# PROJECT REPORT

INTRODUCTION

* 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 Network(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. 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

1EXISTING PROBLEM

Indumathi proposed a method for leaf disease detection and suggest fertilizers to cure leaf diseases[1]. But the method involves less number of train and test sets which results in poor accuracy. Pandi selvi [2] proposed a simple prediction method for soil based fertilizer recommedation system forpredicted crop diseases. This method gives less accuracy and prediction. Shiva reddy [3] proposed an IoT based system for leaf disease detection and fertilizer recommendation which is based on Machine Learning techniques yields less 80 percentage accuracies

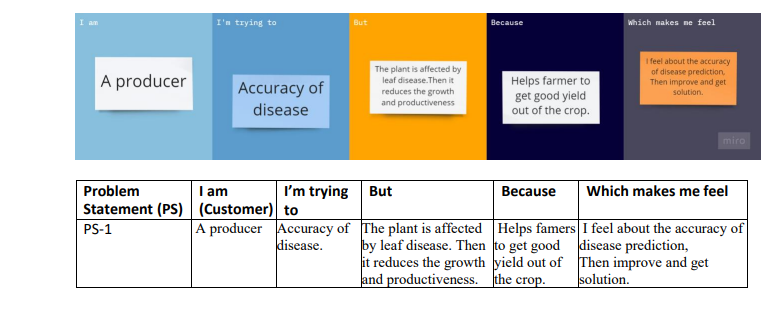
1. LITERATURE SURVEY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **AUTHOR&YEA R** | **TITLE** | **DESCRPTION** | **ACCUR ACY** |
| 01. | AUTHORS: D.N.V.S.L.S indra, M.Sobhana, A.H.L.Swaroop, V.PhaniKumar 2022 | A Machine Learning Based New Recommendaton SSssem To The Farmer | Totally 54% of India's land area is deemed arable, making it the world's largest agrarian economy | 93% |
| 02. | AUTHORS: Sambhav Bhansali, Punis Shah, JinaS Shah, PriSal VSas,Poonam Thakre 2022 | Crop Predicton And Disease Desecton SSssem | Economy of India highly depends on agriculture. Stll traditonal ways of recommendatons are used for agriculture | 83% |
| 03. | AUTHORS: SanidhSaPurohis, DeepSanghani, NamanSenjaliSa, Prof.Anuradha Kapoor 2022 | Agro-Farm-Crop,Fertlizer & Disease Predicton | Data mining is a rising studies area in crop yield analyis Yield prediciton is a complete essental problem in agriculture | 89% |
| 04. | AUTHORS: N.Valarmashi, M.Vengaseshwaran, Kalaimani Shanmugam, R.Sudha 2021 | Neural Neswork Based Fertlizers Recommendaton SSssem For Disorder Classifcaton And Predicton In Pesal Images | Determinaton of plant ailment is basic for early fnding and control of it. The unaided eye method is generally utliied for the conclusion of ailments. | 90% |
| 05. | AUTHORS: Dr.S.UshaKirushikika, Dr.S.Kanaga Suba Raja,S.R. Ronak,S.Rengarajen, P.Ravindran 2020 | Design and Implemensaton of Fertlizer Recommendaton SSssem for Farmer | India is an agrarian naton. But creatng a proftable yield for the farmer in each crop cycle is becoming a major challenge on various factors | 85% |

* 1. REFERENCE
* Reyes Angie . K
* Humrouni . L
* Dimitrovski
* H.S.Narendraswamy
* Sue Han
* Vijay Laxmi
  1. PROBLEM STATEMENT DEFINITION.

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you’ll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.



1. IDEATION & PROPOSED SOLUTION
   1. EMPATHY MAP CANVAS

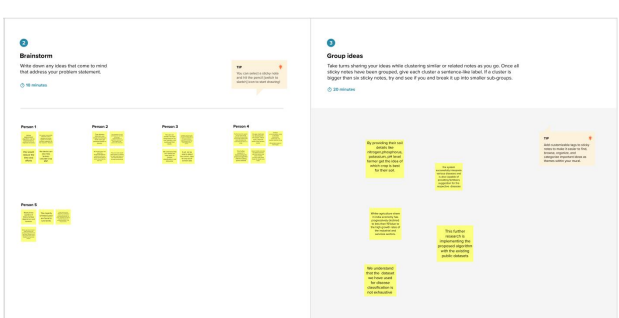
An empathy map is a simple, easy-to-digest visual that captures knowledge about auser’sbehaviours and attitudes. It is a useful tool to helps teams better understand their users.

It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problemand the personwhois experiencing it. The exercise of creating the map helps participants consider things fromthe user’s perspective along with his or her goals and challenges.



* 1. IDEATION & BRAINSTORMING

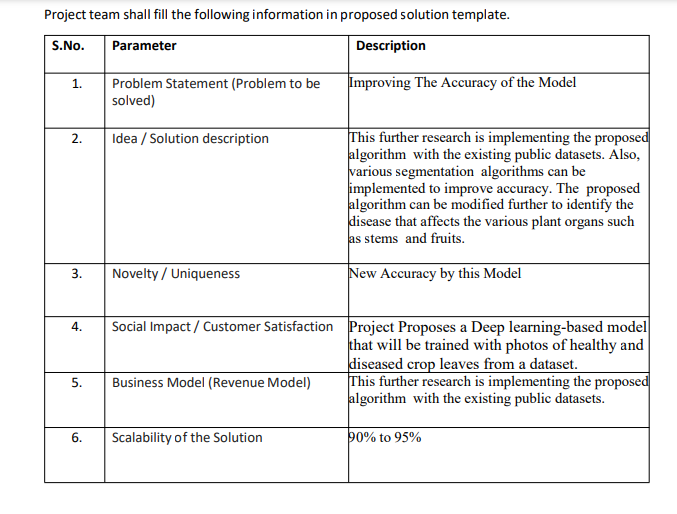
Brainstorming is **a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption**. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind.





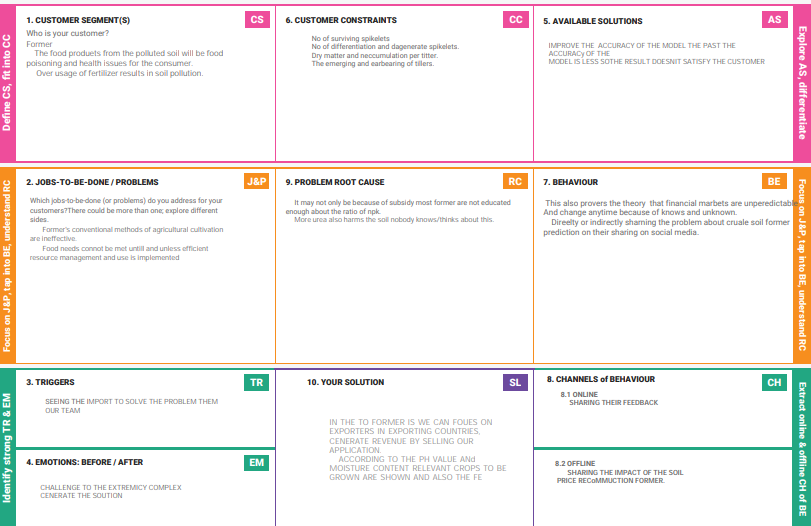
* 1. PROPOSED SOLUTION

In this project work, a deep learning based neural network is used to train the collected datasets and test the same. The deep learning based neural network is CNN which gives more than 90% classiﬁcation accuracies. By increasing the more number of dense layers and by modifying hyperparameters such as number of epochs, batch size, the accuracy rate can be increased to 95% to 98%.



* 1. PROBLEM SOLUTION FIT

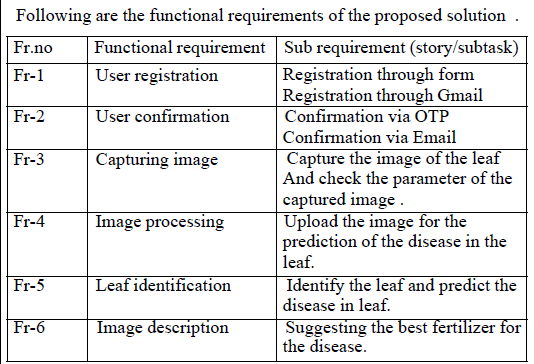
The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem.



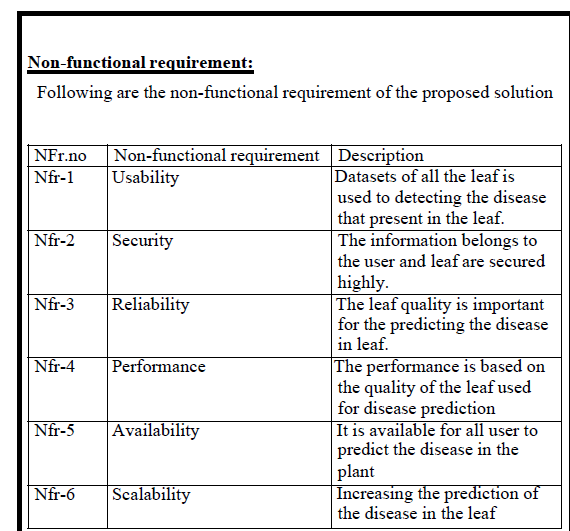
1. REQUIREMENT ANALYSIS

**A functional requirement defines a system or its component. A non-functional requirement defines the quality attribute of a software system**. It specifies “What should the software system do?” It places constraints on “How should the software system fulfill the functional requirements?”

* 1. FUNCTIONAL REQUIREMENT



NON-FUNCTIONAL REQUIREMENT:



1. PROJECT DESIGN

.

* 1. DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right

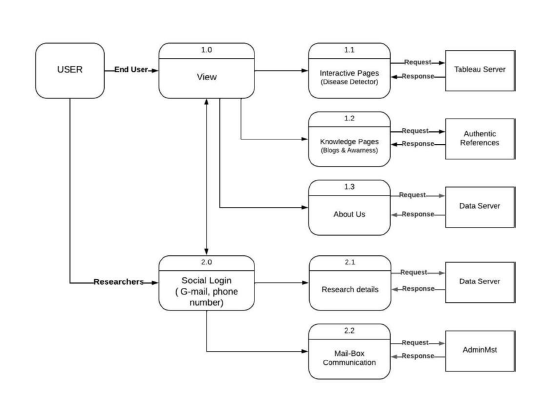
amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored

# v

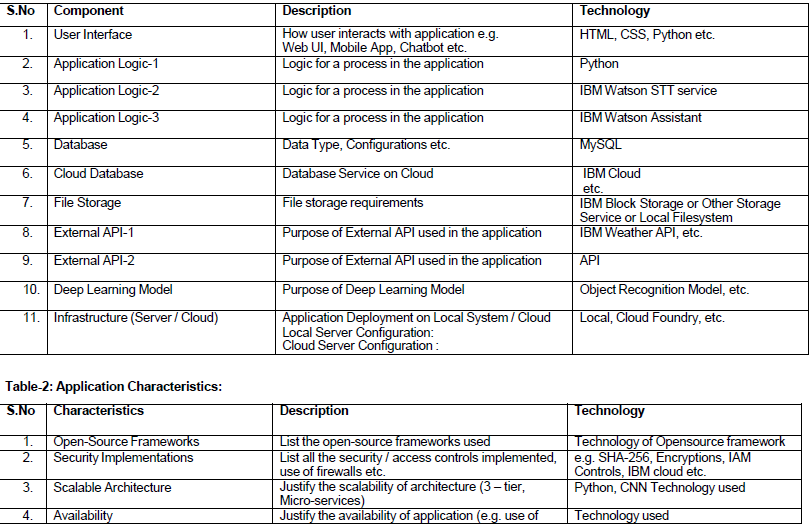
In this flow diagram we are showing that the fertilizer recommendation system

* 1. SOLUTION AND TECHNICAL ARCHITECTURE

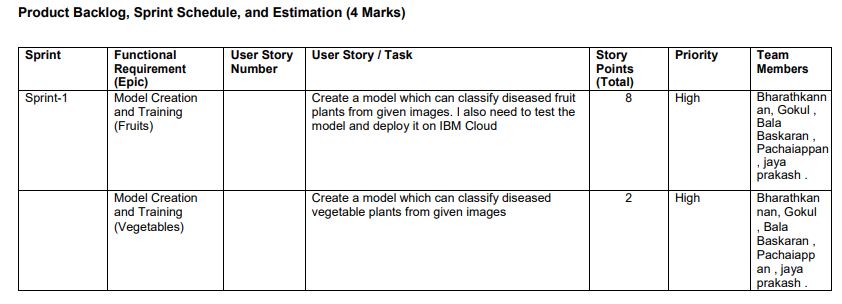
A solution architecture (SA) is **an architectural description of a specific solution**. SAs combine guidance from different enterprise architecture viewpoints (business, information and technical), as well as from the enterprise solution architecture (ESA).

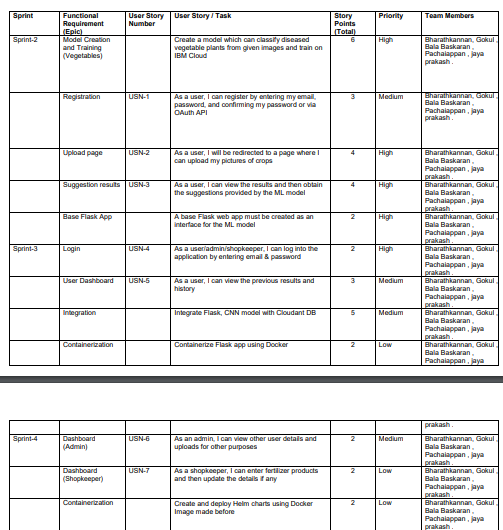


* 1. USER STORIES



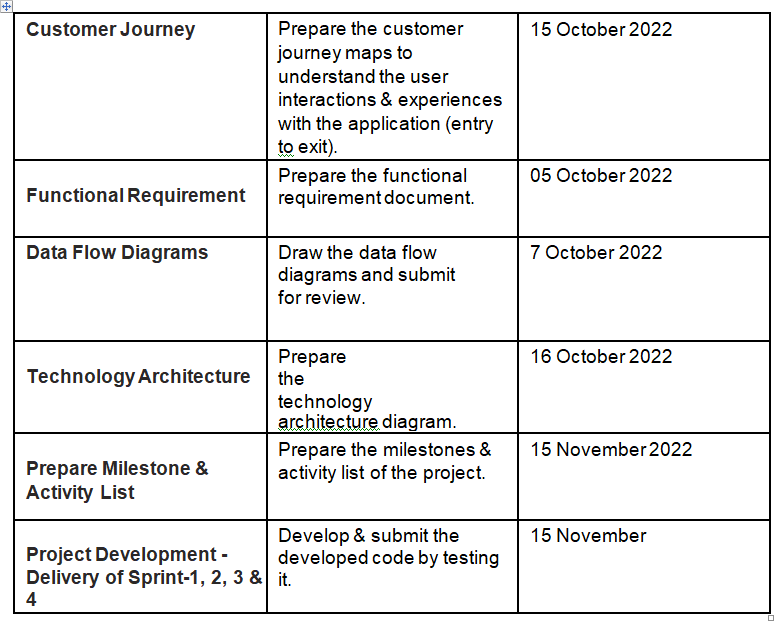
1. PROJECT PLANNING & SCHEDULING
   1. SPRINT PLANNING & ESTIMATION





* 1. SPRINT DELIVERY SCHEDULE





* 1. REPORT FROM JIRA

A burndown chart **shows the amount of work that has been completed in an epic or sprint, and the total work remaining**. Burndown charts are used to predict your team's likelihood of completing their work in the time available.



1. CODING & SOLUTIONING
   1. FEATURE 1

|  |  |  |
| --- | --- | --- |
|  | | |
| A. Source Code (Jupyter notebook python code) fruit.ipynb (due to limited page size the code vegetable.ipynb uploaded in github) #!/usr/bin/env python # coding: utf-8  # In[1]:  pwd  # In[2]:  cd E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset # # Apply ImageDataGenerator functionality to Train and Test set  # # Preprocessing # In[3]:  from keras.preprocessing.image import ImageDataGenerator  train\_datagen =  ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_ﬂi p=True)  test\_datagen = ImageDataGenerator(rescale=1) # In[4]:  pwd  # In[5]:  x\_train = train\_datagen.ﬂow\_from\_directory('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-  dataset/train',target\_size=(128,128),batch\_size=32,class\_mode='categorical') # In[6]:  x\_test=test\_datagen.ﬂow\_from\_directory('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset/test',target\_size=(128,128),  batch\_size=32,class\_mode='categorical')  # # Import the models # In[7]:  from tensorﬂow.keras.models import Sequential  from tensorﬂow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten  ## initializing the models  # In[8]:  model=Sequential() # # Add CNN Layers # In[9]:  model.add(Convolution2D(32,(3,3),input\_shape=(128,128,3),activation='relu')) # In[10]:  x\_train.class\_indices # # Add Pooling layer # In[11]:  model.add(MaxPool2D(pool\_size=(2,2))) # # Add Flatten layer  # In[12]:  model.add(Flatten()) # # Add Dense Layer # In[21]:  model.add(Dense(40, kernel\_initializer='uniform',activation='relu')) model.add(Dense(20, kernel\_initializer='random\_uniform',activation='relu')) # # Add Output Layer  # In[24]:  model.add(Dense(6,activation='softmax', kernel\_initializer='random\_uniform')) # # Compile the model  # In[25]:  model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['accuracy'  ])  # In[26]:  len(x\_train) # In[27]:  5384/32  # # Fit the Model # In[28]:  model.ﬁt\_generator(x\_train,steps\_per\_epoch=168,validation\_data=x\_test,validation\_st eps=52,epochs=3)  # # Save the Model # In[29]:  model.save("fruit.h5") # In[30]:  ls  # # Test the Model # In[32]:  from keras.preprocessing import image  from tensorﬂow.keras.preprocessing.image import img\_to\_array from tensorﬂow.keras.models import load\_model  import numpy as np # In[33]:  model = load\_model("fruit.h5")  # # Test Apple\_Healthy Class images # In[37]:  img = image.load\_img('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit- dataset/fruit-dataset/test/Apple healthy/00fca0da-2db3-481b-b98a- 9b67bb7b105c RS\_HL 7708.JPG',target\_size=(128,128)) | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

1. TESTING
   1. TEST CASES

A test case is nothing but **a series of step executed on a product, using a predefined set of input data, expected to produce a pre-defined set of outputs, in a given environment**. It describes “how” to implement those test cases. Test case specifications are useful as it enlists the specification details of the items.

The purpose of testing is to discover errors . Testing is the process of trying to discover every conceivable fault or weakness in a work product . It provide a way to check the functionality of component , sub assemblies , assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirement and user expectation and does not

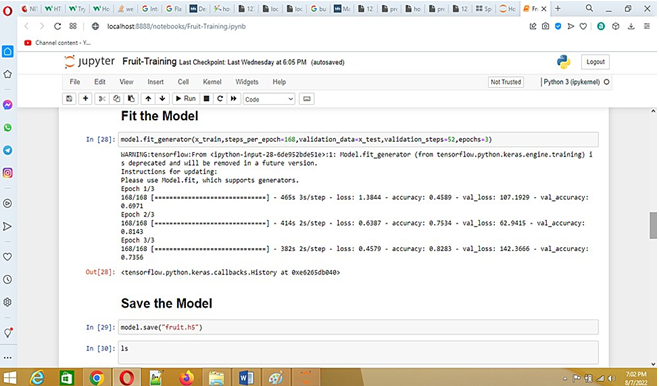
fail in an unacceptable manner. There are various types of testing. Each test type addressing a specific testing requirement.

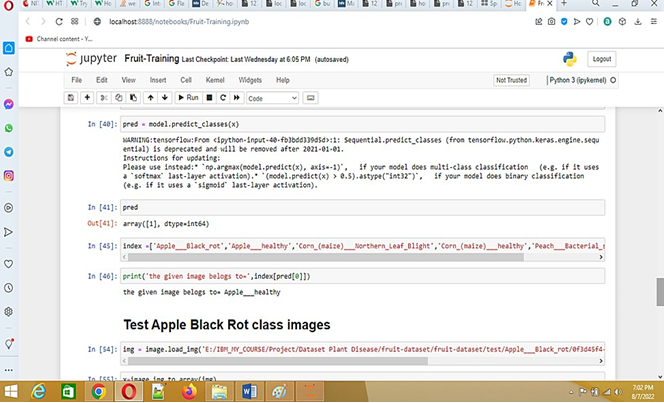
The testing report are submitted in github account.

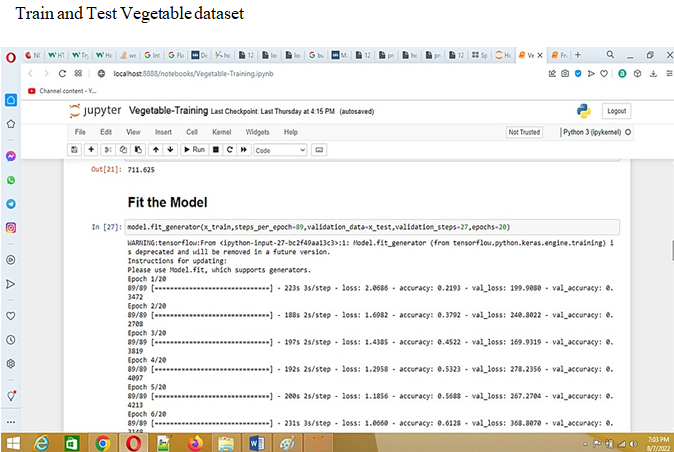
* 1. USER ACCEPTANCE TESTING

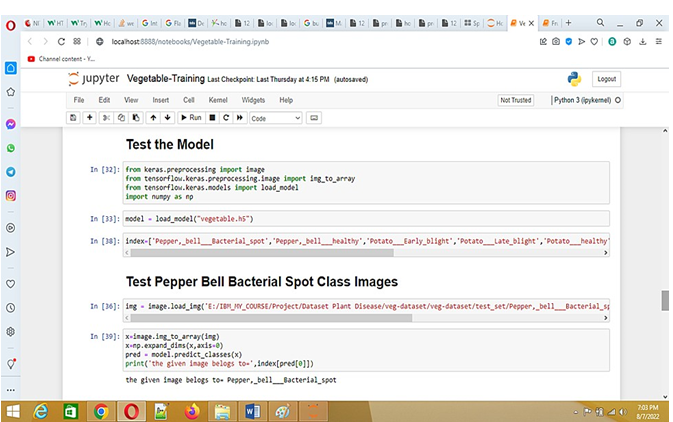
User acceptance testing is a critical phase of any project and requires significant participant by the end user. It also ensure that the system meets the functional requirement.

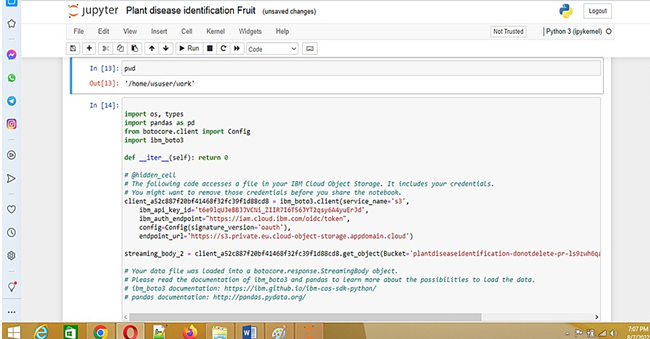
1. RESULTS

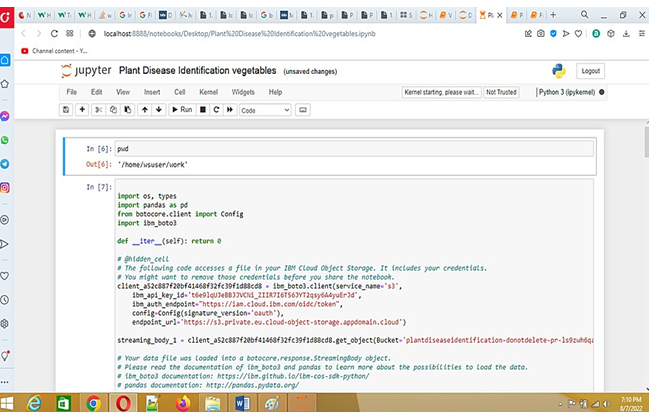












ADVANTAGE & DISADVANTAGE

ADVANTAGE

* 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

DISADVANTAGE

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

|  |  |  |  |
| --- | --- | --- | --- |
| APPLICATIONS  The trained network model used to classify the image patterns with high accuracy.  The proposed model not only used for plant disease classiﬁcation but also for other image pattern classiﬁcation such as animal classiﬁcation.  This project work application involves not only image classiﬁcation 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. |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# BIBILOGR R e ferences

[1]. [R Indumathi.](https://ieeexplore.ieee.org/author/37887897500); [N Saagari.](https://ieeexplore.ieee.org/author/37087060007); [V Thejuswini.](https://ieeexplore.ieee.org/author/37087058056); [R Swarnareka.](https://ieeexplore.ieee.org/author/37087056414)," Leaf Disease Detection and Fertilizer Suggestion", [IEEE International Conference on System, Computation,](https://ieeexplore.ieee.org/xpl/conhome/8870340/proceeding)

[Automation and Networking (ICSCAN)](https://ieeexplore.ieee.org/xpl/conhome/8870340/proceeding), 29-30 March 2019, DOI: [10.1109/ICSCAN.2019.8878781.](https://doi.org/10.1109/ICSCAN.2019.8878781)

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

[3]. H Shiva reddy, Ganesh hedge, Prof. DR Chinnaya3, "IoT based Leaf Disease

Detection and Fertilizer Recommendation", International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 11, Nov 2019, e-ISSN: 2395-

0056.

.

1. 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 vidoe 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.

1. APPENDIX PYTHON

Python is a computer programming language often used to **build websites and software, automate tasks, and conduct data analysis**. Python is a general- purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

SOURCE CODE

A. Source Code (Jupyter notebook python code)

# fruit.ipynb (due to limited page size the code vegetable.ipynb uploaded in github)

#!/usr/bin/env python # coding: utf-8

# In[1]:

pwd

# In[2]:

cd E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset # # Apply ImageDataGenerator functionality to Train and Test set

# # Preprocessing # In[3]:

from keras.preprocessing.image import ImageDataGenerator

train\_datagen =

ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_ﬂi p=True)

test\_datagen = ImageDataGenerator(rescale=1) # In[4]:

pwd

# In[5]:

x\_train = train\_datagen.ﬂow\_from\_directory('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-

dataset/train',target\_size=(128,128),batch\_size=32,class\_mode='categorical') # In[6]:

x\_test=test\_datagen.ﬂow\_from\_directory('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit-dataset/fruit-dataset/test',target\_size=(128,128),

batch\_size=32,class\_mode='categorical')

# # Import the models # In[7]:

from tensorﬂow.keras.models import Sequential

from tensorﬂow.keras.layers import Dense,Convolution2D,MaxPool2D,Flatten # # Initializing the models

# In[8]:

model=Sequential() # # Add CNN Layers # In[9]:

model.add(Convolution2D(32,(3,3),input\_shape=(128,128,3),activation='relu')) # In[10]:

x\_train.class\_indices # # Add Pooling layer # In[11]:

model.add(MaxPool2D(pool\_size=(2,2))) # # Add Flatten layer

# In[12]:

model.add(Flatten()) # # Add Dense Layer # In[21]:

model.add(Dense(40, kernel\_initializer='uniform',activation='relu')) model.add(Dense(20, kernel\_initializer='random\_uniform',activation='relu')) # # Add Output Layer

# In[24]:

model.add(Dense(6,activation='softmax', kernel\_initializer='random\_uniform')) # # Compile the model

# In[25]:

model.compile(loss='categorical\_crossentropy',optimizer='adam',metrics=['accuracy'

])

# In[26]:

len(x\_train) # In[27]:

5384/32

# # Fit the Model # In[28]:

model.ﬁt\_generator(x\_train,steps\_per\_epoch=168,validation\_data=x\_test,validation\_st eps=52,epochs=3)

# # Save the Model # In[29]:

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

ls

# # Test the Model # In[32]:

from keras.preprocessing import image

from tensorﬂow.keras.preprocessing.image import img\_to\_array from tensorﬂow.keras.models import load\_model

import numpy as np # In[33]:

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

# # Test Apple\_Healthy Class images # In[37]:

img = image.load\_img('E:/IBM\_MY\_COURSE/Project/Dataset Plant Disease/fruit- dataset/fruit-dataset/test/Apple healthy/00fca0da-2db3-481b-b98a- 9b67bb7b105c RS\_HL 7708.JPG',target\_size=(128,128))

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

PROJECT DEMO LINK

[*https://heart-disease-predictor-flask.herokuapp.com/*](https://heart-disease-predictor-flask.herokuapp.com/)

[IBM-EPBL](https://github.com/IBM-EPBL)/[**IBM-Project-41501-1660642520**](https://github.com/IBM-EPBL/IBM-Project-41501-1660642520)