Fertilizers Recommendation System For Disease Prediction

IBM Project Report

Team Id : PNT2022TMID48803

Team Members

1. JANANI PRIYA M
2. DIVYA T
3. PANDEESWARI S
4. CHIBI S
5. ANUSHAMEENAMBIGAI E

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project 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 prediced diseases.

2. LITERATURE SURVEY

2.1 Existing problem

1.Detection of Leaf Diseases and Classification using Digital Image Processing International Conference on Innovations in Information, Embedded and Communication Systems(ICIIECS), IEEE, 2017.

Advantages: The system detects the diseases on citrus leaves with 90% accuracy.

Disadvantages: System only able to detect the disease from citrus leaves.

The main objective of this paper is image analysis & classification techniques for detection of leaf diseases and classification. The leaf image is firstly preprocessed and then does the further work. K-Means Clustering used for image segmentation and then system extract the GLCM features from disease detected images. The disease classification done through the SVM classifier.

Algorithm used: Gray-Level Co-Occurrence Matrix (GLCM) features, SVM, K-Means Clustering .

2.Semi-automatic leaf disease detection and classification system for soybean culture IET Image Processing, 2018

Advantages: The system helps to compute the disease severity.

Disadvantages: The system uses leaf images taken from an online dataset, so cannot implement in real time.

This paper mainly focuses on the detecting and classifying the leaf disease of soybean plant. Using SVM the proposed system classifies the leaf disease in 3 classes like i.e. downy mildew, frog eye, reported and septoria leaf blight etc. The proposed system gives maximum average classification accuracy is ~90% using a big dataset of 4775 images.

Algorithm used: SVM.

2.2 Reference

[1] Semi-automatic leaf disease detection and classification system for soybean culture IET Image Processing, 2018

[2] Cloud Based Automated Irrigation And Plant Leaf Disease Detection System Using An Android Application. International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017.

[3] Ms. Kiran R. Gavhale, Ujwalla Gawande, Plant Leaves Disease detection using Image Processing Techniques, January 2014.

https://www.researchgate.net/profile/UjwallaGawande/publication/314436486_An_Overview_of_the

_Research_on_Plant_Leaves_Disease_detection_using_Image_Processing_Techniques/links/5d37106 64585153e591a3d20/An-Overviewof-the-Research-on-Plant-Leaves-Diseae

detection-using-Image-ProcessingTechniques.pdf

[4] Duan Yan-e, Design of Intelligent Agriculture Management Information System Based on

IOTǁ, IEEE,4th, Fourth International reference on Intelligent Computation Technology and Automation, 2011

https://ieeexplore.ieee.org/document/5750779

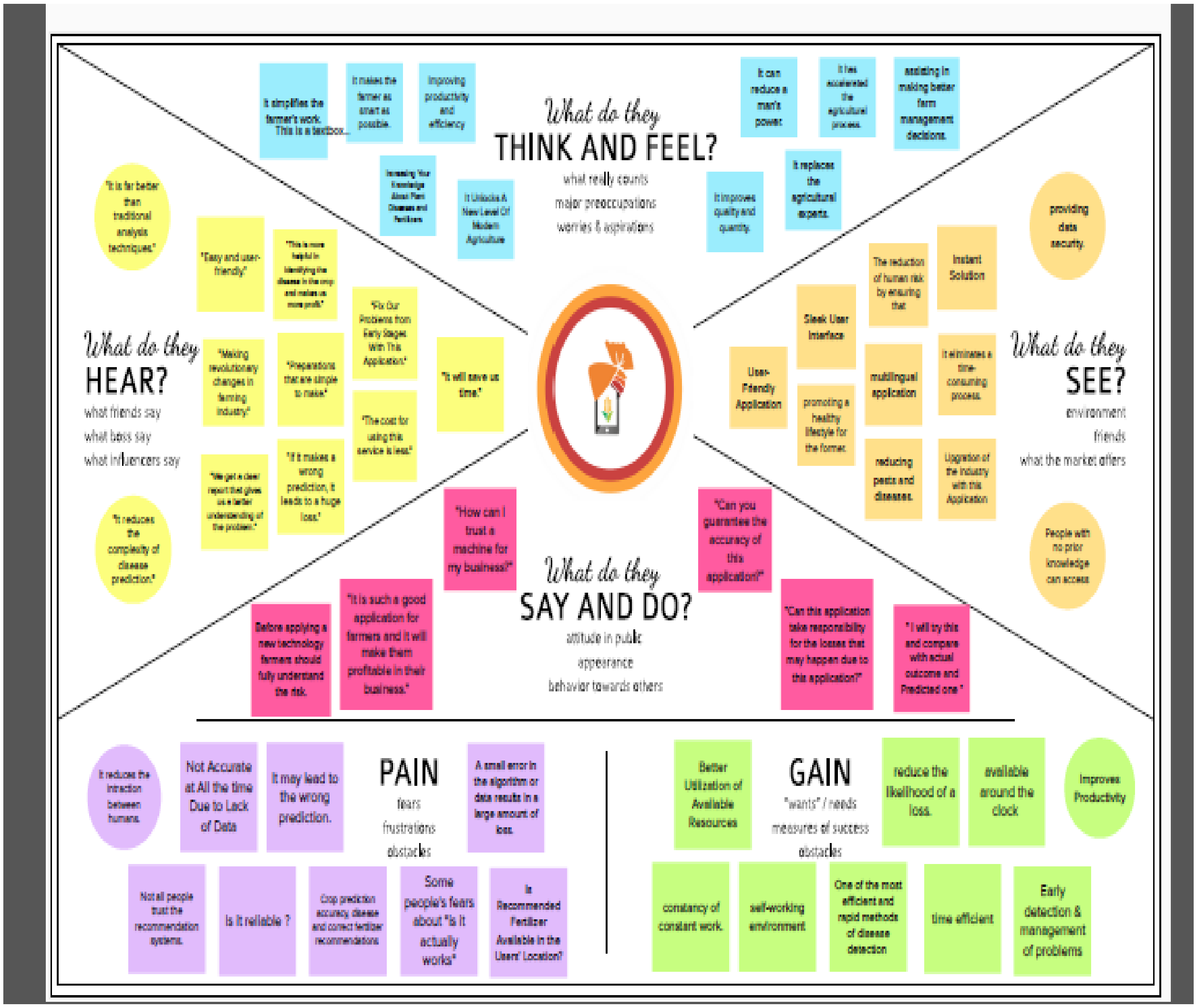
2.3 Problem Statement Definition

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.

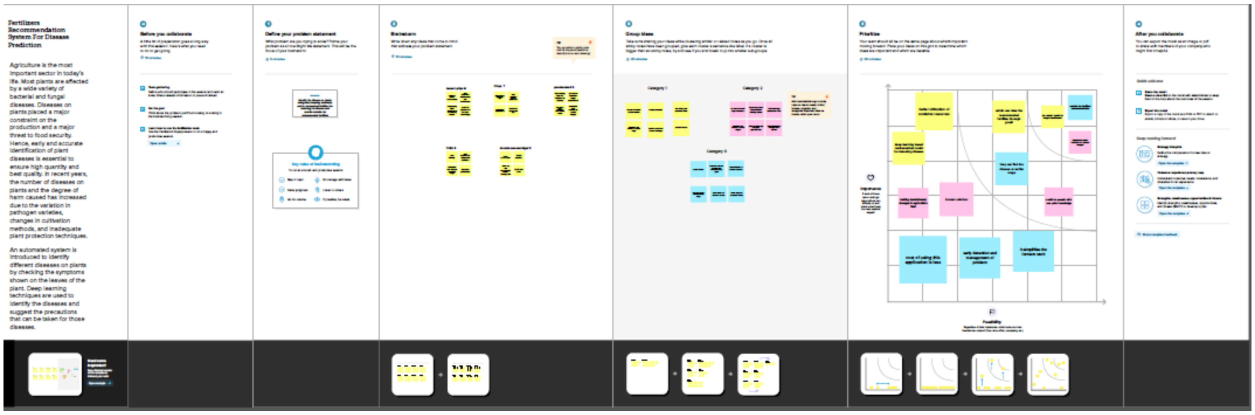
3. IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canvas



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
It simplifies the farmer's work. This is a textbox...
It makes the farmer as smart as possible.
Improving productivity and efficiency
Increasing Your Knowledge About Plant Diseases and Fertilizers
It Unlocks A New Level Of Modern Agriculture
It can reduce a man's power.
It has accelerated the agricultural process.
assisting in making better farm management decisions.
It improves quality and quantity.
It replaces the agricultural experts.
What do they HEAR?
what friends say
what boss say
what influencers say
"It is far better than traditional analysis techniques"
"Easy and user-friendly"
"This is more helpful in identifying the disease in the crop and makes us more profit."
"Fix Our Problems from Early Stages With This Application"
"Making revolutionary changes in farming industry"
"Preparations that are simple to make"
"It will save us time."
"The cost for using this service is less."
"We get a clear report that gives us a better understanding of the problem."
"If it makes a wrong prediction, it leads to a huge loss."
"It reduces the complexity of disease prediction."
What do they SEE?
environment
friends
what the market offers
providing data security.
The reduction of human risk by ensuring that
Instant Solution
Sleek User Interface
User-Friendly Application
multilingual application
It eliminates a time-consuming process.
promoting a healthy lifestyle for the farmer.
reducing pests and diseases.
Upgradation of the Industry with this Application
People with no prior knowledge can access
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
"How can I trust a machine for my business?"
"Can you guarantee the accuracy of this application?"
Before applying a new technology farmers should fully understand the risk.
"It is such a good application for farmers and it will make them profitable in their business."
"Can this application take responsibility for the losses that may happen due to this application?"
" I will try this and compare with actual outcome and Predicted one "
PAIN
fears
frustrations
obstacles
It reduces the intraction between humans.
Not Accurate at All the time Due to Lack of Data
It may lead to the wrong prediction.
A small error in the algorithm or data results in a large amount of loss.
Not all people trust the recommendation systems.
Is it reliable ?
Crop prediction accuracy, disease and correct fertilizer recommendations
Some people's fears about "Is it actually works"
Is Recommended Fertilizer Available in the Users' Location?
GAIN
"wants" / needs
measures of success
obstacles
Better Utilization of Available Resources
reduce the likelihood of a loss.
available around the clock
Improves Productivity
constancy of constant work.
self-working environment
One of the most efficient and rapid methods of disease detection
time efficient
Early detection & management of problems

3.2 Ideation & Brainstorming

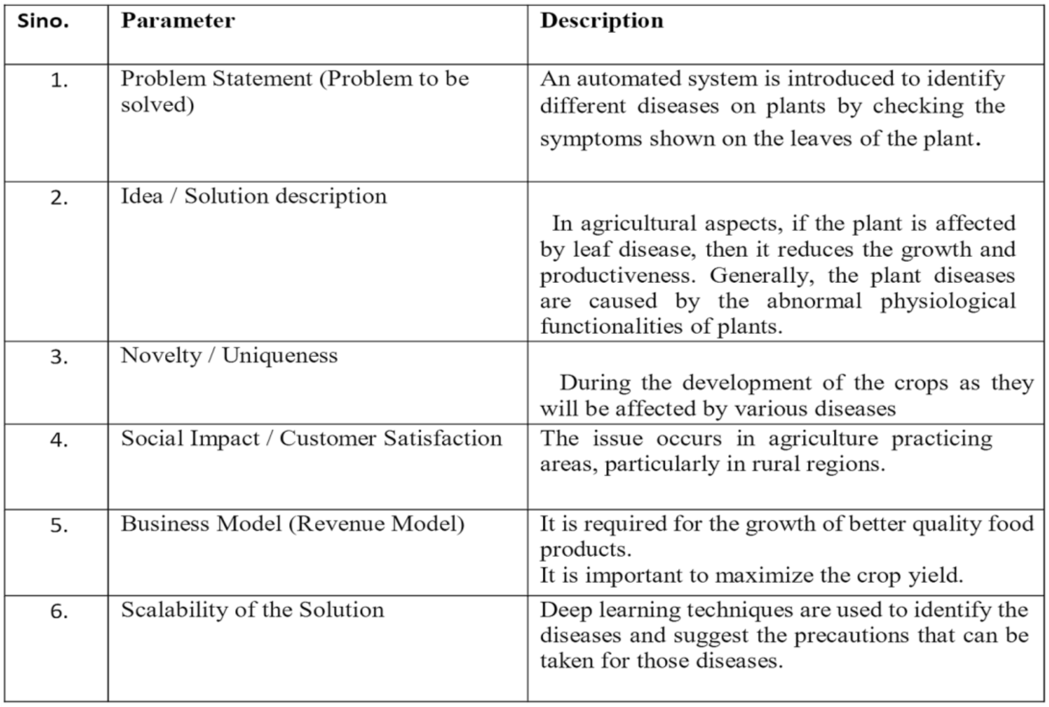


Fertilizers Recommendation System For Disease Prediction

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.

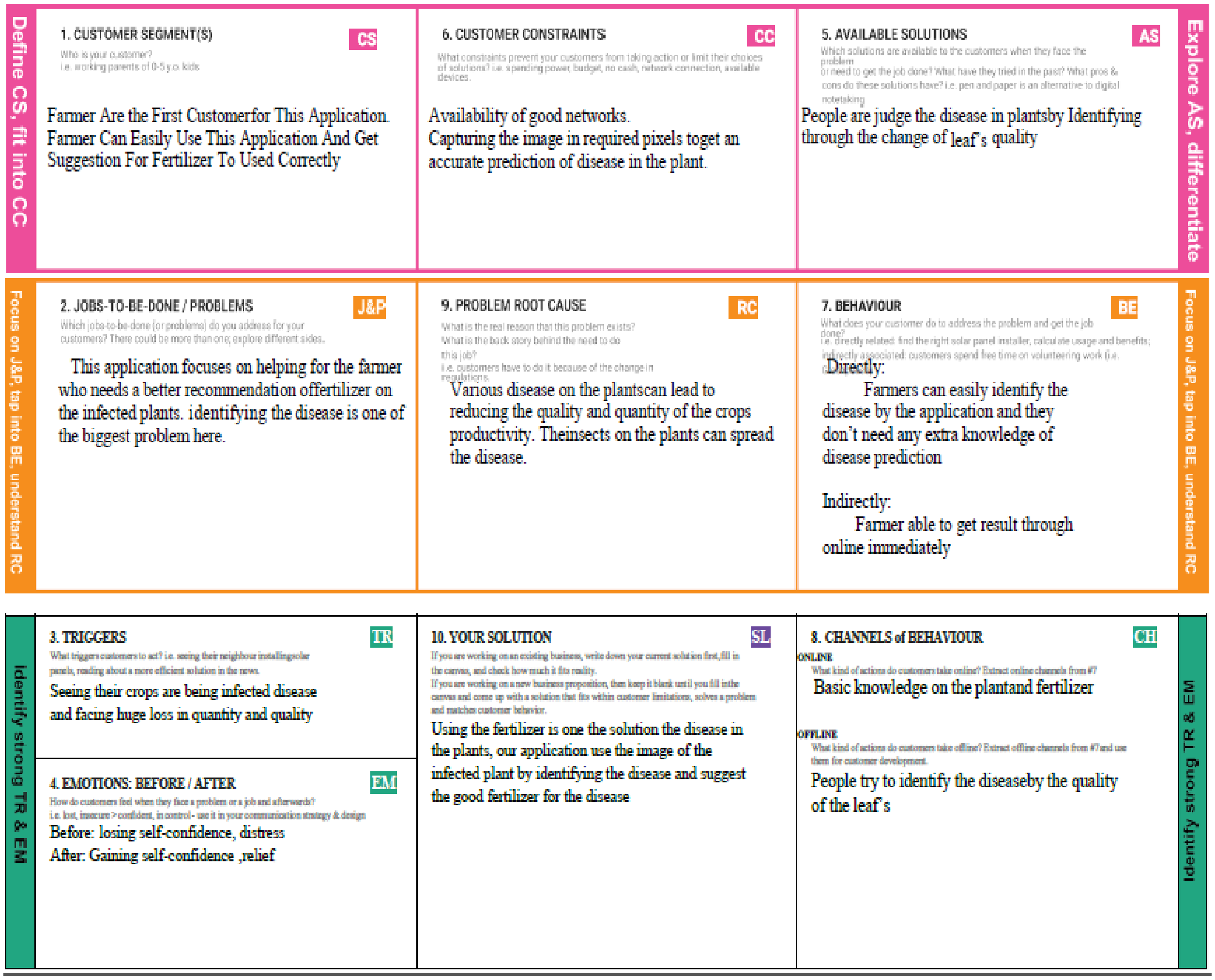
An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

3.3 Proposed Solution



Sino.	Parameter	Description
1.	Problem Statement (Problem to be solved)	An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.
2.	Idea / Solution description	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.
3.	Novelty / Uniqueness	During the development of the crops as they will be affected by various diseases
4.	Social Impact / Customer Satisfaction	The issue occurs in agriculture practicing areas, particularly in rural regions.
5.	Business Model (Revenue Model)	It is required for the growth of better quality food products. It is important to maximize the crop yield.
6.	Scalability of the Solution	Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

3.4 Problem Solution fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

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

Farmer Are the First Customerfor This Application. Farmer Can Easily Use This Application And Get Suggestion For Fertilizer To Used Correctly

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.

Availability of good networks.
Capturing the image in required pixels toget an accurate prediction of disease in the plant.

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

People are judge the disease in plantsby Identifying through the change of leaf's quality

Explore AS, differentiate

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.

This application focuses on helping for the farmer who needs a better recommendation offertilizer on the infected plants. identifying the disease is one of the biggest problem here.

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.

Various disease on the plantscan lead to reducing the quality and quantity of the crops productivity. Theinsects on the plants can spread the disease.

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.

Directly:
Farmers can easily identify the disease by the application and they don't need any extra knowledge of disease prediction

Indirectly:
Farmer able to get result through online immediately

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

Identify strong TR & EM

3. TRIGGERS TR

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

Seeing their crops are being infected disease and facing huge loss in quantity and quality

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

Before: losing self-confidence, distress
After: Gaining self-confidence ,relief

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 inthe canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior.

Using the fertilizer is one the solution the disease in the plants, our application use the image of the infected plant by identifying the disease and suggest the good fertilizer for the disease

8. CHANNELS of BEHAVIOUR CH

ONLINE
What kind of actions do customers take online? Extract online channels from #7

Basic knowledge on the plantand fertilizer

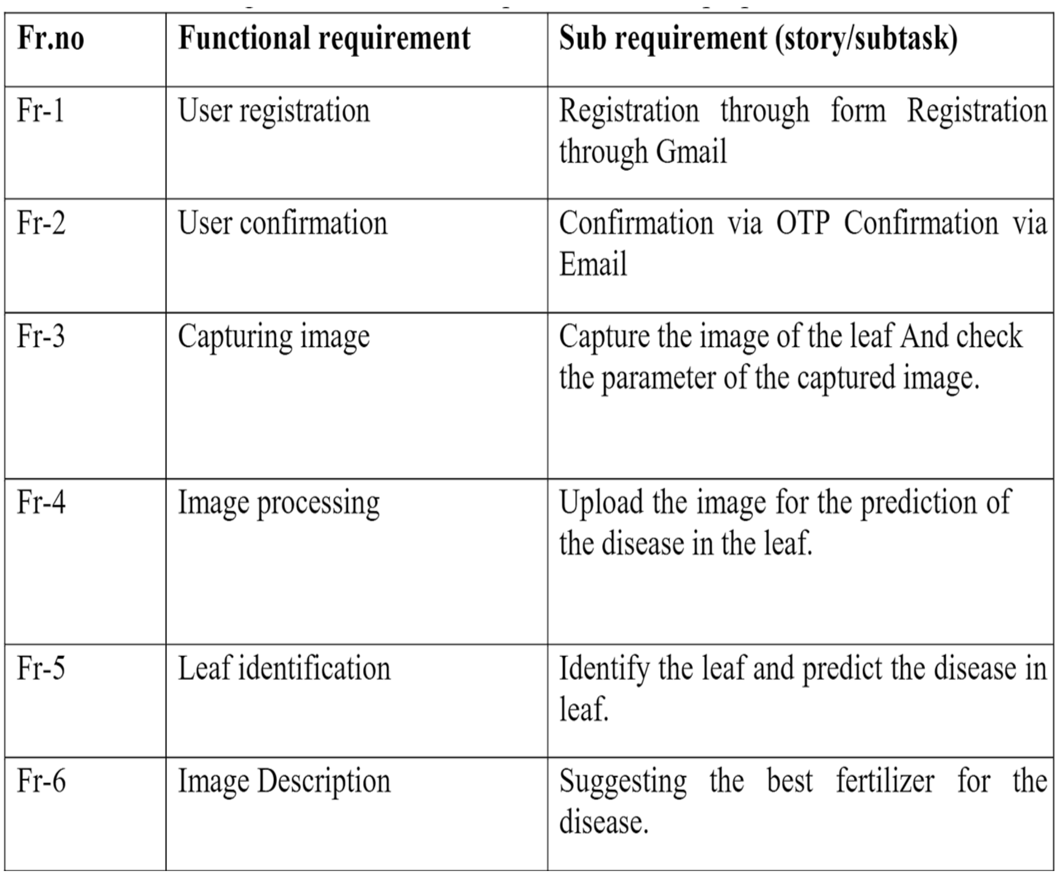
OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7and use them for customer development.

People try to identify the diseaseby the quality of the leaf's

Identify strong TR & EM

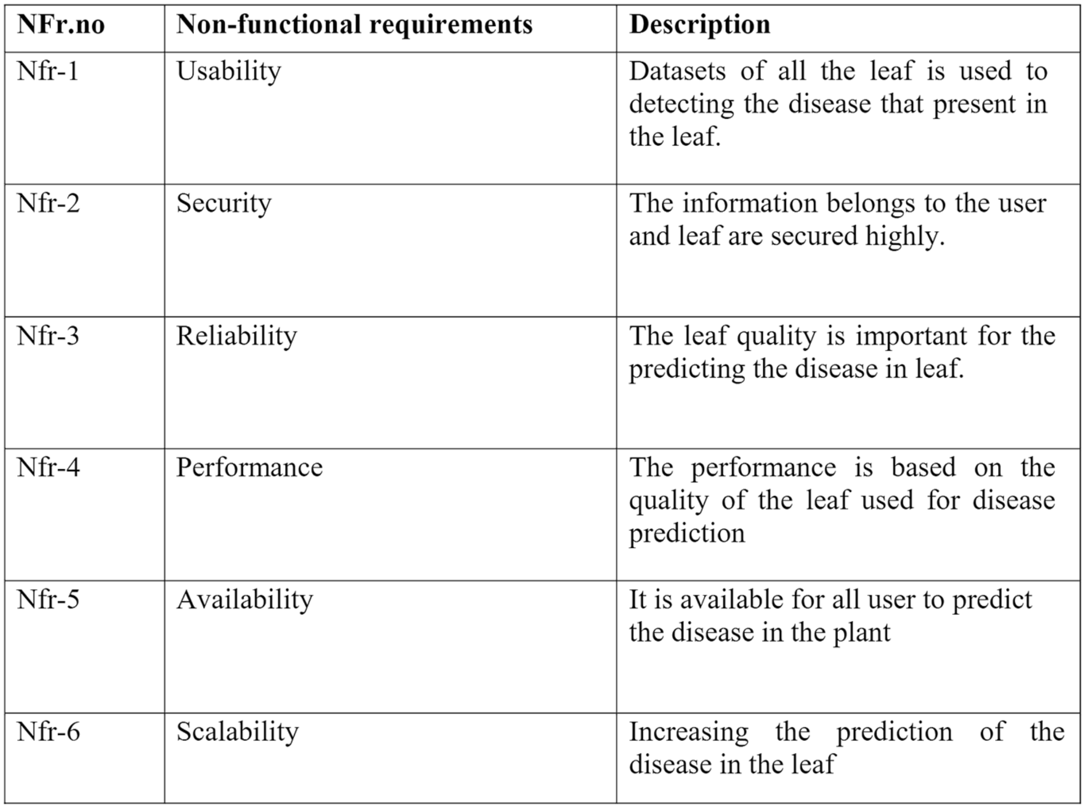
4. REQUIREMENT ANALYSIS

4.1 Functional requirement



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

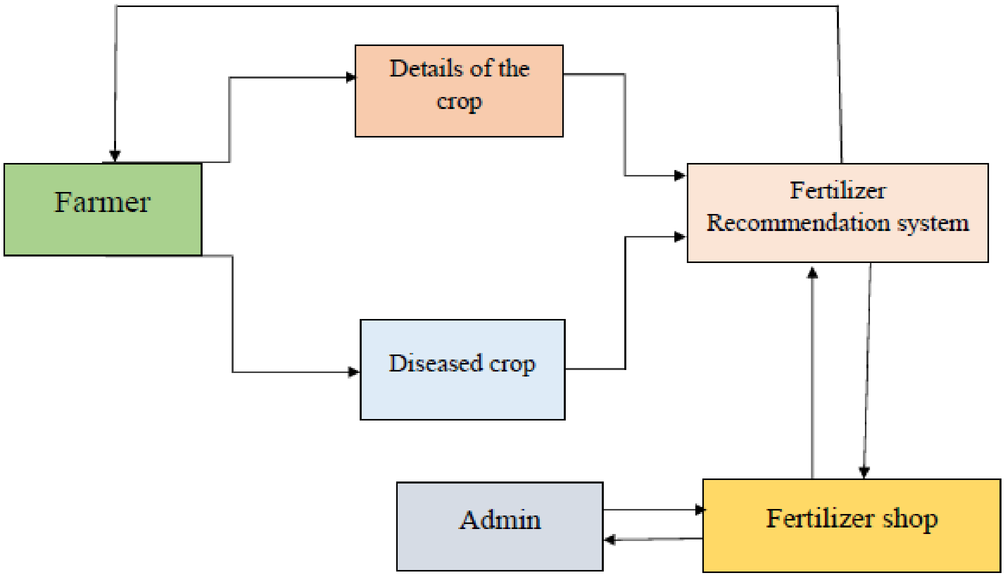
4.2 Non-Functional requirements



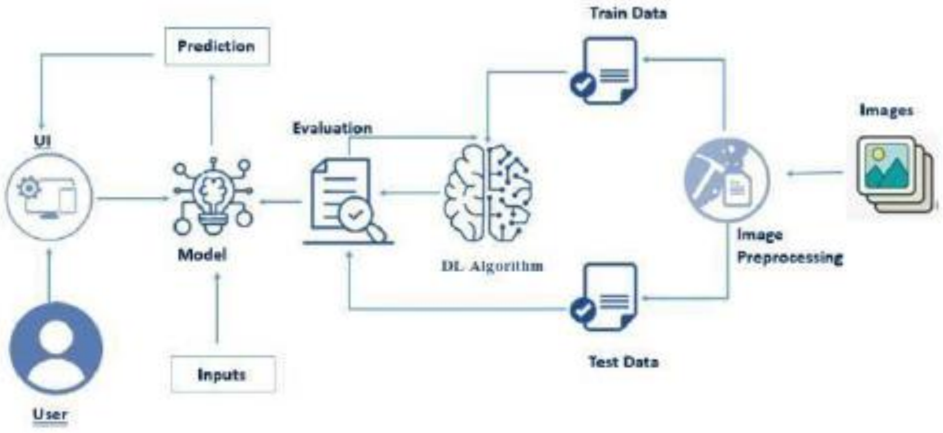
NFr.no	Non-functional requirements	Description
Nfr-1	Usability	Datasets of all the leaf is used to detecting the disease that present in the leaf.
Nfr-2	Security	The information belongs to the user and leaf are secured highly.
Nfr-3	Reliability	The leaf quality is important for the predicting the disease in leaf.
Nfr-4	Performance	The performance is based on the quality of the leaf used for disease prediction
Nfr-5	Availability	It is available for all user to predict the disease in the plant
Nfr-6	Scalability	Increasing the prediction of the disease in the leaf

5. PROJECT DESIGN

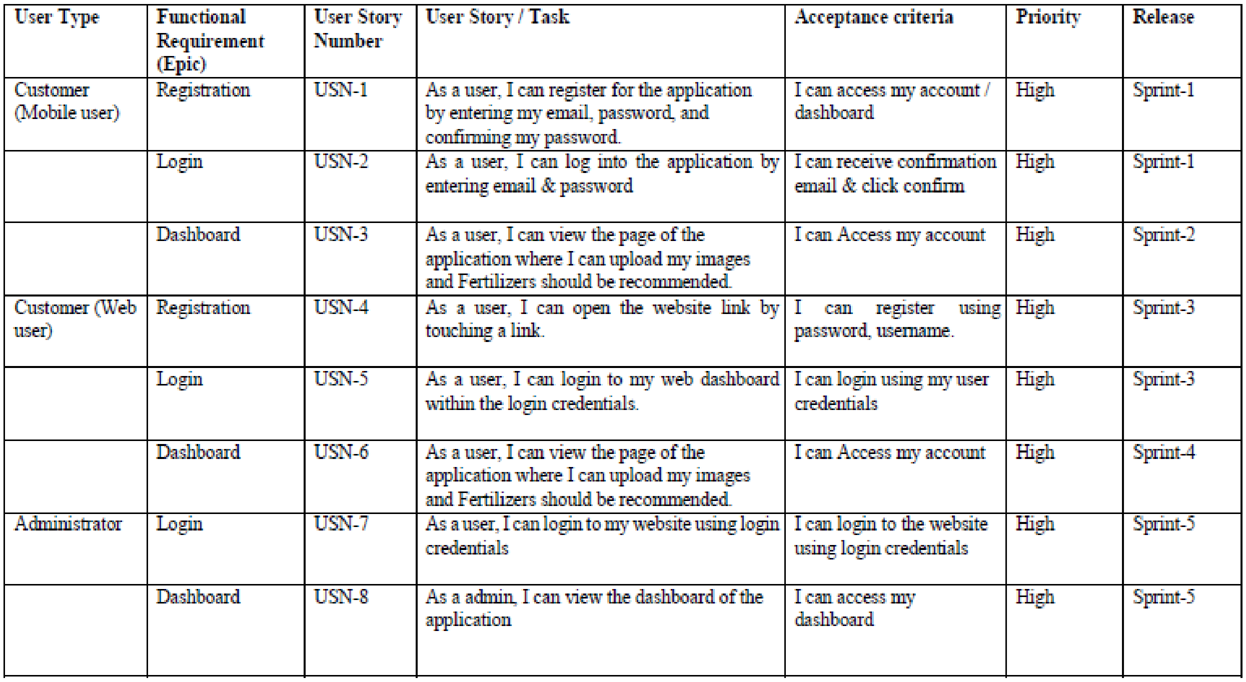
5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard	USN-3	As a user, I can view the page of the application where I can upload my images and Fertilizers should be recommended.	I can Access my account	High	Sprint-2
Customer (Web user)	Registration	USN-4	As a user, I can open the website link by touching a link.	I can register using password, username.	High	Sprint-3
	Login	USN-5	As a user, I can login to my web dashboard within the login credentials.	I can login using my user credentials	High	Sprint-3
	Dashboard	USN-6	As a user, I can view the page of the application where I can upload my images and Fertilizers should be recommended.	I can Access my account	High	Sprint-4
Administrator	Login	USN-7	As a user, I can login to my website using login credentials	I can login to the website using login credentials	High	Sprint-5
	Dashboard	USN-8	As a admin, I can view the dashboard of the application	I can access my dashboard	High	Sprint-5

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

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 register by entering my email ,phone number, date of birth, password and confirm password	3	High	Jananipriya M Pandeeswari S Chibi S Divya T Anushameenambigai E
		USN-2	As a user, I will receive confirmation message in my email once I have registered or OTP will be sent	2	High	Jananipriya M Pandeeswari S Chibi S Divya T Anushameenambigai E
Sprint-2	login	USN-3	Enter the password and mail ID to login the dashboard.	2	medium	Jananipriya M Pandeeswari S Chibi S Divya T Anushameenambigai E

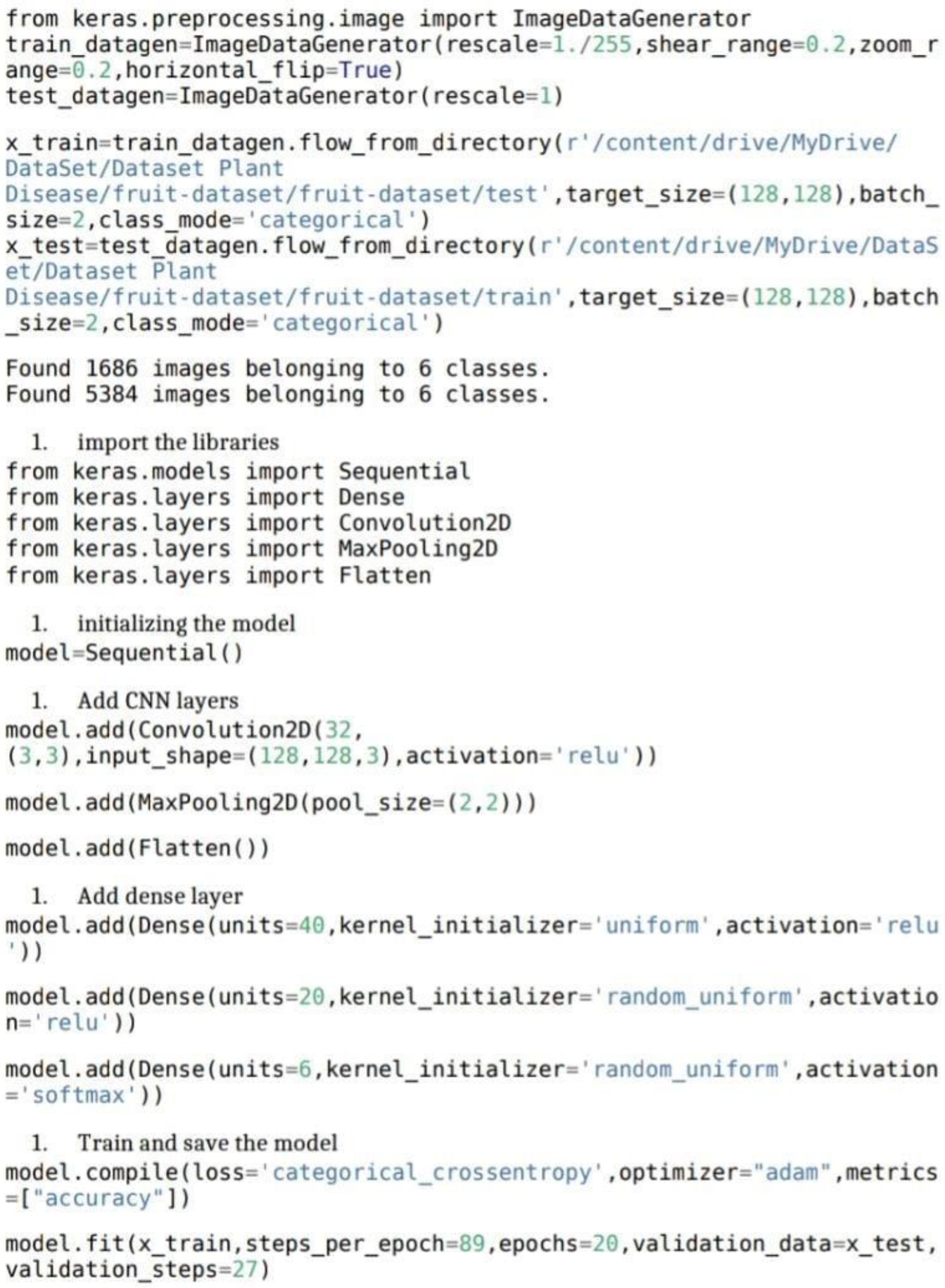
Sprint-3	Dashboard	USN-4	As a user, I can log in by entering email & password	2	medium	Jananipriya M Pandeeswari S Chibi S Divya T Anushameenambigai E
Sprint-4	Forgot password	USN-5	Suppose a user forgot password by clicking forgot password and OTP send to my number or mail.	3	medium	Jananipriya M Pandeeswari S Chibi S Divya T Anushameenambigai E
Sprint-5	Professional responsible	USN-6	As a customer care executive I'm the responsible for communicating the how's and why's regarding service exceptions within a company.	2	medium	Jananipriya M Pandeeswari S Chibi S Divya T Anushameenambigai E
Sprint-6	Data collection	USN-7	As an admin ,I can upload the data set to train the device.	1	low	Jananipriya M Pandeeswari S Chibi S Divya T Anushameenambigai E
		USN-8	Dealing with queries on the phone and by email. Arranging post and deliveries	1	medium	Jananipriya M Pandeeswari S Chibi S Divya T Anushameenambigai E

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Point Completed	Sprint Release Date
Sprint 1	20	4 days	28 Oct 2022	30 Oct 2022	10	30 Oct 2022
Sprint 2	20	5 days	31 Oct 2022	02 Nov 2022	10	02 Nov 2022
Sprint 3	20	4 days	03 Nov 2022	06 Nov 2022	10	06 Nov 2022
Sprint 4	20	6 days	07 Nov 2022	10 Nov 2022	10	10 Nov 2022

7. CODING & SOLUTIONING

7.1 Feature 1



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

```
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/
DataSet/Dataset Plant
Disease/fruit-dataset/fruit-dataset/test',target_size=(128,128),batch_
size=2,class_mode='categorical')
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/DataS
et/Dataset Plant
Disease/fruit-dataset/fruit-dataset/train',target_size=(128,128),batch
_size=2,class_mode='categorical')
```

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

1. import the libraries

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

1. initializing the model

```
model=Sequential()
```

1. Add CNN layers

```
model.add(Convolution2D(32,
(3,3),input_shape=(128,128,3),activation='relu'))
```

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Flatten())
```

1. Add dense layer

```
model.add(Dense(units=40,kernel_initializer='uniform',activation='relu
'))
```

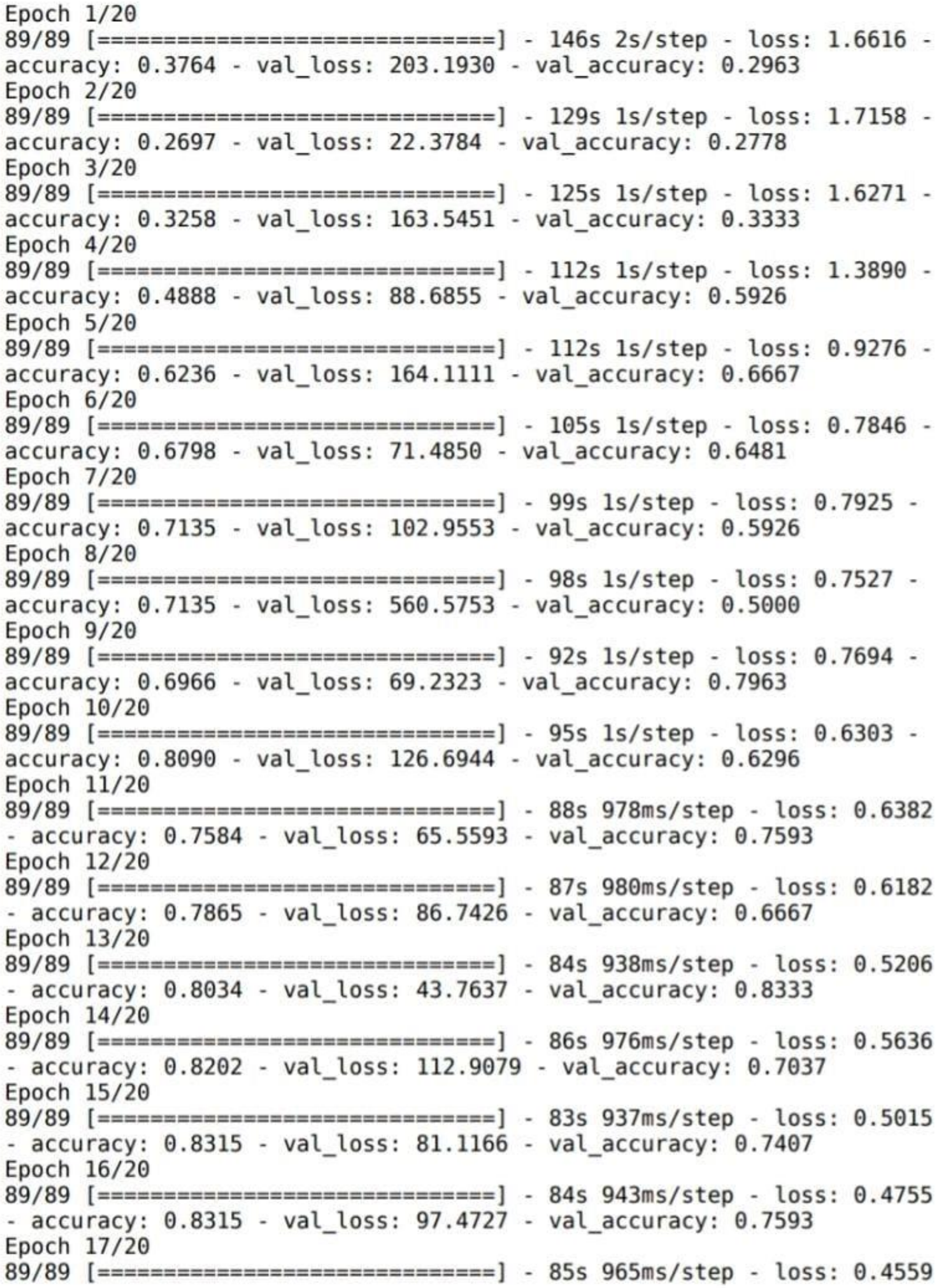
```
model.add(Dense(units=20,kernel_initializer='random_uniform',activatio
n='relu'))
```

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

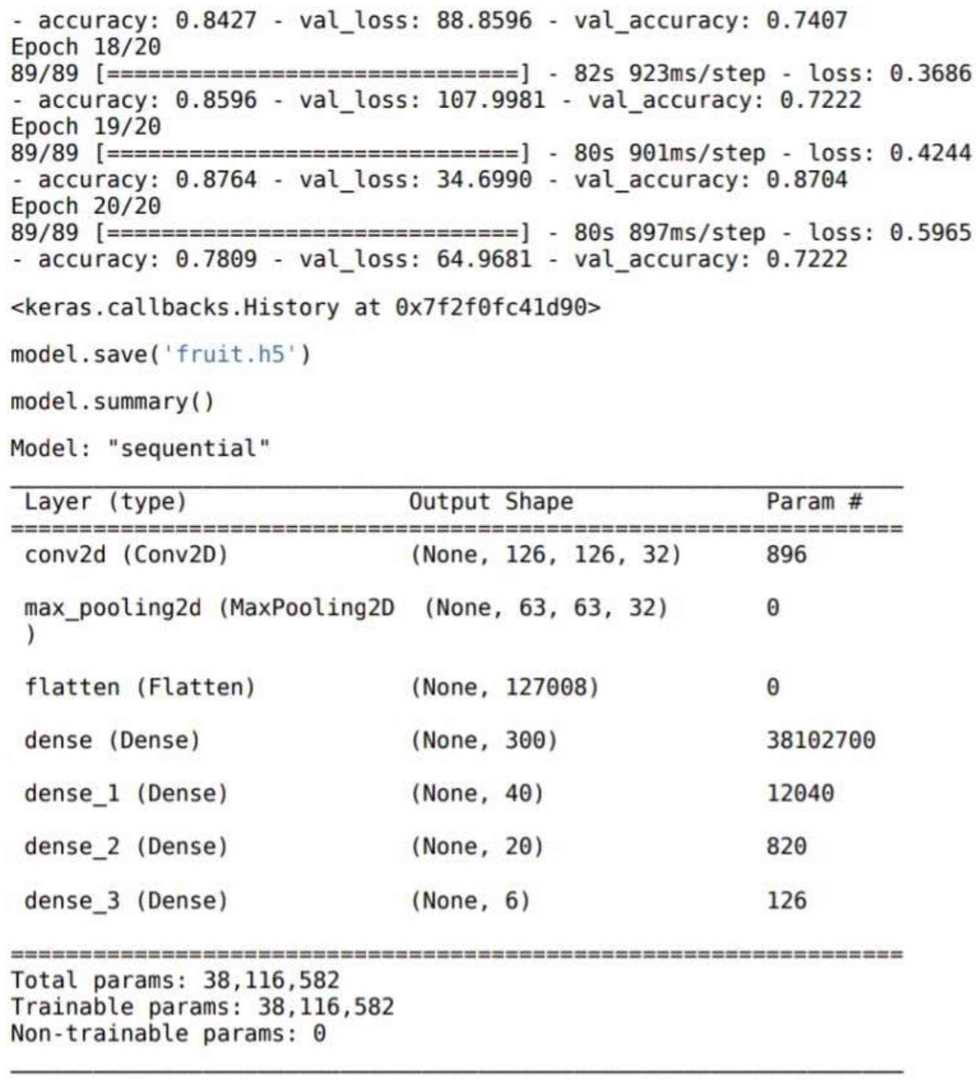
1. Train and save the model

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

```
model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,
validation_steps=27)
```

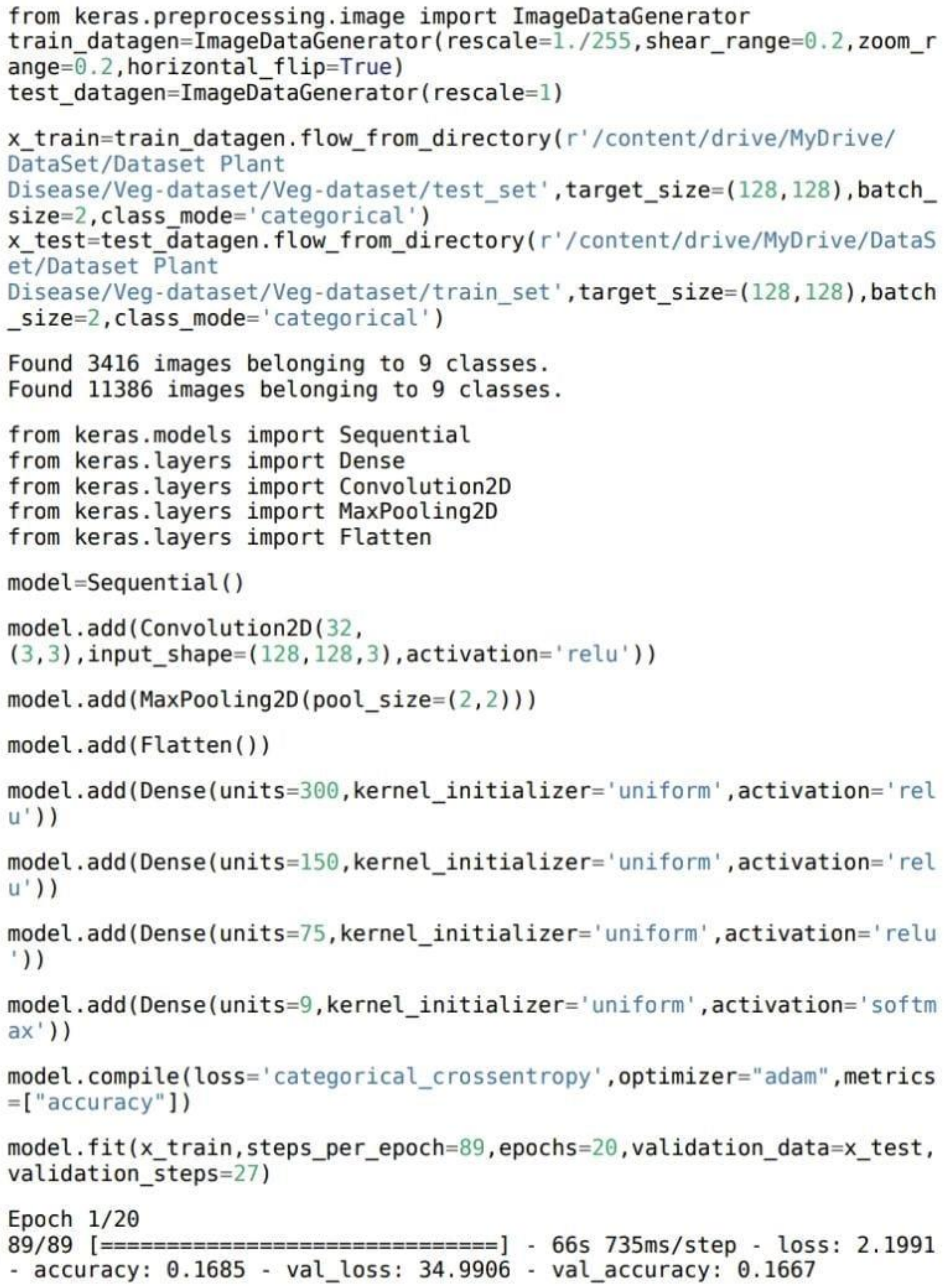


```
Epoch 1/20
89/89 [==============================] - 146s 2s/step - loss: 1.6616 -
accuracy: 0.3764 - val_loss: 203.1930 - val_accuracy: 0.2963
Epoch 2/20
89/89 [==============================] - 129s 1s/step - loss: 1.7158 -
accuracy: 0.2697 - val_loss: 22.3784 - val_accuracy: 0.2778
Epoch 3/20
89/89 [==============================] - 125s 1s/step - loss: 1.6271 -
accuracy: 0.3258 - val_loss: 163.5451 - val_accuracy: 0.3333
Epoch 4/20
89/89 [==============================] - 112s 1s/step - loss: 1.3890 -
accuracy: 0.4888 - val_loss: 88.6855 - val_accuracy: 0.5926
Epoch 5/20
89/89 [==============================] - 112s 1s/step - loss: 0.9276 -
accuracy: 0.6236 - val_loss: 164.1111 - val_accuracy: 0.6667
Epoch 6/20
89/89 [==============================] - 105s 1s/step - loss: 0.7846 -
accuracy: 0.6798 - val_loss: 71.4850 - val_accuracy: 0.6481
Epoch 7/20
89/89 [==============================] - 99s 1s/step - loss: 0.7925 -
accuracy: 0.7135 - val_loss: 102.9553 - val_accuracy: 0.5926
Epoch 8/20
89/89 [==============================] - 98s 1s/step - loss: 0.7527 -
accuracy: 0.7135 - val_loss: 560.5753 - val_accuracy: 0.5000
Epoch 9/20
89/89 [==============================] - 92s 1s/step - loss: 0.7694 -
accuracy: 0.6966 - val_loss: 69.2323 - val_accuracy: 0.7963
Epoch 10/20
89/89 [==============================] - 95s 1s/step - loss: 0.6303 -
accuracy: 0.8090 - val_loss: 126.6944 - val_accuracy: 0.6296
Epoch 11/20
89/89 [==============================] - 88s 978ms/step - loss: 0.6382
- accuracy: 0.7584 - val_loss: 65.5593 - val_accuracy: 0.7593
Epoch 12/20
89/89 [==============================] - 87s 980ms/step - loss: 0.6182
- accuracy: 0.7865 - val_loss: 86.7426 - val_accuracy: 0.6667
Epoch 13/20
89/89 [==============================] - 84s 938ms/step - loss: 0.5206
- accuracy: 0.8034 - val_loss: 43.7637 - val_accuracy: 0.8333
Epoch 14/20
89/89 [==============================] - 86s 976ms/step - loss: 0.5636
- accuracy: 0.8202 - val_loss: 112.9079 - val_accuracy: 0.7037
Epoch 15/20
89/89 [==============================] - 83s 937ms/step - loss: 0.5015
- accuracy: 0.8315 - val_loss: 81.1166 - val_accuracy: 0.7407
Epoch 16/20
89/89 [==============================] - 84s 943ms/step - loss: 0.4755
- accuracy: 0.8315 - val_loss: 97.4727 - val_accuracy: 0.7593
Epoch 17/20
89/89 [==============================] - 85s 965ms/step - loss: 0.4559
```

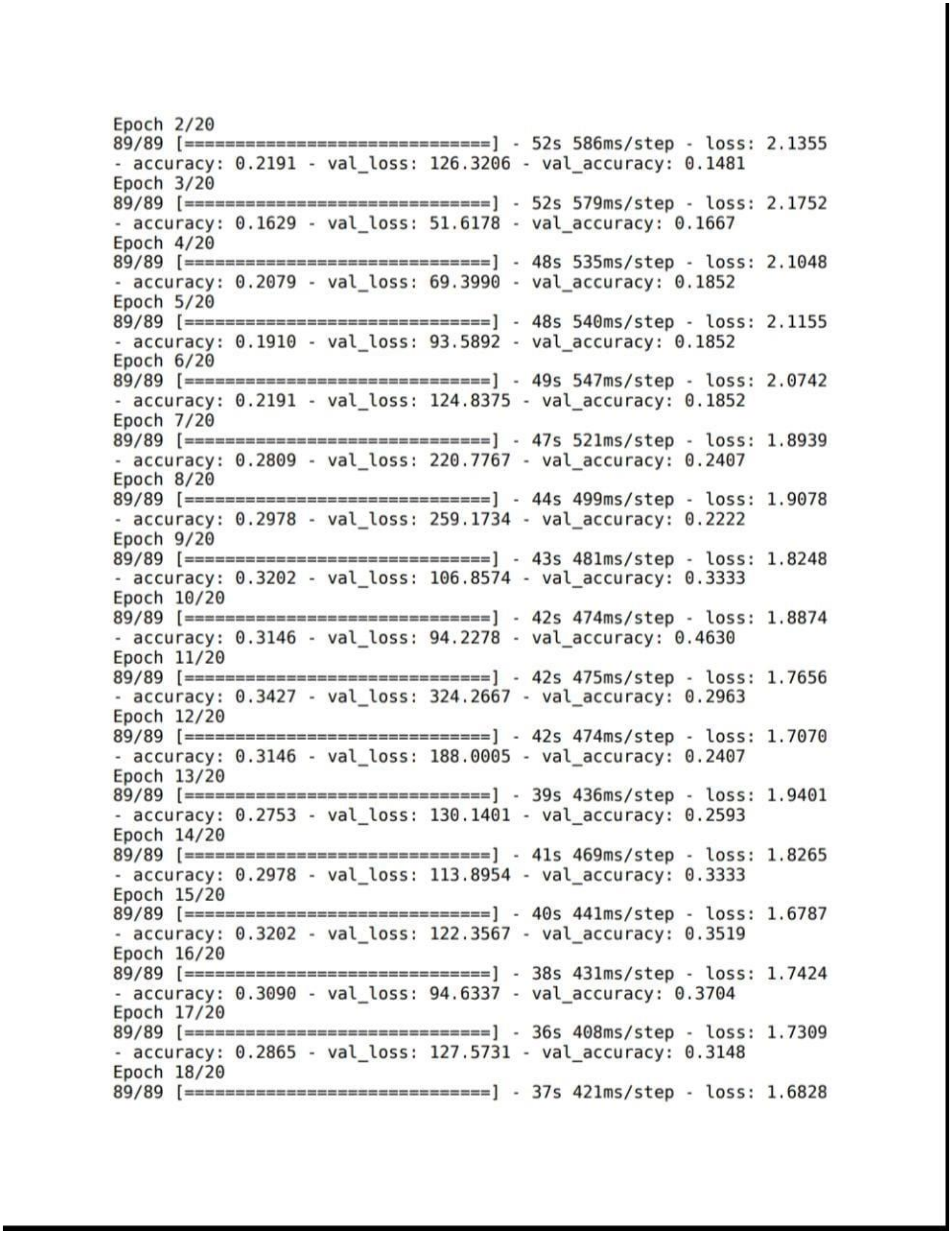


```
- accuracy: 0.8427 - val_loss: 88.8596 - val_accuracy: 0.7407
Epoch 18/20
89/89 [==============================] - 82s 923ms/step - loss: 0.3686
- accuracy: 0.8596 - val_loss: 107.9981 - val_accuracy: 0.7222
Epoch 19/20
89/89 [==============================] - 80s 901ms/step - loss: 0.4244
- accuracy: 0.8764 - val_loss: 34.6990 - val_accuracy: 0.8704
Epoch 20/20
89/89 [==============================] - 80s 897ms/step - loss: 0.5965
- accuracy: 0.7809 - val_loss: 64.9681 - val_accuracy: 0.7222

<keras.callbacks.History at 0x7f2f0fc41d90>

model.save('fruit.h5')

model.summary()

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

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

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 300)               38102700

 dense_1 (Dense)             (None, 40)                12040

 dense_2 (Dense)             (None, 20)                820

 dense_3 (Dense)             (None, 6)                 126

=================================================================
Total params: 38,116,582
Trainable params: 38,116,582
Non-trainable params: 0
_________________________________________________________________
```

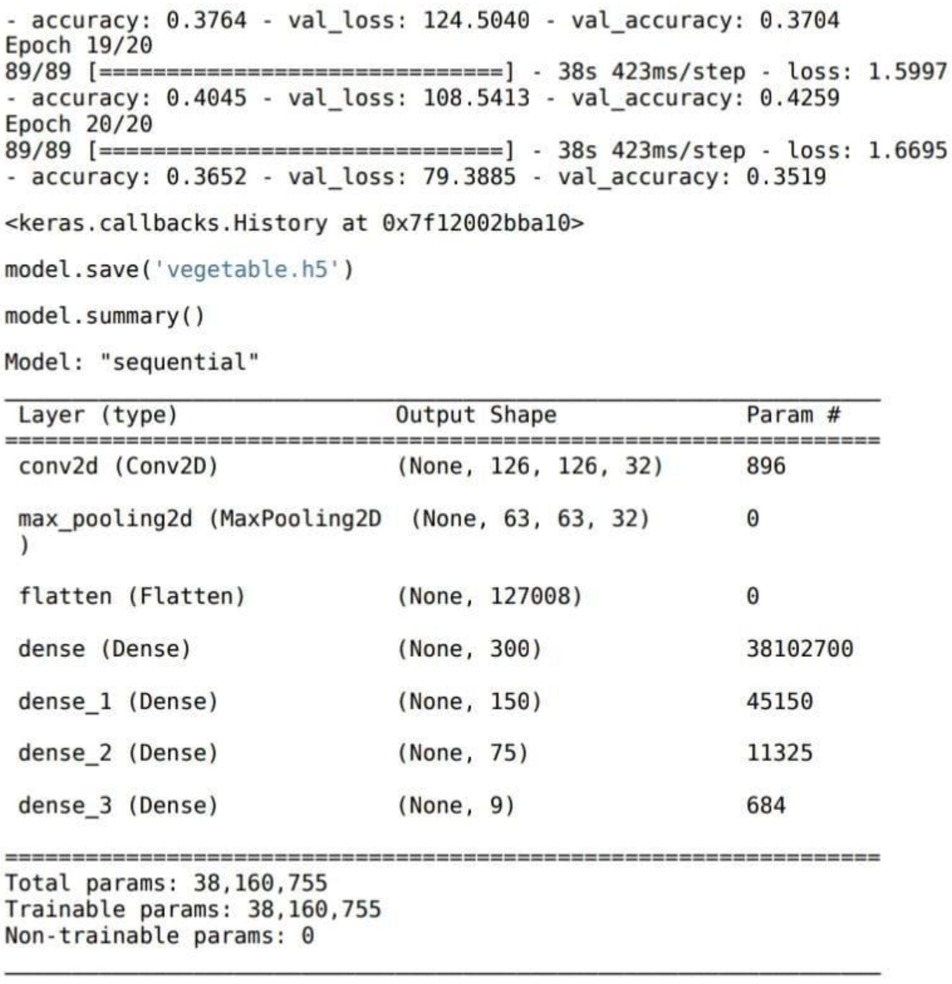
7.2 Feature 2



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

x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/
DataSet/Dataset Plant
Disease/Veg-dataset/Veg-dataset/test_set',target_size=(128,128),batch_
size=2,class_mode='categorical')
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/DataS
et/Dataset Plant
Disease/Veg-dataset/Veg-dataset/train_set',target_size=(128,128),batch
_size=2,class_mode='categorical')

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

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

model=Sequential()

model.add(Convolution2D(32,
(3,3),input_shape=(128,128,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.add(Dense(units=300,kernel_initializer='uniform',activation='rel
u'))

model.add(Dense(units=150,kernel_initializer='uniform',activation='rel
u'))

model.add(Dense(units=75,kernel_initializer='uniform',activation='relu
'))

model.add(Dense(units=9,kernel_initializer='uniform',activation='softm
ax'))

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

model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,
validation_steps=27)

Epoch 1/20
89/89 [==============================] - 66s 735ms/step - loss: 2.1991
- accuracy: 0.1685 - val_loss: 34.9906 - val_accuracy: 0.1667
```



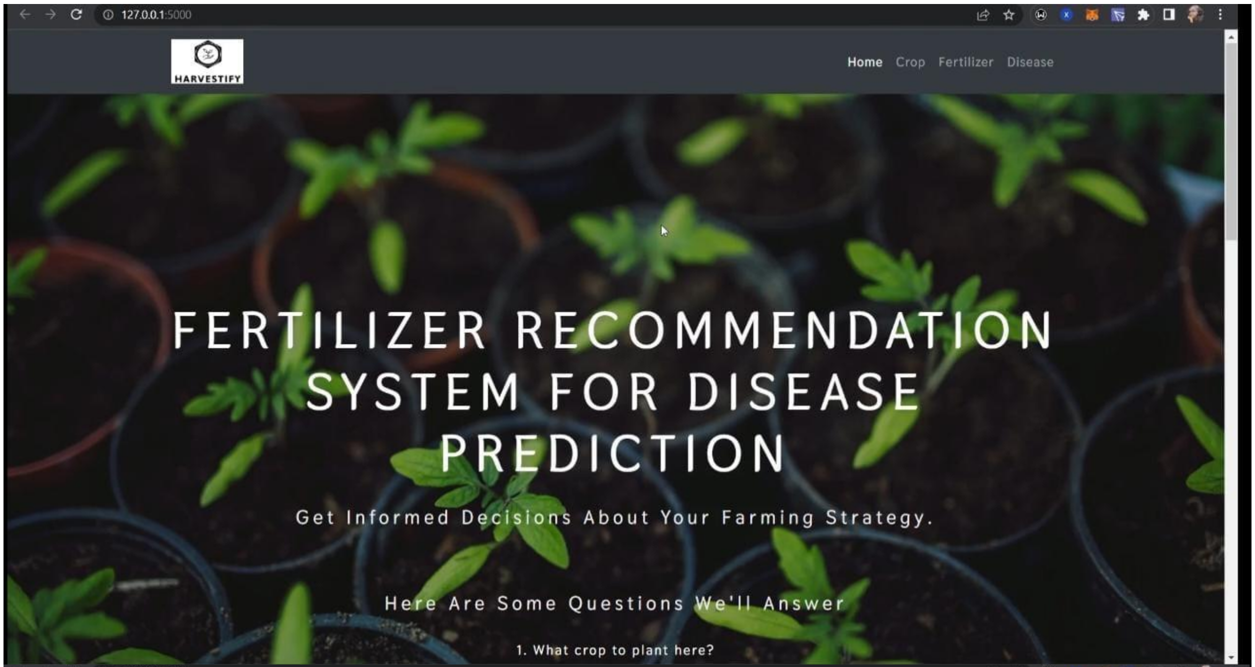
```
Epoch 2/20
89/89 [==============================] - 52s 586ms/step - loss: 2.1355
- accuracy: 0.2191 - val_loss: 126.3206 - val_accuracy: 0.1481
Epoch 3/20
89/89 [==============================] - 52s 579ms/step - loss: 2.1752
- accuracy: 0.1629 - val_loss: 51.6178 - val_accuracy: 0.1667
Epoch 4/20
89/89 [==============================] - 48s 535ms/step - loss: 2.1048
- accuracy: 0.2079 - val_loss: 69.3990 - val_accuracy: 0.1852
Epoch 5/20
89/89 [==============================] - 48s 540ms/step - loss: 2.1155
- accuracy: 0.1910 - val_loss: 93.5892 - val_accuracy: 0.1852
Epoch 6/20
89/89 [==============================] - 49s 547ms/step - loss: 2.0742
- accuracy: 0.2191 - val_loss: 124.8375 - val_accuracy: 0.1852
Epoch 7/20
89/89 [==============================] - 47s 521ms/step - loss: 1.8939
- accuracy: 0.2809 - val_loss: 220.7767 - val_accuracy: 0.2407
Epoch 8/20
89/89 [==============================] - 44s 499ms/step - loss: 1.9078
- accuracy: 0.2978 - val_loss: 259.1734 - val_accuracy: 0.2222
Epoch 9/20
89/89 [==============================] - 43s 481ms/step - loss: 1.8248
- accuracy: 0.3202 - val_loss: 106.8574 - val_accuracy: 0.3333
Epoch 10/20
89/89 [==============================] - 42s 474ms/step - loss: 1.8874
- accuracy: 0.3146 - val_loss: 94.2278 - val_accuracy: 0.4630
Epoch 11/20
89/89 [==============================] - 42s 475ms/step - loss: 1.7656
- accuracy: 0.3427 - val_loss: 324.2667 - val_accuracy: 0.2963
Epoch 12/20
89/89 [==============================] - 42s 474ms/step - loss: 1.7070
- accuracy: 0.3146 - val_loss: 188.0005 - val_accuracy: 0.2407
Epoch 13/20
89/89 [==============================] - 39s 436ms/step - loss: 1.9401
- accuracy: 0.2753 - val_loss: 130.1401 - val_accuracy: 0.2593
Epoch 14/20
89/89 [==============================] - 41s 469ms/step - loss: 1.8265
- accuracy: 0.2978 - val_loss: 113.8954 - val_accuracy: 0.3333
Epoch 15/20
89/89 [==============================] - 40s 441ms/step - loss: 1.6787
- accuracy: 0.3202 - val_loss: 122.3567 - val_accuracy: 0.3519
Epoch 16/20
89/89 [==============================] - 38s 431ms/step - loss: 1.7424
- accuracy: 0.3090 - val_loss: 94.6337 - val_accuracy: 0.3704
Epoch 17/20
89/89 [==============================] - 36s 408ms/step - loss: 1.7309
- accuracy: 0.2865 - val_loss: 127.5731 - val_accuracy: 0.3148
Epoch 18/20
89/89 [==============================] - 37s 421ms/step - loss: 1.6828
```



```
- accuracy: 0.3764 - val_loss: 124.5040 - val_accuracy: 0.3704
Epoch 19/20
89/89 [==============================] - 38s 423ms/step - loss: 1.5997
- accuracy: 0.4045 - val_loss: 108.5413 - val_accuracy: 0.4259
Epoch 20/20
89/89 [==============================] - 38s 423ms/step - loss: 1.6695
- accuracy: 0.3652 - val_loss: 79.3885 - val_accuracy: 0.3519

<keras.callbacks.History at 0x7f12002bba10>

model.save('vegetable.h5')

model.summary()

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

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

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 300)               38102700

 dense_1 (Dense)             (None, 150)               45150

 dense_2 (Dense)             (None, 75)                11325

 dense_3 (Dense)             (None, 9)                 684

=================================================================
Total params: 38,160,755
Trainable params: 38,160,755
Non-trainable params: 0
_________________________________________________________________
```

8. TESTING

8.1 Test Cases

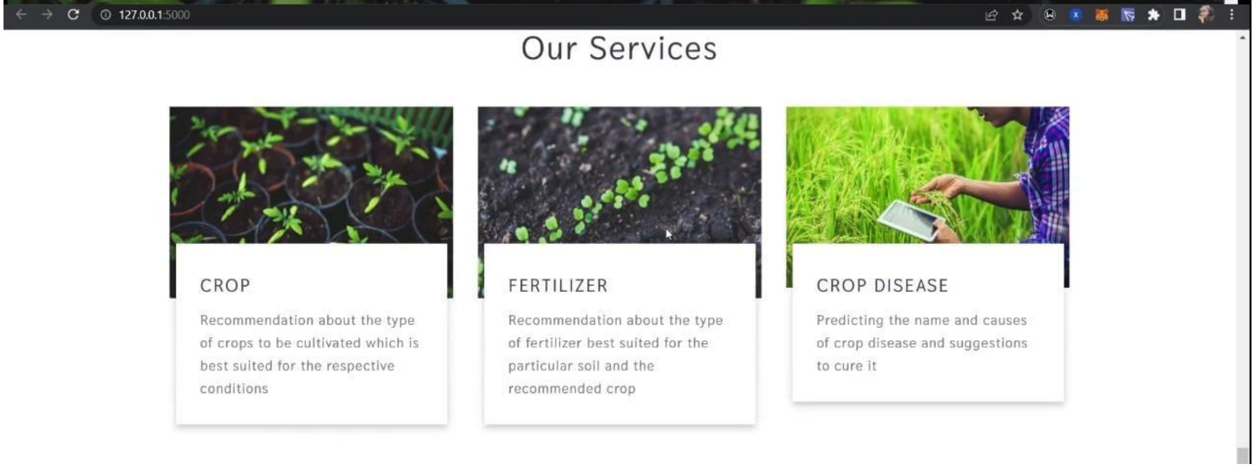


FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

Get Informed Decisions About Your Farming Strategy.

Here Are Some Questions We'll Answer

1. What crop to plant here?



Our Services

CROP

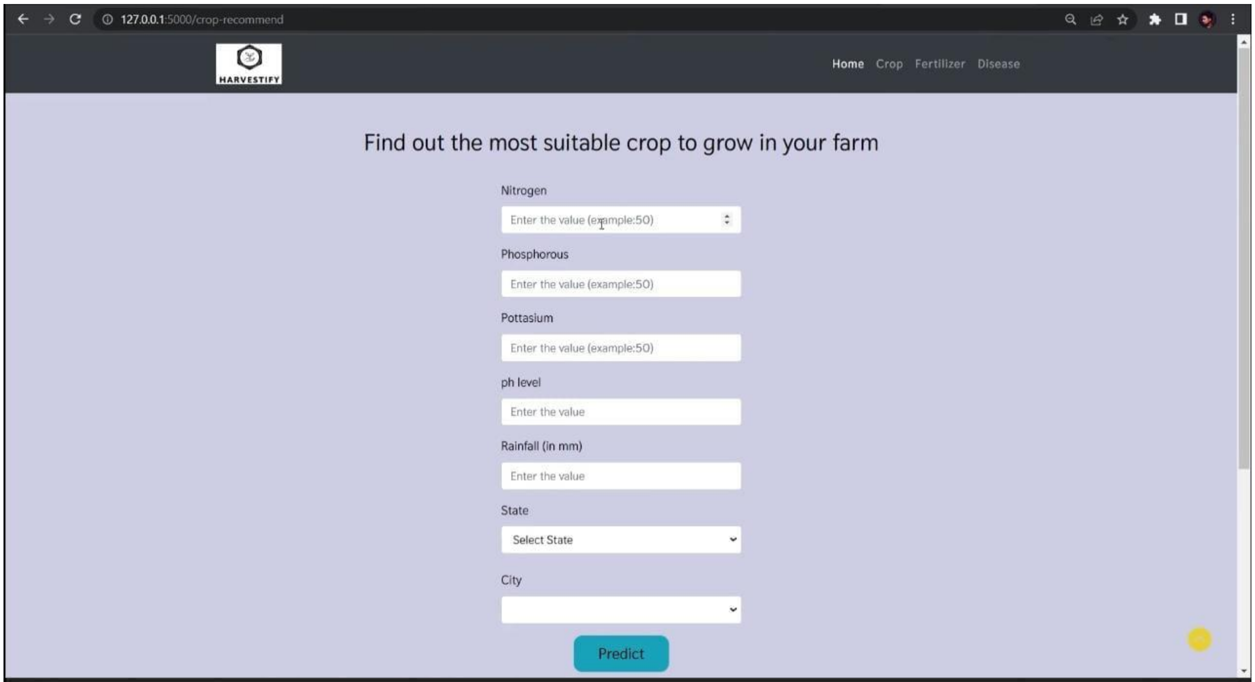
Recommendation about the type of crops to be cultivated which is best suited for the respective conditions

FERTILIZER

Recommendation about the type of fertilizer best suited for the particular soil and the recommended crop

CROP DISEASE

Predicting the name and causes of crop disease and suggestions to cure it



Find out the most suitable crop to grow in your farm

Nitrogen

Enter the value (example:50)

Phosphorous

Enter the value (example:50)

Pottasium

Enter the value (example:50)

ph level

Enter the value

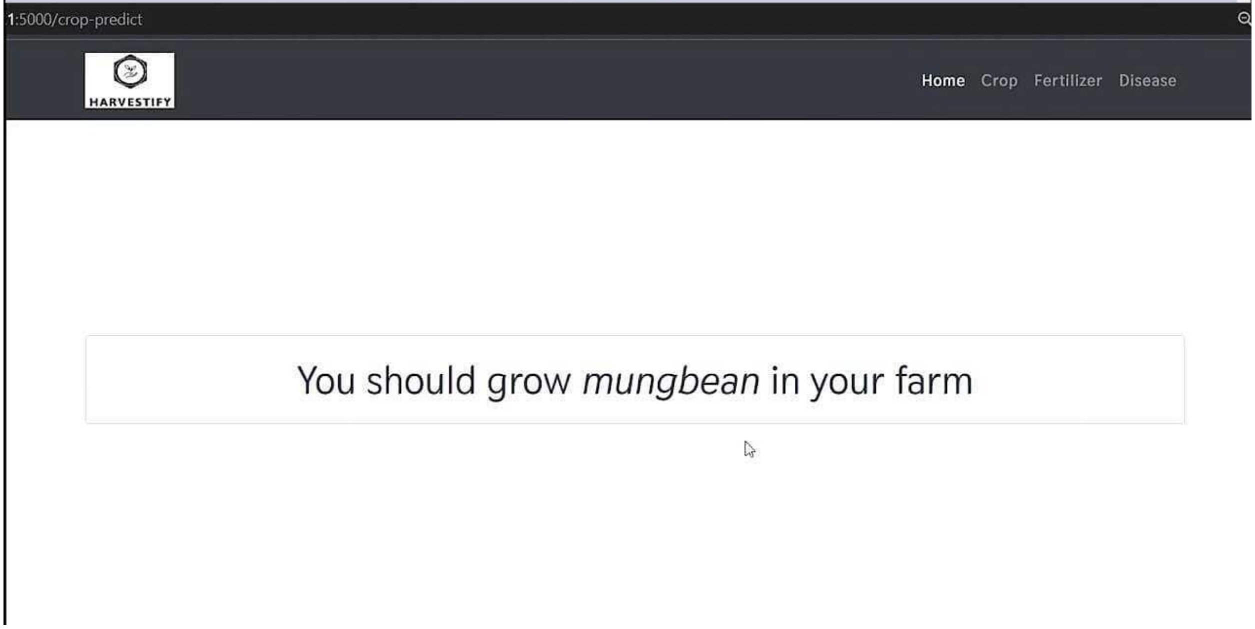
Rainfall (in mm)

Enter the value

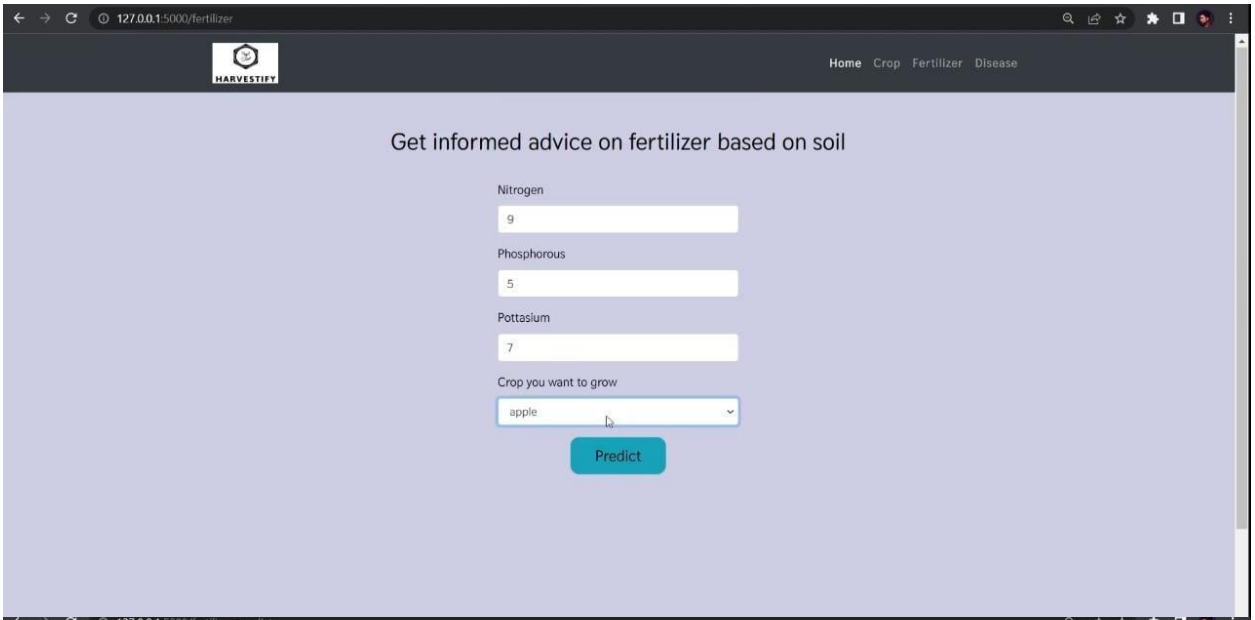
State

Select State

City



You should grow *mungbean* in your farm



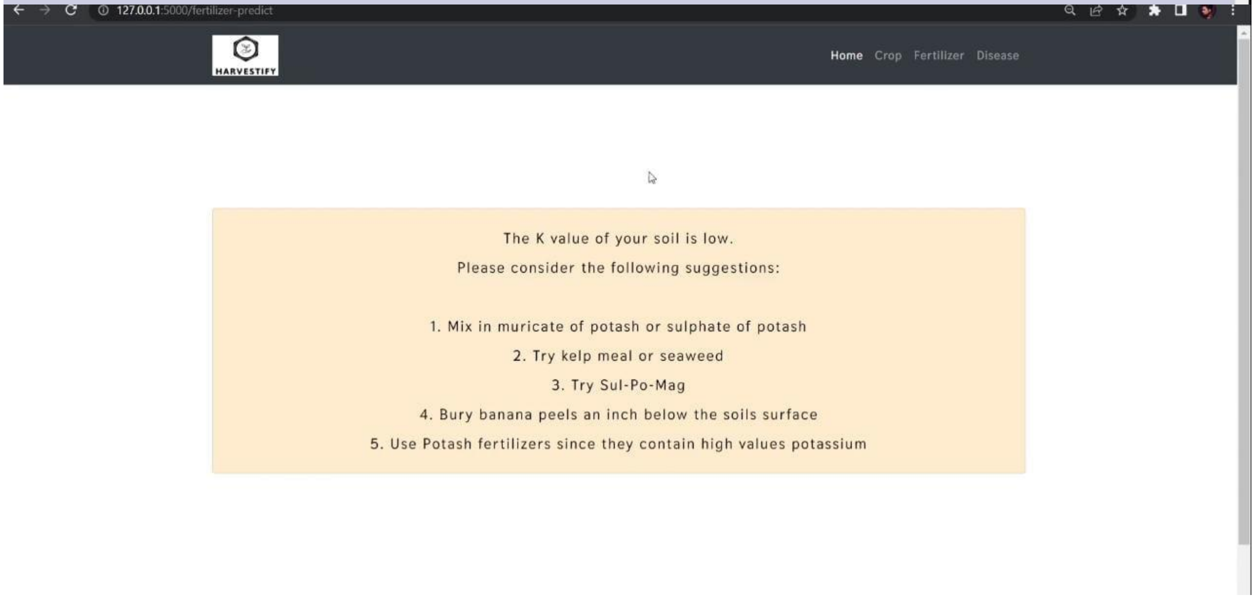
Get informed advice on fertilizer based on soil

Nitrogen: 9

Phosphorous: 5

Pottasium: 7

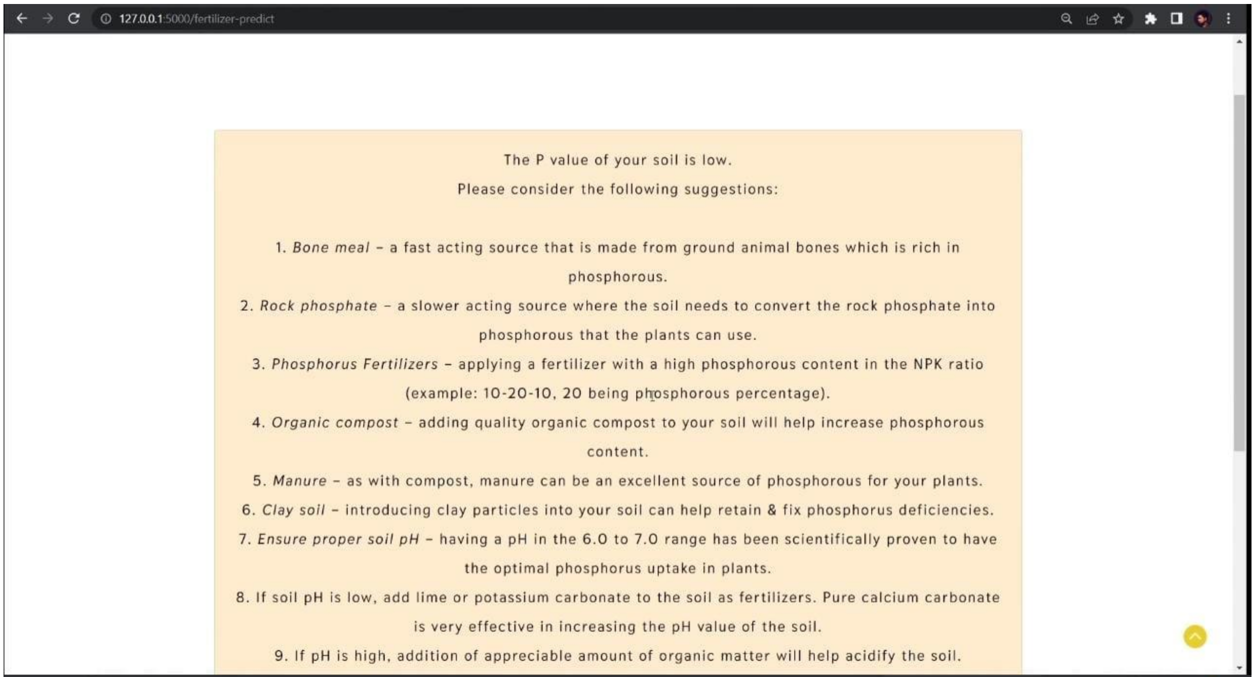
Crop you want to grow: apple



The K value of your soil is low.

Please consider the following suggestions:

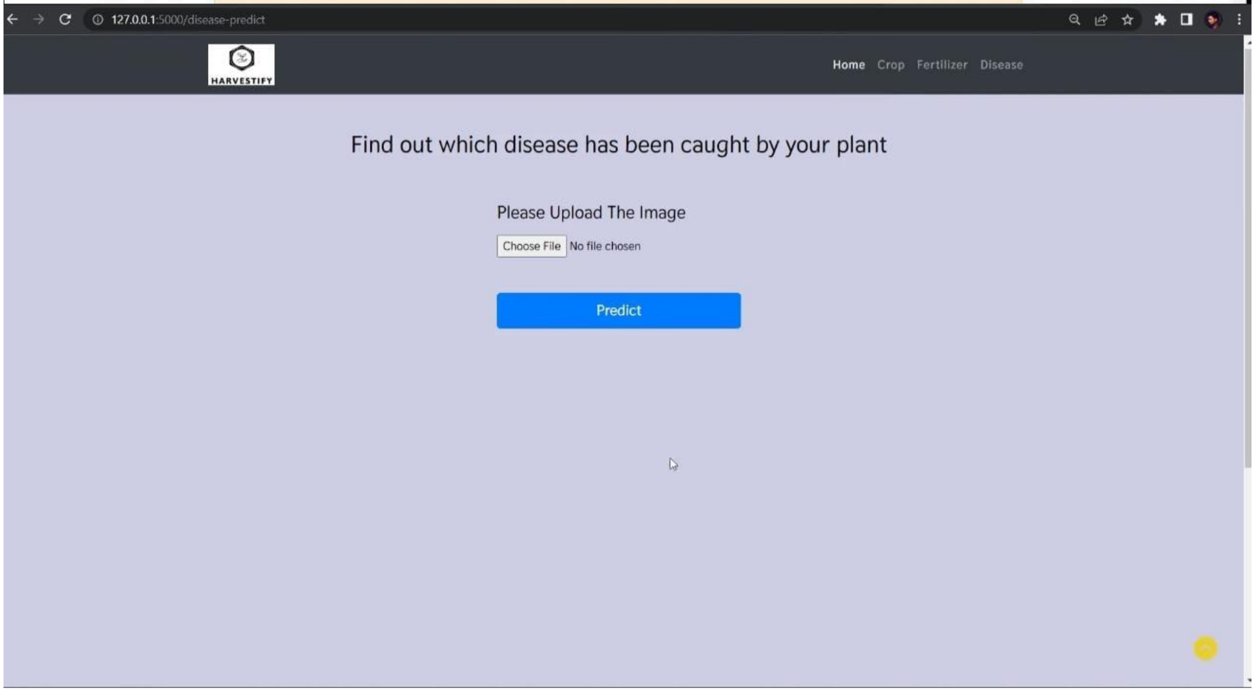
1. Mix in muricate of potash or sulphate of potash
2. Try kelp meal or seaweed
3. Try Sul-Po-Mag
4. Bury banana peels an inch below the soils surface
5. Use Potash fertilizers since they contain high values potassium



The P value of your soil is low.

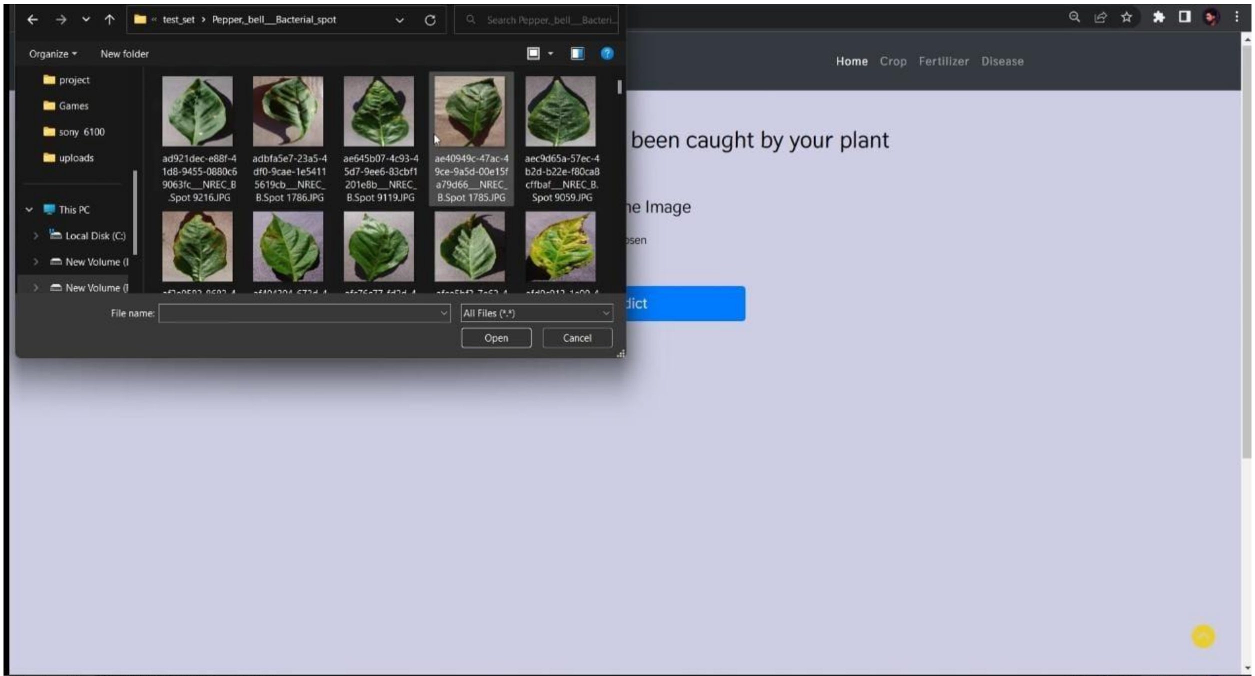
Please consider the following suggestions:

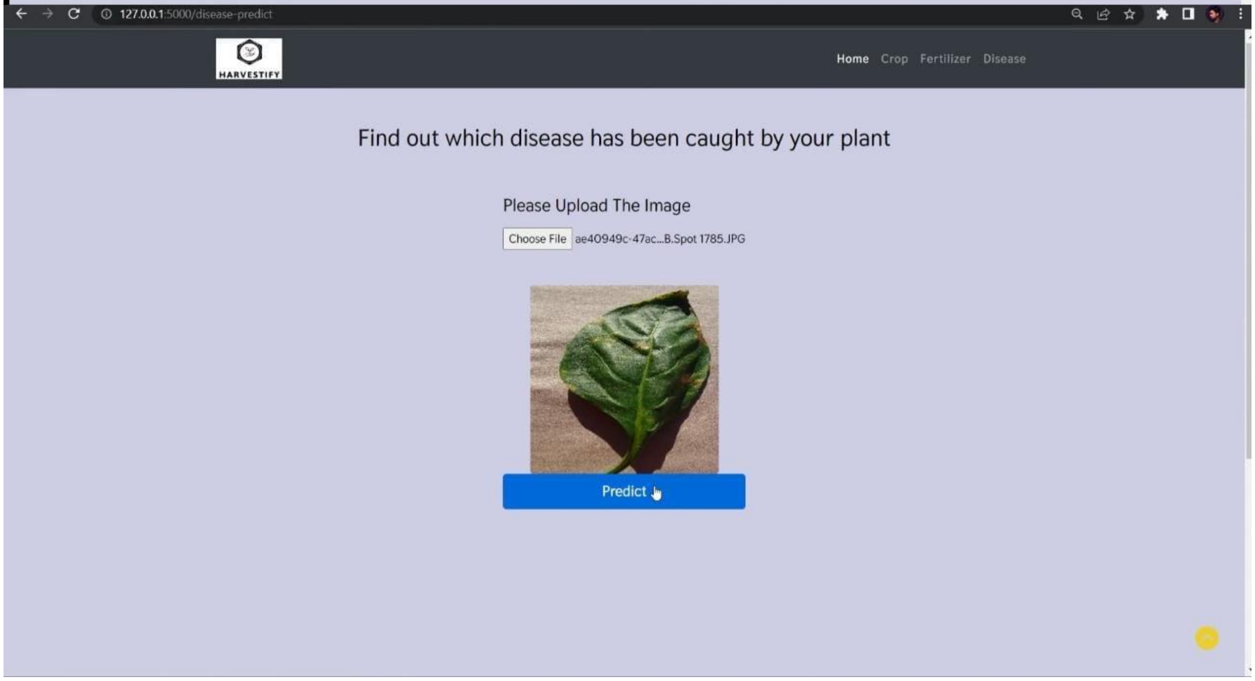
1. *Bone meal* – a fast acting source that is made from ground animal bones which is rich in phosphorous.
2. *Rock phosphate* – a slower acting source where the soil needs to convert the rock phosphate into phosphorous that the plants can use.
3. *Phosphorus Fertilizers* – applying a fertilizer with a high phosphorous content in the NPK ratio (example: 10-20-10, 20 being phosphorous percentage).
4. *Organic compost* – adding quality organic compost to your soil will help increase phosphorous content.
5. *Manure* – as with compost, manure can be an excellent source of phosphorous for your plants.
6. *Clay soil* – introducing clay particles into your soil can help retain & fix phosphorus deficiencies.
7. *Ensure proper soil pH* – having a pH in the 6.0 to 7.0 range has been scientifically proven to have the optimal phosphorus uptake in plants.
8. If soil pH is low, add lime or potassium carbonate to the soil as fertilizers. Pure calcium carbonate is very effective in increasing the pH value of the soil.
9. If pH is high, addition of appreciable amount of organic matter will help acidify the soil.



Find out which disease has been caught by your plant

Please Upload The Image


test_set > Pepper,_bell___Bacterial_spot
Search Pepper,_bell___Bacteri...
Organize
New folder
project
Games
sony 6100
uploads
This PC
Local Disk (C:)
New Volume (I
New Volume (I
ad921dec-e88f-41d8-9455-0880c69063fc___NREC_B.Spot 9216.JPG
adbfa5e7-23a5-4df0-9cae-1e54115619cb___NREC_B.Spot 1786.JPG
ae645b07-4c93-45d7-9ee6-83cbf1201e8b___NREC_B.Spot 9119.JPG
ae40949c-47ac-49ce-9a5d-00e15fa79d66___NREC_B.Spot 1785.JPG
aec9d65a-57ec-4b2d-b22e-f80ca8cffbaf___NREC_B.Spot 9059.JPG
File name:
All Files (*.*)
Open
Cancel
Home
Crop
Fertilizer
Disease
been caught by your plant
he Image
osen
dict

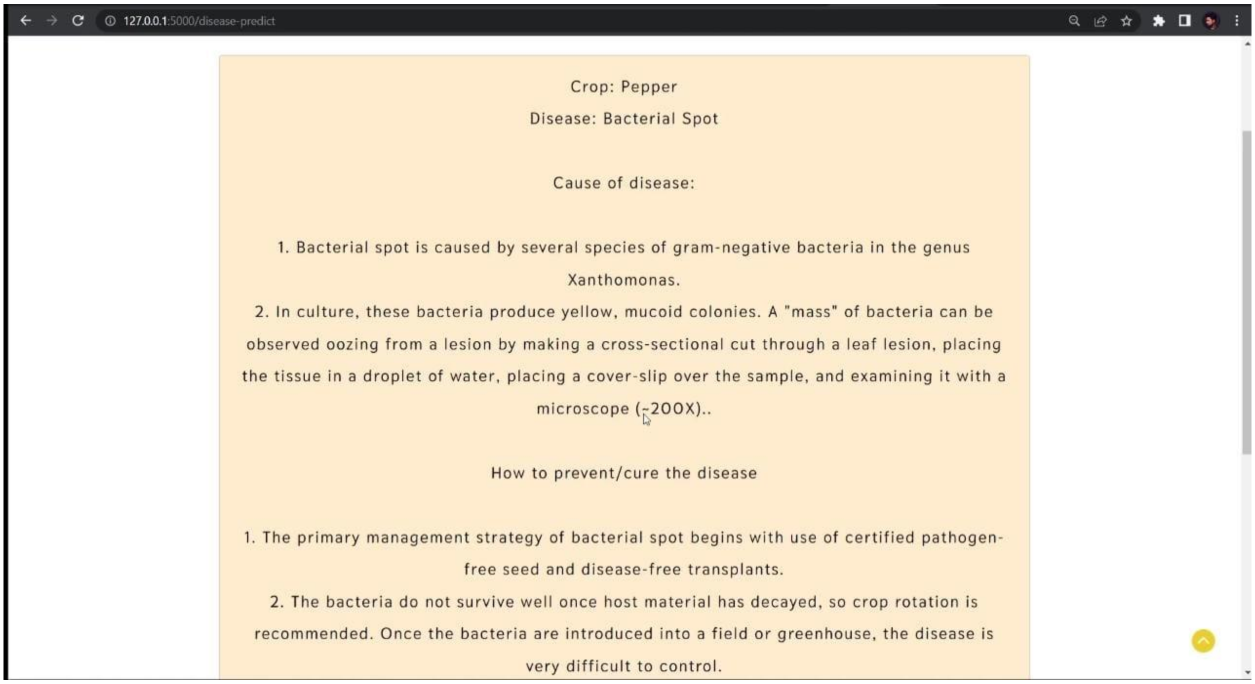


Find out which disease has been caught by your plant

Please Upload The Image

Choose File ae40949c-47ac...B.Spot 1785.JPG

Predict



Crop: Pepper

Disease: Bacterial Spot

Cause of disease:

1. Bacterial spot is caused by several species of gram-negative bacteria in the genus Xanthomonas.
2. In culture, these bacteria produce yellow, mucoid colonies. A "mass" of bacteria can be observed oozing from a lesion by making a cross-sectional cut through a leaf lesion, placing the tissue in a droplet of water, placing a cover-slip over the sample, and examining it with a microscope (~200X)..

How to prevent/cure the disease

1. The primary management strategy of bacterial spot begins with use of certified pathogen-free seed and disease-free transplants.
2. The bacteria do not survive well once host material has decayed, so crop rotation is recommended. Once the bacteria are introduced into a field or greenhouse, the disease is very difficult to control.

8.2 User Acce ptance Testing

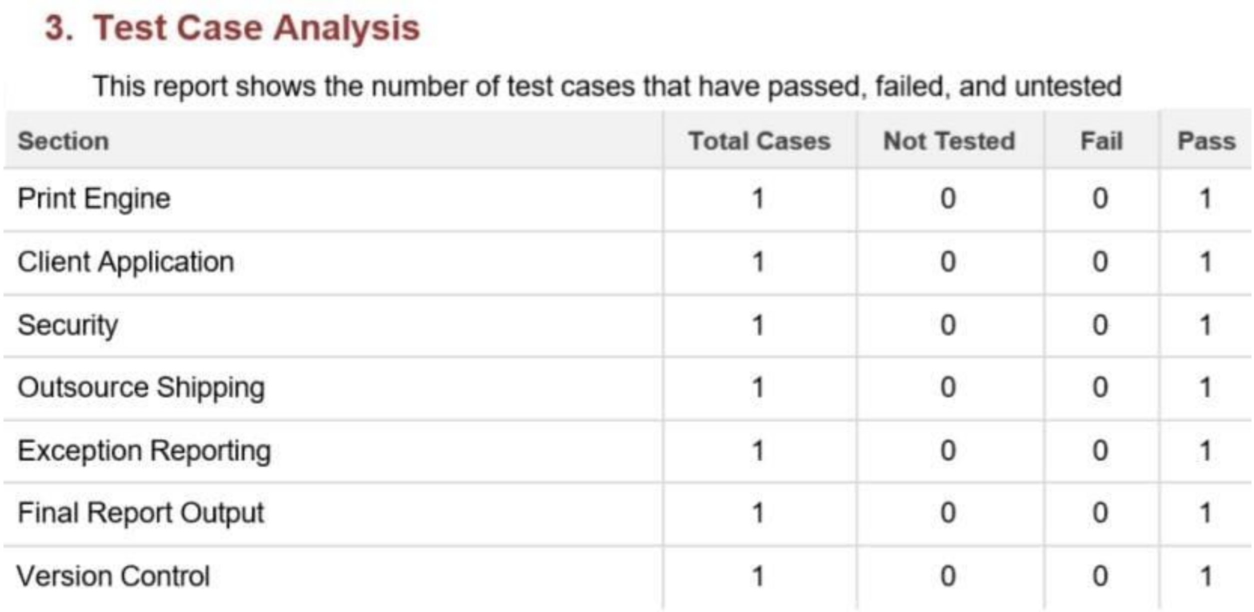
1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Fertilizers Recommendation System for Disease Prediction project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

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



3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	1	0	0	1
Client Application	1	0	0	1
Security	1	0	0	1
Outsource Shipping	1	0	0	1
Exception Reporting	1	0	0	1
Final Report Output	1	0	0	1
Version Control	1	0	0	1

9. RESULTS

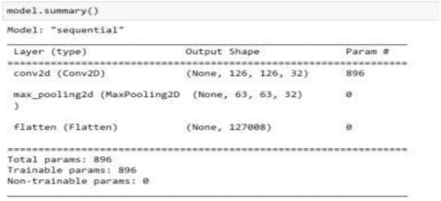
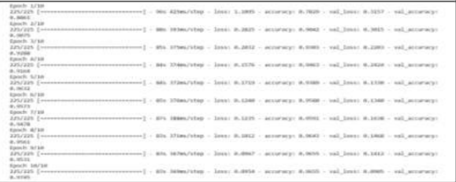
9.1 Performance Metrics

Project Development Phase
Model Performance Test

Date	19 November 2022
Team ID	PNT2022TMID48803
Project Name	Project - Fertilizers Recommendation System for Disease Prediction
Maximum Marks	10 Marks

Model Performance Testing:

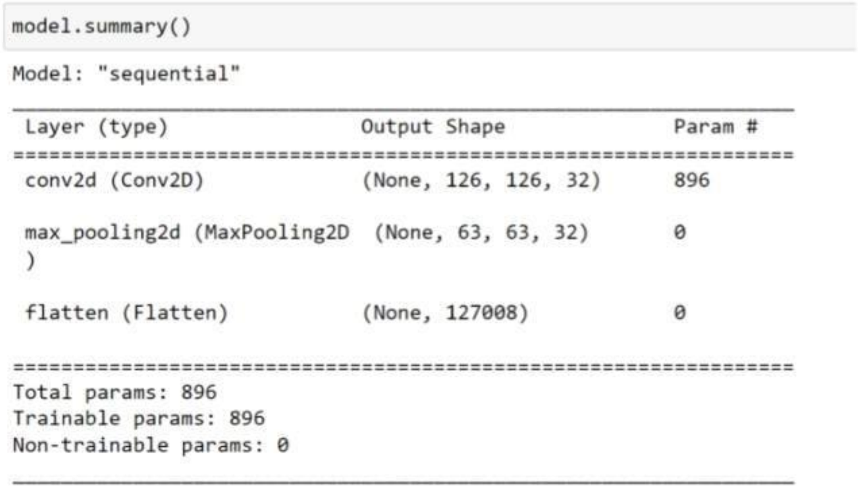
Project team shall fill the following information in model performance testing template.

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: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 2/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 3/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 4/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 5/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 6/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 7/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 8/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 9/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
Epoch 10/10
225/225 [==============================] - [illegible] - loss: [illegible] - accuracy: [illegible] - val_loss: [illegible] - val_accuracy:
[illegible]
```

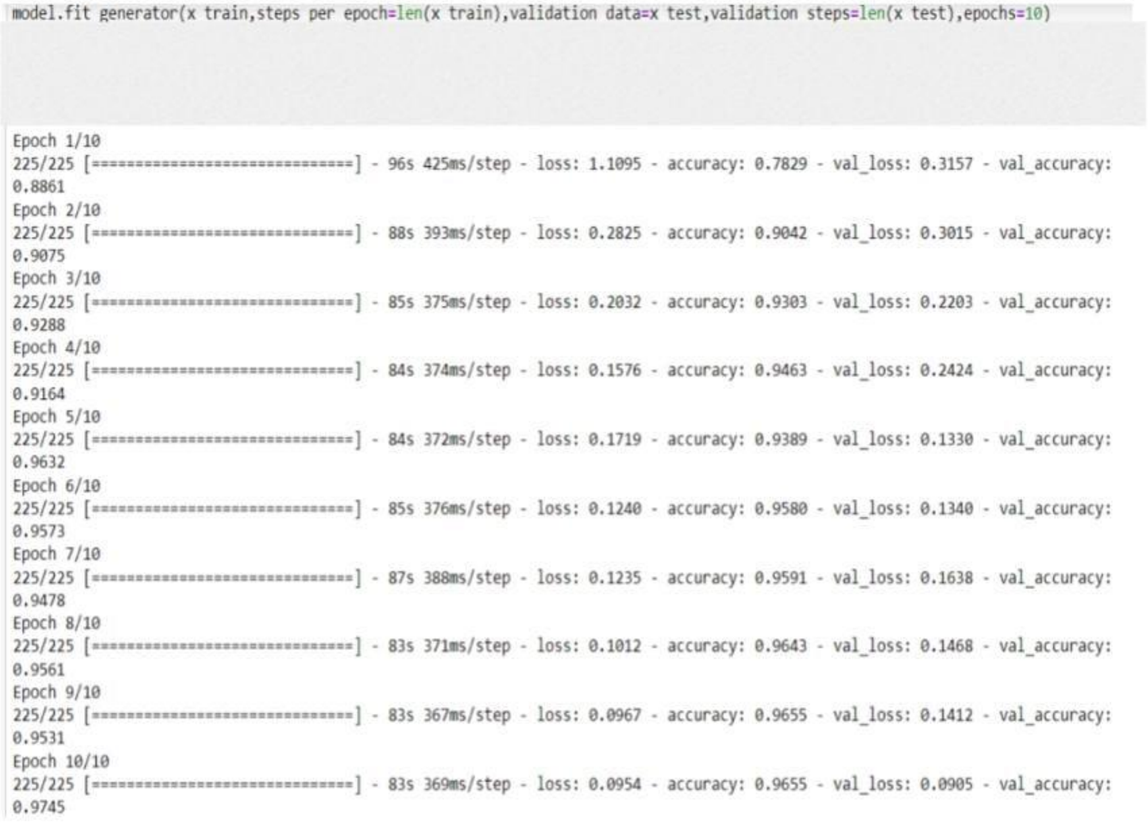
Model Summary



```
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
Non-trainable params: 0
_________________________________________________________________
```

Accuracy



```
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 & DISADVANTAGES

List of advantages

- The proposed model here produces very high accuracy of classification.
- Very large dataset can 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 computational time.
- The neural network architecture used in this project work has high complexity

11. CONCLUSION

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

● The accuracy of classification increased by increasing the number of epochs.

●For different batch sizes, different classification accuracies are obtained.

●The accuracies are increased by increasing more convolution layers.

●The accuracy of classification also increased by varying dense layers.

●Different accuracies are obtained by varying the size of kernel used in the convolution layer output.

● Accuracies are different while varying the size of the train and test datasets.

12. FUTURE SCOPE

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

13. APPENDIX

GitHub & Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-41529-1660642672
https://drive.google.com/file/d/1JbBcWvvKZnKy46WqVYOlGu_3NWtuRroU
/view?usp=share_link