.1875rem;
     font-weight: 400;
     letter-spacing: 0.1rem;
     text-transform: capitalize;
```

```
          color: #FFFFFF;

          background-color: #5ca8f0;

          border-radius: 0.625rem;

          -webkit-border-radius: 0.625rem;

          -moz-border-radius: 0.625rem;

          padding: 0.3rem 0.4rem;

          border-style: solid;

          border-width: 0rem;

          border-color: #EF2D56;

          -webkit-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);

          -moz-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);

          -box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);">


            <p class="text-center" style="color:black">{{ prediction }}</p>

          </div>

        </div>

  </div>

</div>

<style>

  html body {

  background-color: #7cfd83;

}

</style>

{% endblock %}
```

7.3. Database Schema

Fruit data set



453 lines (453 sloc) | 12.9 KB

Model Building For Fruit Disease Prediction

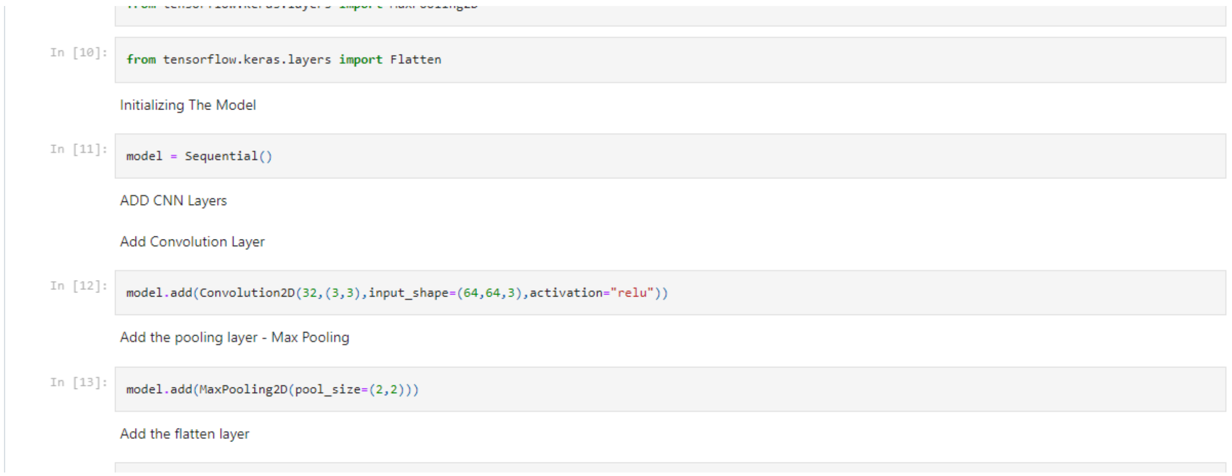
Image Preprocessing

```
In [1]: from keras.preprocessing.image import ImageDataGenerator

In [2]: train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True,vertical_flip=True)

In [3]: test_datagen = ImageDataGenerator(rescale=1./255)

In [4]: x_train = train_datagen.flow_from_directory(r"C:\Users\ELCOT\fruit-dataset\train",target_size=(64,64),batch_size=32,class_mode="categorical")

Found 5384 images belonging to 6 classes.

In [5]: x_test = test_datagen.flow_from_directory(r"C:\Users\ELCOT\fruit-dataset\test",target_size=(64,64),batch_size=32,class_mode="categorical")

Found 1686 images belonging to 6 classes.
```

Import The Libraries

```
In [6]: from tensorflow.keras.models import Sequential
```

Veg data set



```
In [10]: from tensorflow.keras.layers import Flatten
```

Initializing The Model

```
In [11]: model = Sequential()
```

ADD CNN Layers

Add Convolution Layer

```
In [12]: model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation="relu"))
```

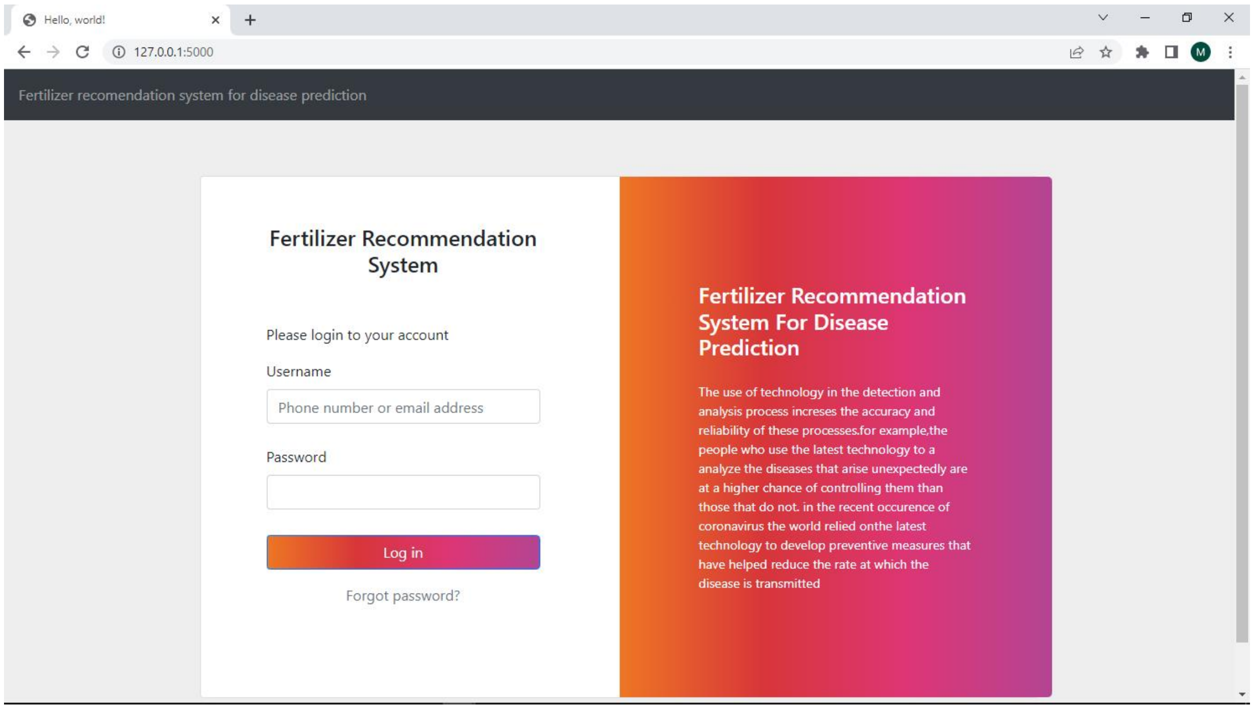
Add the pooling layer - Max Pooling

```
In [13]: model.add(MaxPooling2D(pool_size=(2,2)))
```

Add the flatten layer

8. TESTING

8.1. Test Cases


Hello, world!

127.0.0.1:5000

Fertilizer recomendation system for disease prediction

Fertilizer Recommendation System

Please login to your account

Username

Phone number or email address

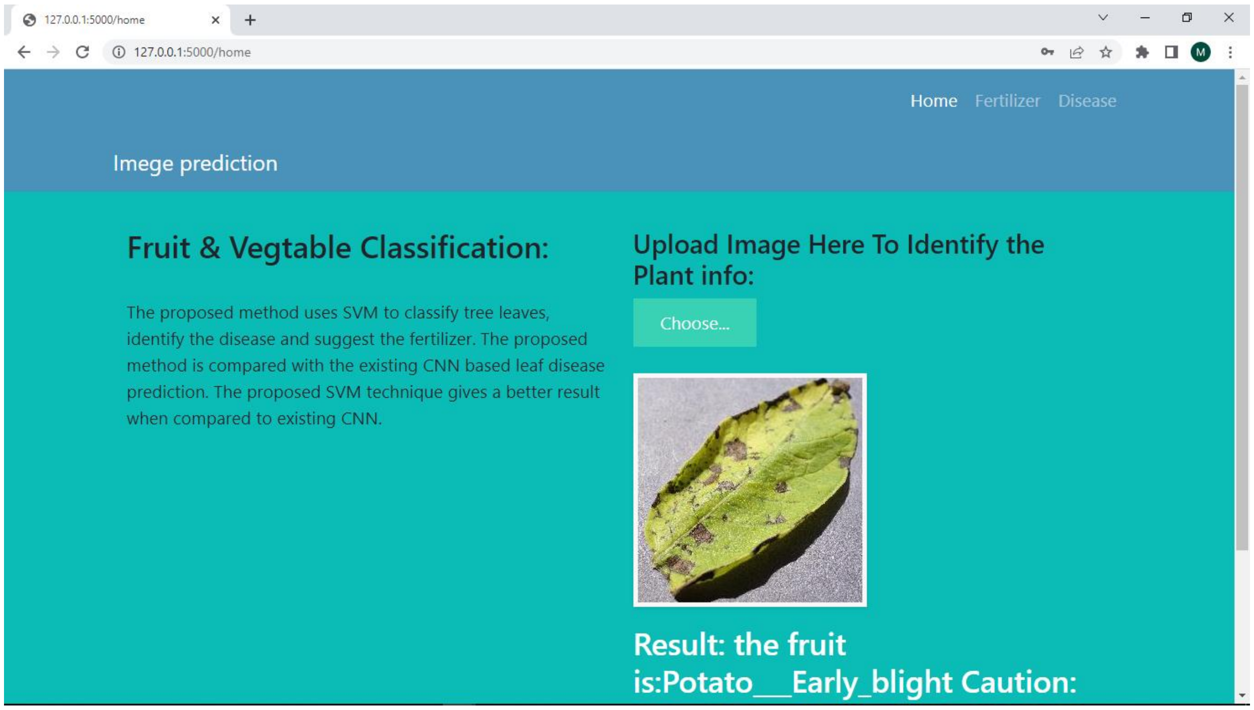
Password

Log in

Forgot password?

Fertilizer Recommendation System For Disease Prediction

The use of technology in the detection and analysis process increses the accuracy and reliability of these processes.for example,the people who use the latest technology to a analyze the diseases that arise unexpectedly are at a higher chance of controlling them than those that do not. in the recent occurence of coronavirus the world relied onthe latest technology to develop preventive measures that have helped reduce the rate at which the disease is transmitted


127.0.0.1:5000/home

127.0.0.1:5000/home

Home Fertilizer Disease

Imege prediction

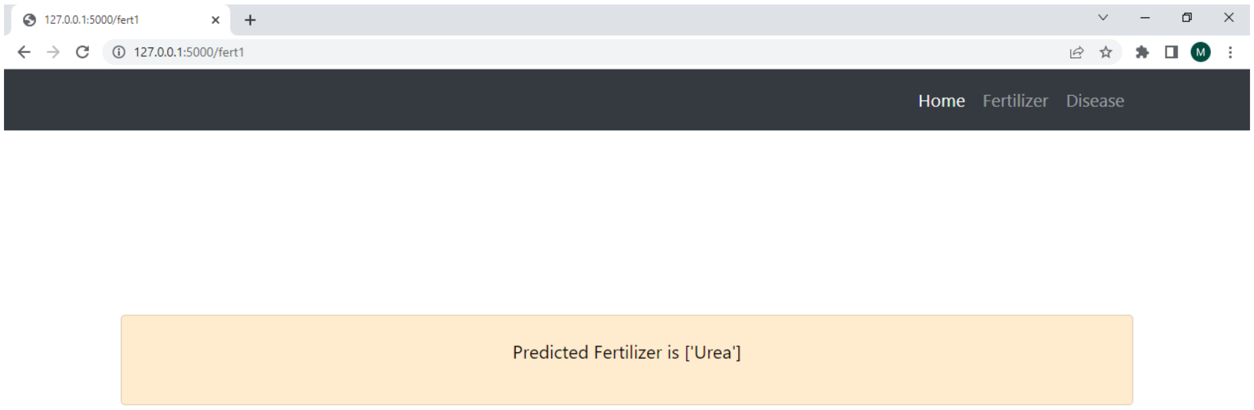
Fruit & Vegtable Classification:

The proposed method uses SVM to classify tree leaves, identify the disease and suggest the fertilizer. The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result when compared to existing CNN.

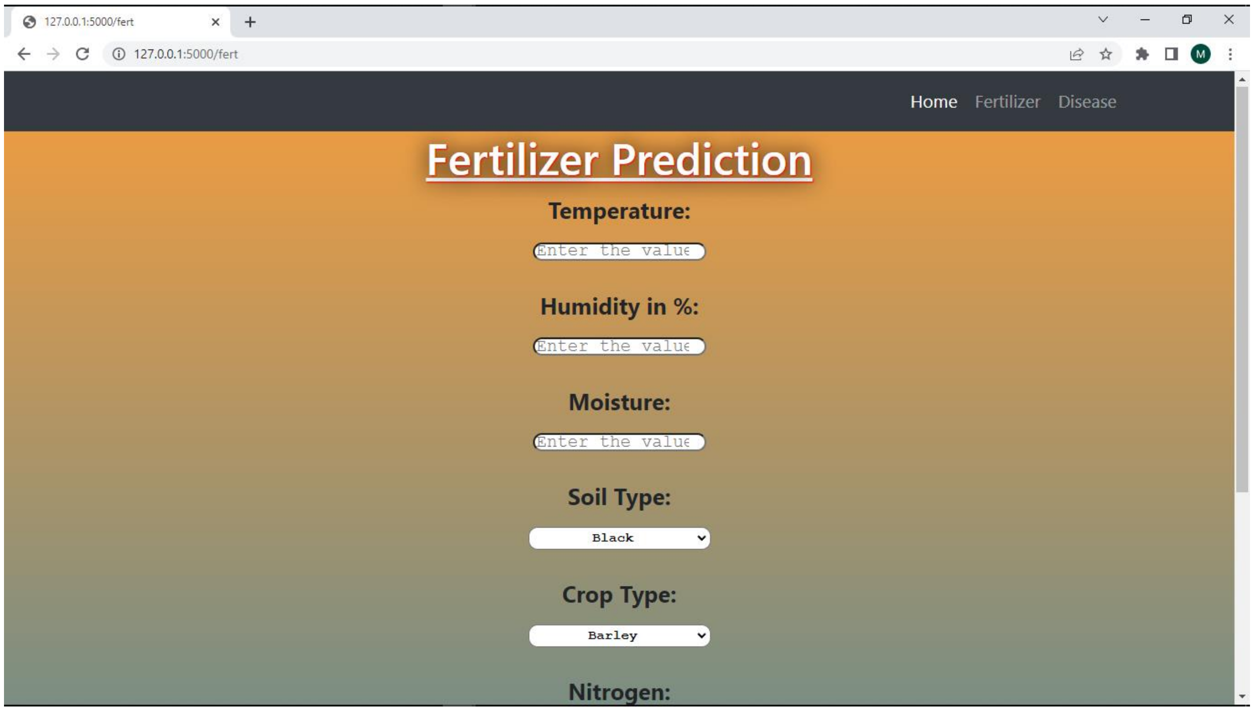
Upload Image Here To Identify the Plant info:

Choose...

Result: the fruit is:Potato___Early_blight Caution:


127.0.0.1:5000/fert1

Predicted Fertilizer is ['Urea']


127.0.0.1:5000/fert

Fertilizer Prediction

Temperature: Enter the value

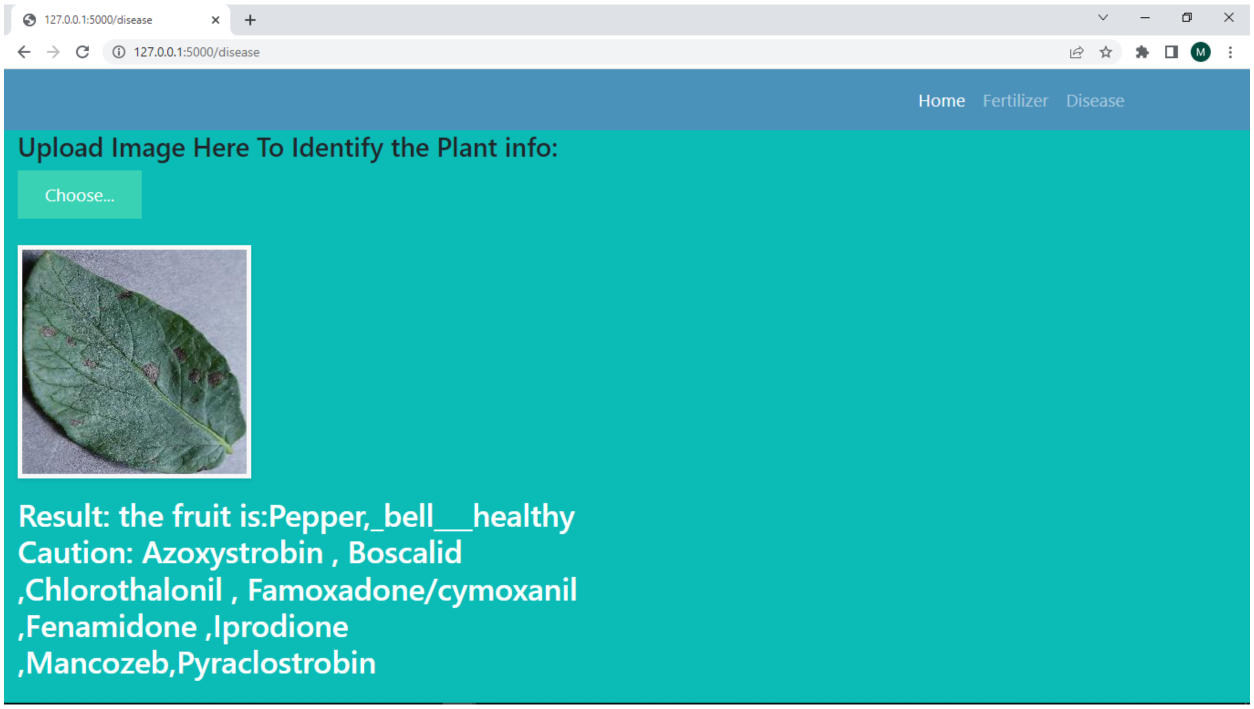
Humidity in %: Enter the value

Moisture: Enter the value

Soil Type: Black

Crop Type: Barley

Nitrogen:


127.0.0.1:5000/disease

Upload Image Here To Identify the Plant info:

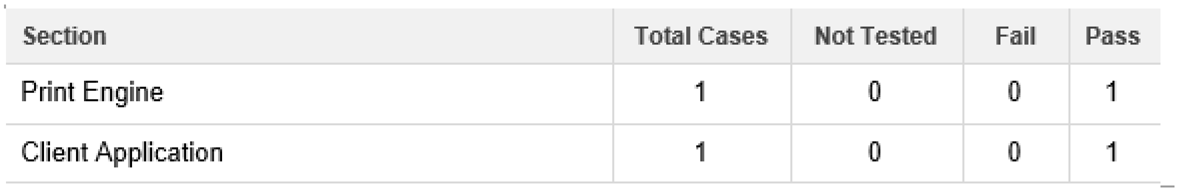
Result: the fruit is:Pepper,_bell___healthy
Caution: Azoxystrobin , Boscalid ,Chlorothalonil , Famoxadone/cymoxanil ,Fenamidone ,Iprodione ,Mancozeb,Pyraclostrobin

8.2.User Acceptance Test

Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	0	1	0	1
Duplicate	1	3	2	2	8
External	2	3	0	0	5
Fixed	4	4	4	4	16
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	7	10	7	7	31

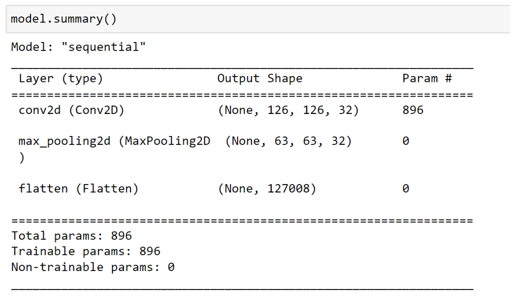
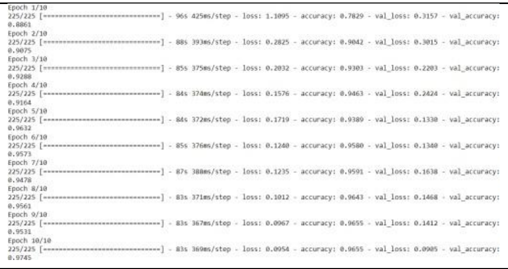
Test Case Analysis



Section	Total Cases	Not Tested	Fail	Pass
Print Engine	1	0	0	1
Client Application	1	0	0	1

9. RESULTS

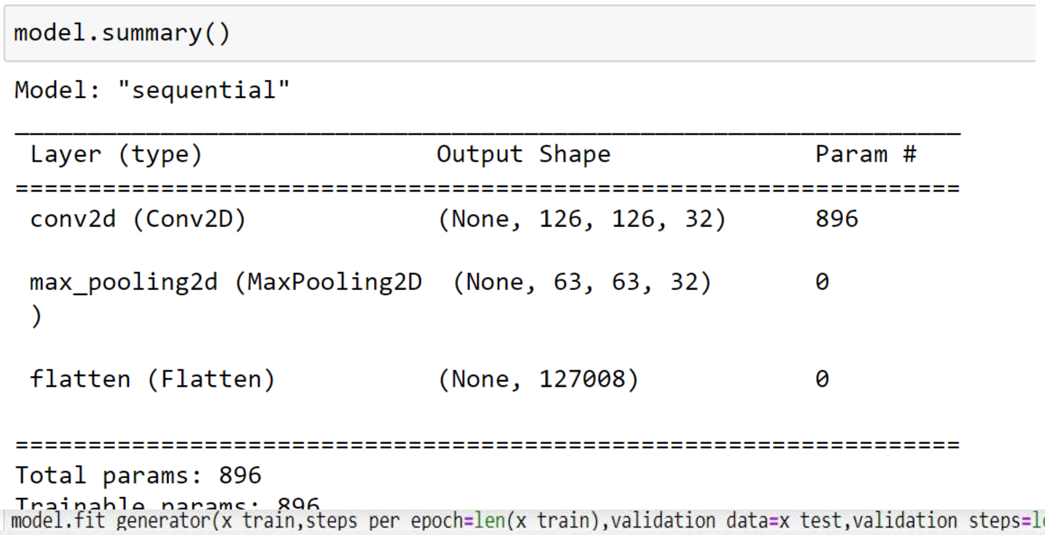
9.1. Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 896 Trainable params: 896 Non-trainable params: 0	
2.	Accuracy	Training Accuracy – 96.55 Validation Accuracy – 97.45	

```
model.summary()

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

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

 flatten (Flatten)           (None, 127008)            0

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

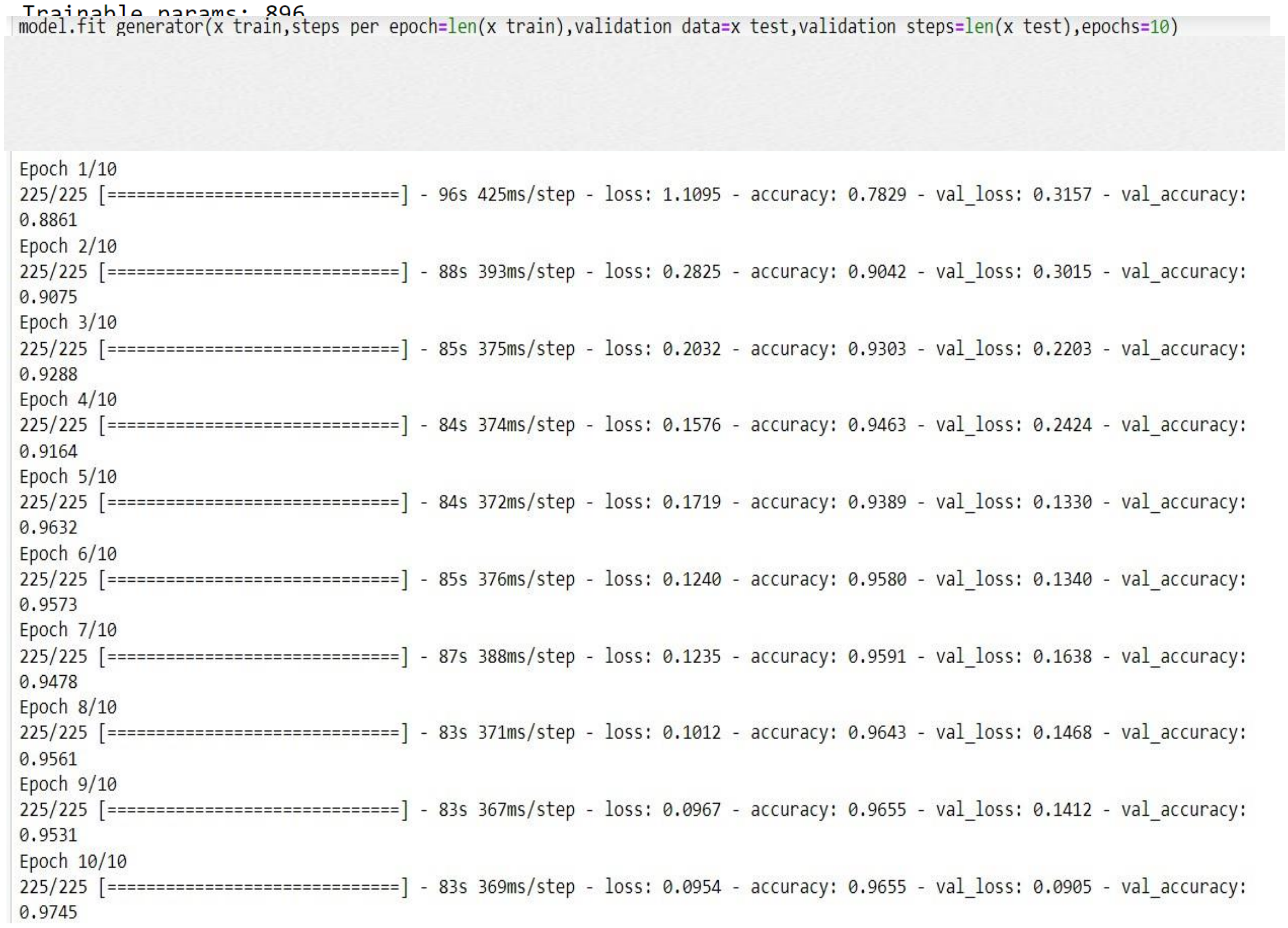
```
Epoch 1/10
225/225 [==============================] - 96s 425ms/step - loss: 1.1095 - accuracy: 0.7829 - val_loss: 0.3157 - val_accuracy:
0.8861
Epoch 2/10
225/225 [==============================] - 88s 393ms/step - loss: 0.2825 - accuracy: 0.9042 - val_loss: 0.3015 - val_accuracy:
0.9075
Epoch 3/10
225/225 [==============================] - 85s 375ms/step - loss: 0.2032 - accuracy: 0.9303 - val_loss: 0.2203 - val_accuracy:
0.9288
Epoch 4/10
225/225 [==============================] - 84s 374ms/step - loss: 0.1576 - accuracy: 0.9463 - val_loss: 0.2424 - val_accuracy:
0.9164
Epoch 5/10
225/225 [==============================] - 84s 372ms/step - loss: 0.1719 - accuracy: 0.9389 - val_loss: 0.1330 - val_accuracy:
0.9632
Epoch 6/10
225/225 [==============================] - 85s 376ms/step - loss: 0.1240 - accuracy: 0.9580 - val_loss: 0.1340 - val_accuracy:
0.9573
Epoch 7/10
225/225 [==============================] - 87s 388ms/step - loss: 0.1235 - accuracy: 0.9591 - val_loss: 0.1638 - val_accuracy:
0.9478
Epoch 8/10
225/225 [==============================] - 83s 371ms/step - loss: 0.1012 - accuracy: 0.9643 - val_loss: 0.1468 - val_accuracy:
0.9561
Epoch 9/10
225/225 [==============================] - 83s 367ms/step - loss: 0.0967 - accuracy: 0.9655 - val_loss: 0.1412 - val_accuracy:
0.9531
Epoch 10/10
225/225 [==============================] - 83s 369ms/step - loss: 0.0954 - accuracy: 0.9655 - val_loss: 0.0905 - val_accuracy:
0.9745
```

Model summary



```
model.summary()
```

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

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

 flatten (Flatten)           (None, 127008)            0

=================================================================
Total params: 896
Trainable params: 896
```



```
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 [==============================] - 96s 425ms/step - loss: 1.1095 - accuracy: 0.7829 - val_loss: 0.3157 - val_accuracy:
0.8861
Epoch 2/10
225/225 [==============================] - 88s 393ms/step - loss: 0.2825 - accuracy: 0.9042 - val_loss: 0.3015 - val_accuracy:
0.9075
Epoch 3/10
225/225 [==============================] - 85s 375ms/step - loss: 0.2032 - accuracy: 0.9303 - val_loss: 0.2203 - val_accuracy:
0.9288
Epoch 4/10
225/225 [==============================] - 84s 374ms/step - loss: 0.1576 - accuracy: 0.9463 - val_loss: 0.2424 - val_accuracy:
0.9164
Epoch 5/10
225/225 [==============================] - 84s 372ms/step - loss: 0.1719 - accuracy: 0.9389 - val_loss: 0.1330 - val_accuracy:
0.9632
Epoch 6/10
225/225 [==============================] - 85s 376ms/step - loss: 0.1240 - accuracy: 0.9580 - val_loss: 0.1340 - val_accuracy:
0.9573
Epoch 7/10
225/225 [==============================] - 87s 388ms/step - loss: 0.1235 - accuracy: 0.9591 - val_loss: 0.1638 - val_accuracy:
0.9478
Epoch 8/10
225/225 [==============================] - 83s 371ms/step - loss: 0.1012 - accuracy: 0.9643 - val_loss: 0.1468 - val_accuracy:
0.9561
Epoch 9/10
225/225 [==============================] - 83s 367ms/step - loss: 0.0967 - accuracy: 0.9655 - val_loss: 0.1412 - val_accuracy:
0.9531
Epoch 10/10
225/225 [==============================] - 83s 369ms/step - loss: 0.0954 - accuracy: 0.9655 - val_loss: 0.0905 - val_accuracy:
0.9745
```

10.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.

DISADVANTAGES

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

11. Conclusion

The key objective of this work is to analyze di_erent machine learning techniques widely used in the prediction of plant diseases and how advancement can be made in the future in this technique to achieve higher accuracy, robustness, cost-e_cient disease prediction system. The steps involved in image processing techniques like pre-processing, segmentation, extracting feature and classi_cation based on symptoms in the plant are discussed in this survey. Machine learning techniques play a key role in the machine vision system. In the future, deep learning framework can be used for disease prediction system. Integrating image processing techniques and deep learning techniques proved to be more potential in disease prediction system. Still, more investigations have to be made in these techniques for achieving better prediction system. If disease is detected, then it will tell what type of disease it is and will provide solutions such as pesticides or chemicals to cure that disease.

12.FUTURE SCOPE

Agriculture is one among the most important sector of Indian Economy. More than 50% of population depends on agriculture as their source of income. Growing crops over thousands of years without caring about replenishing has led to depletion and exhaustion of soil nutrients resulting in their low productivity. To improve the production, chemical fertilizers were added. But excessive use of these not only makes the plants dependent on artificial fertilizers but also erodes the natural quality of land. It is therefore important to make sure that only the sufficient amount of fertilizers is added so that yield can be increased and at the same time, the natural quality of soil remain intact. For this, it is essential to know the soil nutrient levels. Our project aims at finding the soil nutrient richness and predict the fertility of a given soil sample in real time. Based on the result obtained, the system will also be giving recommendation on the type of fertilizer and in what quantity it is to be used in the soil sample so that the yield can be maximized

13.APPENDIX

Source Code

Python code:

```
import numpy as np

import os

import pandas as pd

fromtensorflow.keras.models import load_model

fromtensorflow.keras.preprocessing import image

fromflask import Flask,render_template,request

import pickle

import utils

fromutils.model import ResNet9

fromutils.disease import disease_dic

import torch


disease_classes = ['Apple___Apple_scab',

            'Apple___Black_rot',

            'Apple___Cedar_apple_rust',

            'Apple___healthy',

            'Blueberry___healthy',

            'Cherry_(including_sour)___Powdery_mildew',

            'Cherry_(including_sour)___healthy',
            'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',
            'Corn_(maize)___Common_rust_',
            'Corn_(maize)___Northern_Leaf_Blight',
            'Corn_(maize)___healthy',
            'Grape___Black_rot',
            'Grape___Esca_(Black_Measles)',
            'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)',
            'Grape___healthy',
            'Orange___Haunglongbing_(Citrus_greening)',
            'Peach___Bacterial_spot',
            'Peach___healthy',
            'Pepper,_bell___Bacterial_spot',
            'Pepper,_bell___healthy',
            'Potato___Early_blight',
            'Potato___Late_blight',
            'Potato___healthy',
            'Raspberry___healthy',
            'Soybean___healthy',
            'Squash___Powdery_mildew',
            'Strawberry___Leaf_scorch',
            'Strawberry___healthy',

            'Tomato___Bacterial_spot',

            'Tomato___Early_blight',

            'Tomato___Late_blight',

            'Tomato___Leaf_Mold',

            'Tomato___Septoria_leaf_spot',

            'Tomato___Spider_mites Two-spotted_spider_mite',

            'Tomato___Target_Spot',

            'Tomato___Tomato_Yellow_Leaf_Curl_Virus',

            'Tomato___Tomato_mosaic_virus',

            'Tomato___healthy']



disease_model_path = 'models/plant_disease_model.pth'

disease_model = ResNet9(3, len(disease_classes))

disease_model.load_state_dict(torch.load(

  disease_model_path, map_location=torch.device('cpu')))

disease_model.eval()
```

```
def predict_image(img, model=disease_model):
    """
    Transforms image to tensor and predicts disease label
    :params: image
    :return: prediction (string)
    """
    transform = transforms.Compose([
        transforms.Resize(256),
        transforms.ToTensor(),
    ])
    image = Image.open(io.BytesIO(img))
    img_t = transform(image)
    img_u = torch.unsqueeze(img_t, 0)

    # Get predictions from model
    yb = model(img_u)
    # Pick index with highest probability
    _, preds = torch.max(yb, dim=1)
    prediction = disease_classes[preds[0].item()]
    # Retrieve the class label
    return prediction



app=Flask(__name__)


model = pickle.load(open('classifier.pkl','rb'))

ferti = pickle.load(open('fertilizer.pkl','rb'))


##vmodel = load_model("vegetable.h5")

fmodel = load_model("fruit.h5")


@app.route('/')

def login():

    return render_template("login.html")


# render disease prediction input page

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

def home():

    return render_template('index1.html')
```

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

def predict():

  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=(64,64))

    x = image.img_to_array(img)

    x = np.expand_dims(x,axis=0)

    pred = np.argmax(fmodel.predict(x),axis=1)

    index = ['Apple1', 'Apple2', 'corn1', 'corn2', 'peach1', 'peach2']

    text = "the animal is:"+str(index[pred[0]])

  return text


@app.route('/fert')

def fert():

  return render_template('index.html')


@app.route('/fert1',methods=['POST'])
```

```
def fert1():
    temp = request.form.get('temp')
    humi = request.form.get('humid')
    mois = request.form.get('mois')
    soil = request.form.get('soil')
    crop = request.form.get('crop')
    nitro = request.form.get('nitro')
    pota = request.form.get('pota')
    phosp = request.form.get('phos')
    input = [int(temp),int(humi),int(mois),int(soil),int(crop),int(nitro),int(pota),int(phosp)]

    res = ferti.classes_[model.predict([input])]

    return render_template('fertilizer-result.html',x = ('Predicted Fertilizer is 
{}'.format(res)))



@app.route('/disease1', methods=['GET', 'POST'])
def disease1():
    title = 'Disease Detection'
```

```
  if request.method == 'POST':
    if 'file' not in request.files:
      return redirect(request.url)
    file = request.files.get('file')
    if not file:
      return render_template('disease.html', title=title)
    try:
      img = file.read()

      prediction = predict_image(img)

      prediction = Markup(str(disease_dic[prediction]))
      return render_template('disease-result.html', prediction=prediction, title=title)
    except:
      pass
  return render_template('disease.html', title=title)


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

HTML CODING;

```
{% extends "login new.html" %} {% block body %}


<div class="bg"></div>

<div class="bg bg2"></div>

<div class="bg bg3"></div>

<div class="content">

  <h1></h1>

</div>


<style>

 html, body {

  background: rgb(2,0,36);

background: linear-gradient(11deg, rgba(2,0,36,1) 0%, rgba(15,121,9,1) 0%, rgba(12,141,63,1) 42%, rgba(0,212,255,1) 100%);

 }

</style>

<style>
```

```
 label {


font-family: 'Raleway', sans-serif;

font-size: 1.1875rem;

font-weight: 400;

letter-spacing: 0.1rem;

font-style: normal;

text-transform: capitalize;

color: #FFFFFF;

background-color: #5ca8f0;

border-radius: 0.625rem;

-webkit-border-radius: 0.625rem;

-moz-border-radius: 0.625rem;

padding: 0.3rem 0.4rem;

border-style: solid;

border-width: 0rem;

border-color: #EF2D56;

-webkit-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);

-moz-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);

-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);
}

</style>

<br /><br />

 <h2 style="text-align: center; margin: 0px; color: black"><b>Wanna know which disease
Caught to your plant ?</b></h2>

 <br />

 <br>


<div style="

  width: 350px;

  height: 50rem;

  margin: 0px auto;

  color: black;

  border-radius: 25px;

  padding: 10px 10px;

  font-weight: bold;

 ">


<style>
```

```
 form{


font-family: 'Raleway', sans-serif;

font-size: 1.1875rem;

font-weight: 400;

letter-spacing: 0.1rem;

font-style: normal;

text-transform: capitalize;

color: #FFFFFF;

background-color: #5ca8f0;

border-radius: 0.625rem;

-webkit-border-radius: 0.625rem;

-moz-border-radius: 0.625rem;

padding: 0.3rem 0.4rem;

border-style: solid;

border-width: 0rem;

border-color: #EF2D56;

-webkit-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);

-moz-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);

-box-shadow: 5px 7px 8px -1px rgba(51,51,51,0.51);
}
</style>


 <form class="form-signin" method=post enctype=multipart/form-data>

  <h2 class="h4 mb-3 font-weight-normal">Upload Your Image</h2>
  <input type="file" name="file" class="form-control-file" id="inputfile"
onchange="preview_image(event)" style="font-weight: bold;">
  <br>
  <br>
  <img id="output-image" class="rounded mx-auto d-block" />
  <button class="btn btn-lg btn-primary btn-block" type="submit" style="font-weight:
bold;">Predict</button>


 </form>

</div>
```

```
<script type="text/javascript">
 function preview_image(event) {
  var reader = new FileReader();
  reader.onload = function () {
   var output = document.getElementById('output-image')
   output.src = reader.result;
  }
  reader.readAsDataURL(event.target.files[0]);
 }
</script>


</div>
{% endblock %}
```

Project Demo Link:

https://drive.google.com/file/d/1RGkjglYaB_YAewb5ynlviQJQAiV2M6_w/view?usp=drivesdk