PROJECT REPORT

Fertilizers Recommendation System for Disease Prediction

TeamID:PNT2022TMID03970

TEAMMEMBERS:

SNEHA E (TEAM LEAD) [412519106087]

NIVETHA M [412519106093]

KAVYA B [4125106065]

AARDRA KRISHNA C [412519106001]

TABLE OF CONTENTS

INTRODUCTION

1.1 PROJECT OVERVIEW:

- Agriculture is the most important sector in today's life. Most plants areaffectedbyawidevarietyofbacterialandfungaldiseases.Diseasesonp lantsplaced a major constraint on the production and a major threat tofoodsecurity.Hence,earlyandaccurateidentificationofplantdiseasesi sessentialtoensurehighquantityandbestquality.Inrecentyears,thenum ber of diseases on plants and the degree of harm caused has increaseddue to the variation in pathogen varieties, changes in cultivation methods,andinadequateplantprotection techniques.
- For this problem an automated system is introduced to identify different diseases on plantsby checking the symptoms shown on the leaves of the plant. Deep learningtechniquesareusedtoidentifythediseasesandsuggesttheprecau tionsthatcanbetaken for thosediseases.

1.2 PURPOSE:

- **Agricultural productivity is something on which economy highly depends**. This is the one of the reasons that disease detection in plants plays an important role in agriculture field, as having disease in plants are quite natural.
- The disease-based similarity measure is used for fertilizer recommendation. Detection and recognition of plant diseases using machine learning are very efficient in **providing symptoms of identifying diseases at its earliest**.
- Plant pathologists can analyze the digital images using digital image processing for diagnosis of plant diseases.
- Growers want to control pests and diseases **to get maximum production and prevent crop loss**. Prevention of pests and diseases before any damage is done is most desirable

2 LITERATURESURVEY:

2.1 ExistingProblem:

- Adequate mineral nutrition is central to crop production. However, it canalsoexertconsiderableInfluenceondiseasedevelopment.Fertilizerapplication can increase or decrease development of diseases caused bydifferentpathogens,andthemechanismsresponsiblearecomplex,includingeffectsofnutrientsonplantgrowth,plantresistancemechanismsanddirecteffects on the pathogen. The effects of mineral nutrition on plant diseaseand the mechanisms responsible for those effects have been dealt withcomprehensively elsewhere. In India, around 40% of land is kept and grownusing reliable irrigation technologies, while the rest relies on the monsoonenvironmentforwater.Irrigationdecreasesrelianceonthemonsoon,increasesfood security,and boostsagriculturalproduction.
- Most research articles use humidity, moisture, and temperature sensorsneartheplant'sroot,withanexternaldevicehandlingallofthedata provided by the sensors and transmitting it directly to an external display oranAndroidapplication.Theapplicationwascreatedtomeasuretheapproximate values of temperature, humidity and moisture sensors thatwereprogrammedintoamicrocontrollertomanagetheamountof water.
- The focus of this research work is on two popular architectures: AlexNet and GoogLeNet. The authors of this work investigated the performance of both of these architectures on the PlantVillage dataset by training the model from scratch in one case and then using transfer learning to adjust already trained modelsin the other. A model based on the Training From Scratch method achieves an accuracy of around 91.80% when photos other than those from the dataset are tested, and an accuracy of around 31% when images not from the dataset are tested. Pre-trained deep learning models AlexNet and VGG16 are

used to detect diseases in plants as well as a healthy plant using photographs taken immediately from the field with cell phones.

2.2 REFERENCE:

- FertilizersRecommendationSystemForDiseasePredictionInTreeLeave|SemanticScholar
- SoilBasedFertilizerRecommendationSystemforCropDiseasePredictionSystem (ijetajournal.org)
- LeafDiseaseDetectionandFertilizerSuggestion|IEEEConferencePublication|IEEEXplore
- IRJET-V7I1004.pdf
- Anutrientrecommendationsystemforsoilfertilizationbasedonevolutionarycomputation-ScienceDirect
- Fertilizers-Recommendation-System-For-Disease-Prediction-In-Tree-Leave.pdf(ijstr.org)
- 2204.11340.pdf(arxiv.org)
- 371-376,Tesma405,IJEAST.pdf
- CROFED-CropandFertilizerRecommendationandDiseasediagnosissystemusingMachineLearningandInternetofThings.(ijirt.org)
- PredictionofCrop, Fertilizer andDiseaseDetection for PrecisionAgriculture byIJRASET-Issuu

2.3 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 inappropriate crop for their land based on a conventional or non-scientific approach. This is a challengingtask 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.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:

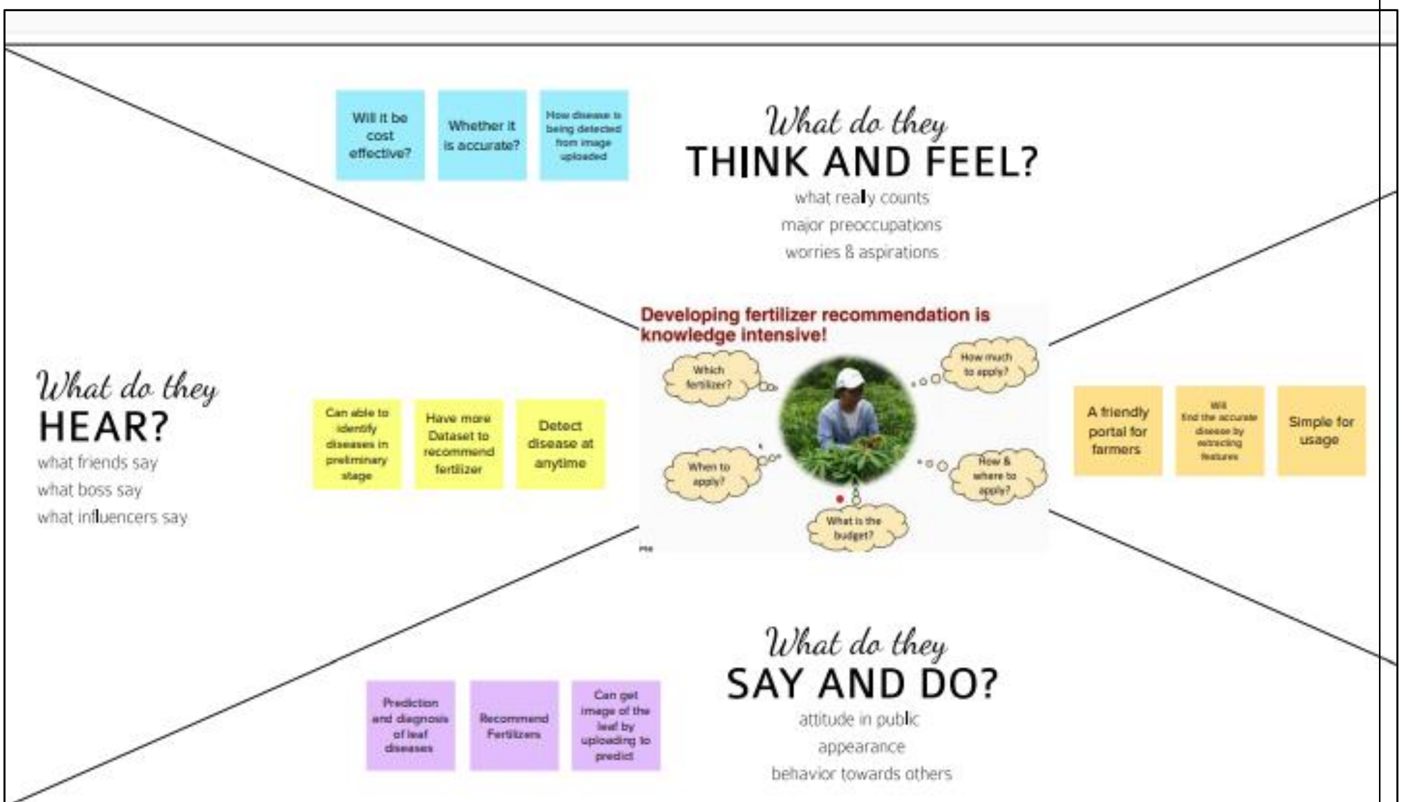


FIG [1]

3.2BRAINSTROMING:

3.3 PROPOSED SOLUTION:

- The proposed solution of this project uses Deep Learning algorithm to classify leaves, and identify the disease and suggest the fertilizers.
- The Deep learning solution includes the MobileNetV2 and VGG19

model for training.
- ❖ Based on the leaf disease detected, the model recommend fertilizer for prevention.
- ❖ The Farmers, Researches are the end users get benefited by this system.

NOVELTY

- ❖ More accurate than other models.
- ❖ The model is embedded in a website which is easy to use by the customers.
- ❖ This system is more robust by incorporating more image dataset with wider variations.
- ❖ This system also estimates the probability of the infected plant.

FEASIBILITY

- ❖ Improves accuracy, generality and training efficiency.
- ❖ Quick diagnosis of disease which is a significant part in early detection of disease.
- ❖ Farmers can easily interact with the portal through simple User Interface.
- ❖ Can reduce the cost which may occur due to wrongly used fertilizer.

SCALABILITY

- ❖ It helps the farmers to pick the right fertilizer toward the start of the product cycle and amplify the yield.
- ❖ This system can be used by anyone in the world.
- ❖ Instantly gives the results.

SOCIAL IMPACT

- ❖ Plant growth can be enhanced.
- ❖ Ensures plants are getting supplied with every nutrient they need.
- ❖ Multiple crops yields every season.
- ❖ It help support people's nutritional needs.

3.4 SOLUTION FIT:

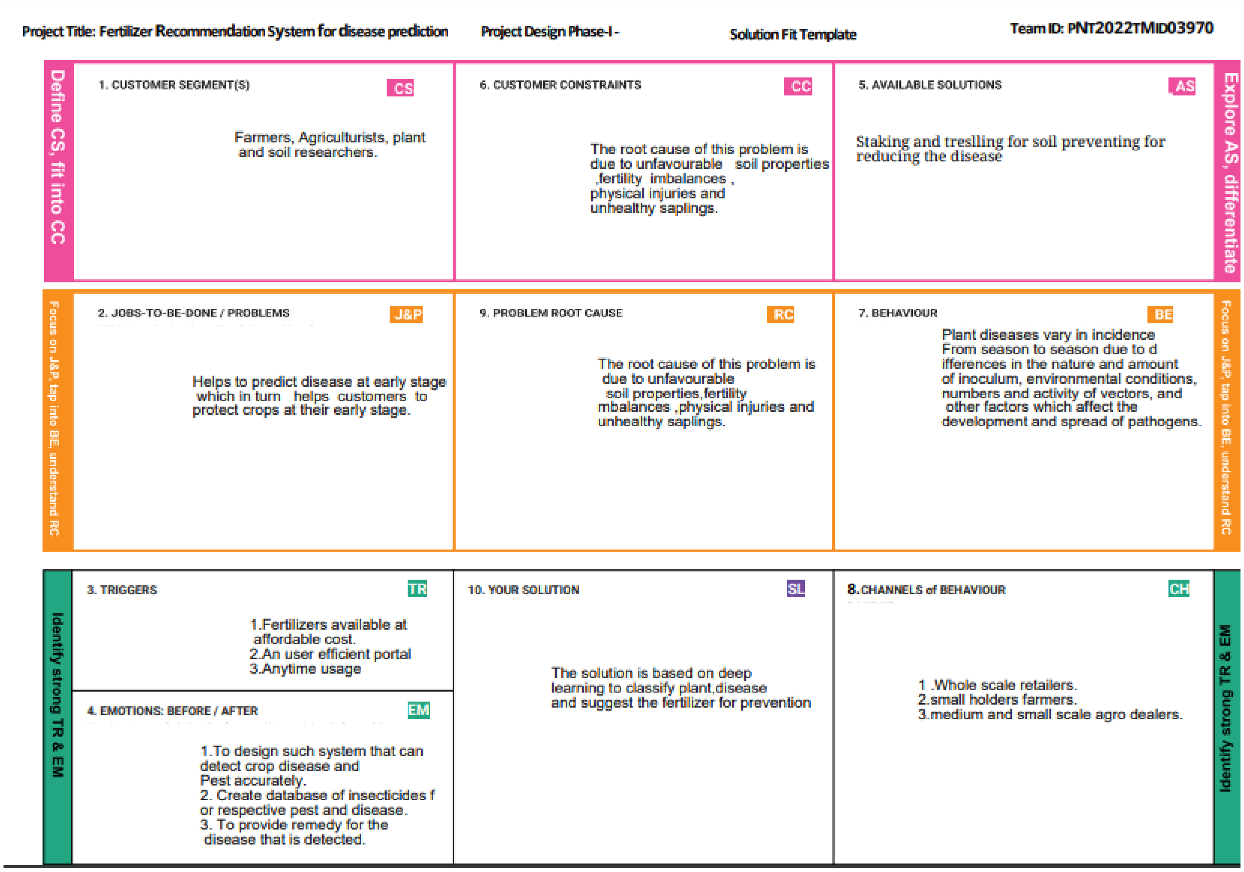


FIG [2]

4. REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

The following are the functional requirements of the proposed solution

FR NO	FUNCTIONAL REQUIREMENT(EPIC)	SUB REQUIREMENT(STORY/SUB-TASK)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Profile	Filling the profile page after logging in
FR-4	Uploading Data(Leaf)	Image of the leaves is to be

		uploaded
FR-5	Requesting solution	Uploaded image is compared with the pre-defined model and solution is generated
FR-6	Fertilizer Recommendation	Based on the type of disease identified, suitable fertilizers are recommended

4.2NON-FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution

FR NO	NON-FUNCTIONAL REQUIREMENTS	DESCRIPTION
NFR-1	Usability	The system allows the user to perform the task easily, efficiently and effectively.
NFR-2	Security	Information about the user and their data’s are highly secured with the authorization technology
NFR-3	Reliability	The model deployed should be reliable and able to give accurate disease prediction and recommendation.
NFR-4	Performance	Response time and total processing time is fast.
NFR-5	Availability	The application should be available anytime and anywhere to all the registered users.
NFR-6	Scalability	Increase in the number of user does not affect the performance of the system.

5. PROJECT DESIGN:

5.1 DATA FLOW DIAGRAM:

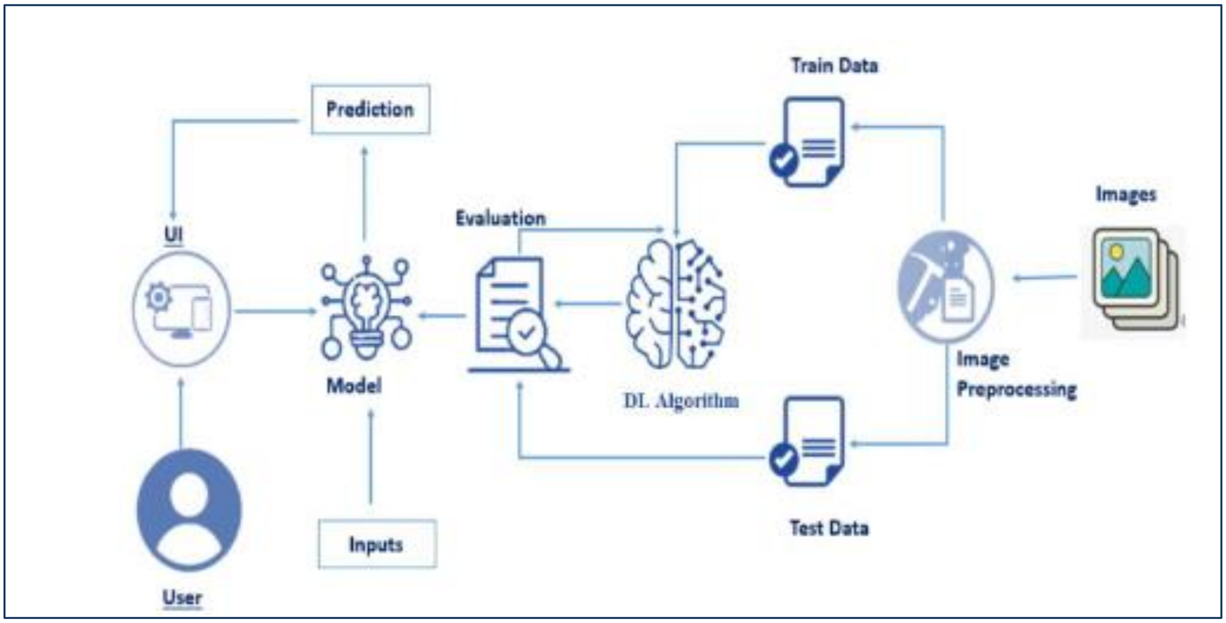


FIG [3]

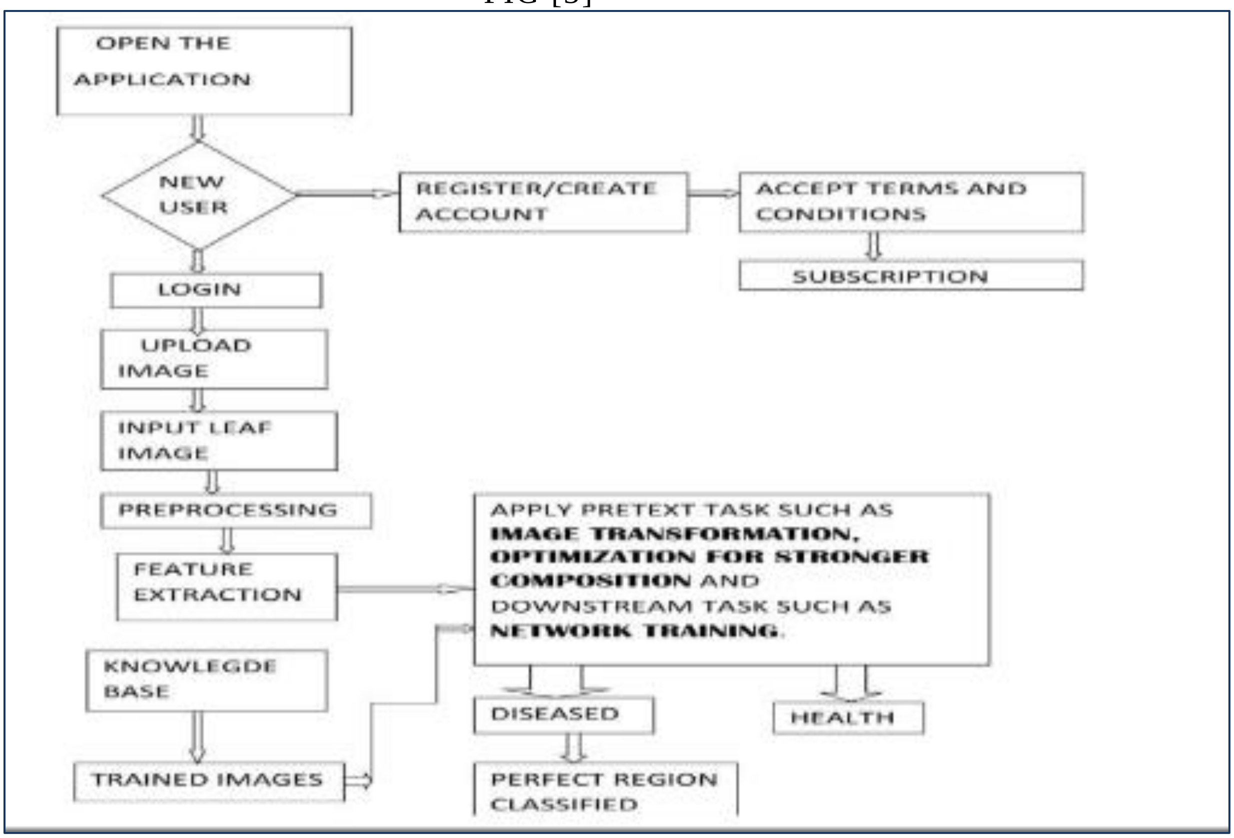


FIG [4]

5.2SOLUTION ARCHITECTURE:

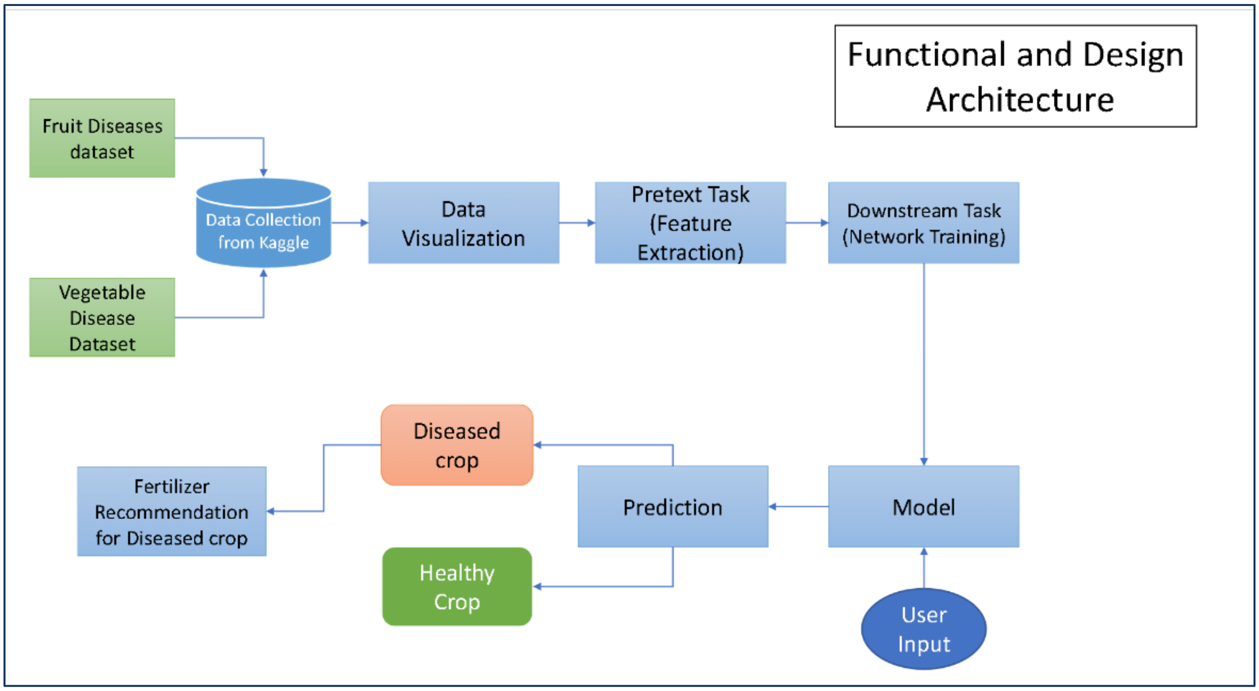
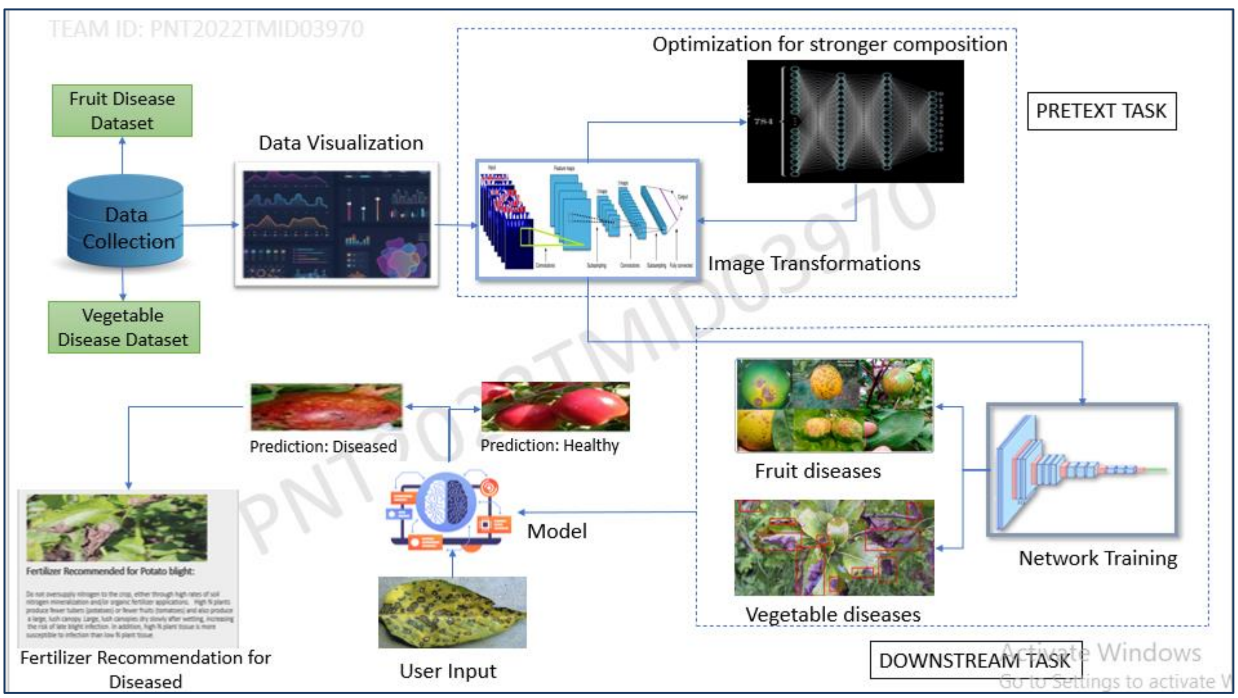


FIG [5]

TECHINCAL ARCHITECTURE:



Toaccomplishtheabovetask,youmustcompletethebelowactivitiesandt asks:

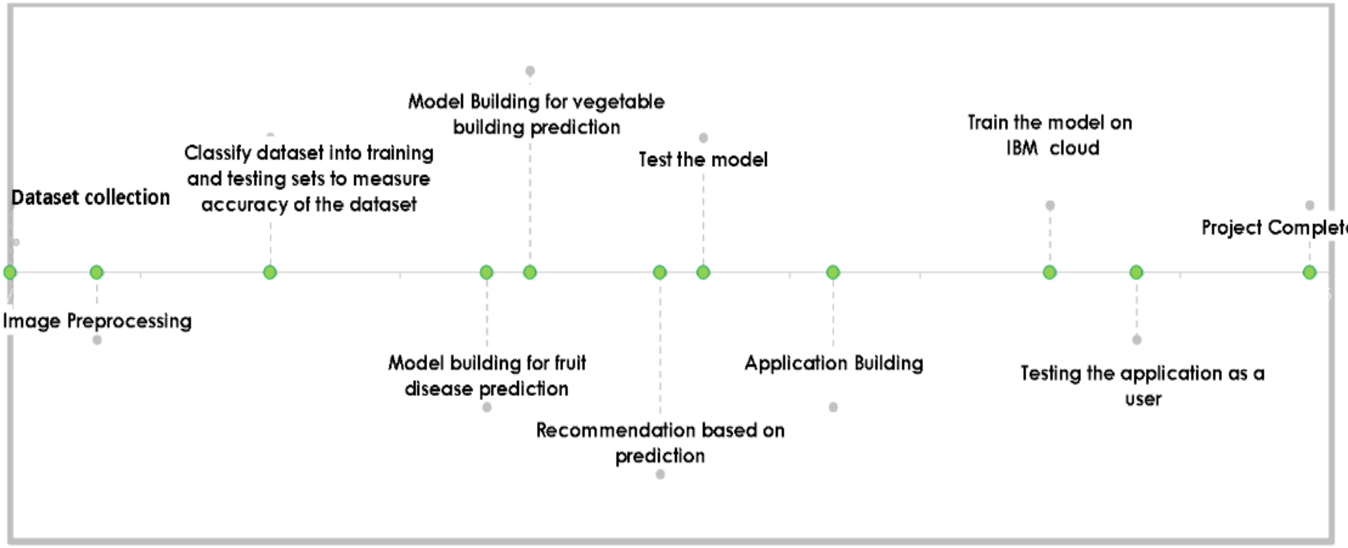
- Downloadthedataset.
- Classifythedatasetintotrainandtestsets.
- Addtheneuralnetworklayers.
- Loadthetrainedimagesandfitthemodel.
- Testthemodel.
- Savethemodeland itsdependencies.
- BuildaWebapplication usingaflaskthatintegrateswiththemodelbuilt.

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
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer (Web 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
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1

PROJECT PLANNING& SCHEDULING:

SPRINT PLANNING & ESTIMATION: FIG[6]



SPRINT DELIVERY SCHEDULE:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	MODELLING PHASE	USN-1	As a user (farmers),I have a problem in identifying the actual plant disease and to recommend best fertilizer.	4	High	SNEHA E,NIVETHA M
Sprint-1		USN-2	Collection of Dataset (sample images of disease affected leaves of different varieties of plants).	2	Medium	SNEHA E,NIVETHA M
Sprint-1		USN-3	Image Preprocessing-Preprocess the collected affected leaf images(rotating to grayscale,calling etc)	3	Low	AARDRA KRISHNA C,KAVYA B
Sprint-1		USN-4	Classify dataset into training and testing sets to measure accuracy of dataset	4	Medium	SNEHA E
Sprint-2	DEVELOPMENT PHASE	USN-5	Model building-Create a CNN model for fruit disease predicton.	5	High	SNEHA E,AARDRA KRISHNA C,NIVETHA M,KAVYA B
Sprint-2		USN-6	Model building-Create a CNN model for vegetable prediction.	4	High	NIVETHA M,KAVYA B
Sprint-2		USN -7	Recommendation based on prediction.	5	High	AARDRA KRISHNA C,KAVYA B

Sprint-2		USN-8	Test the model	3	Medium	SNEHA E,AARDRA KRISHNA C
Sprint-3	DEPLOYMENT PHASE	USN-9	Application building- Creating Login page (to login the user with phone number and email id)	2	low	SNEHA E,NIVETHA M
Sprint-3		USN-10	Dashboard page Creation(contains user profiles and predicting accuracy). In Input page, images of diseased leaf are uploaded.	3	Medium	AARDRA KRISHNA C,KAVYA B
Sprint-3		USN-11	Prediction page creation- Show the prediction based on the user input.	3	Medium	SNEHA E,AARDRA KRISHNA C
Sprint-4	TESTING PHASE	USN-12	Train the model on the IBM cloud	4	High	NIVETHA M,KAVYA B
Sprint-4		USN-13	Testing the application as a user –all user interfaces will be working properly and check prediction accuracy.	4	High	SNEHA E NIVETHA M

CODING AND SOLUTIONS:

FEATURES:

FEATURE 1:

- Dataset is classified as Training and Testing. We will add features to train the dataset by using Convolutional Neural Network(Convolutional neural networks are a specialized type of artificial neural networks that use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers. They are specifically designed to process pixel data and are used in image recognition and processing.)
- The detection accuracy is enhanced with proposed algorithm

```
In [8]:  import tensorboard
         tensorboard.__version__

Out[8]:  '2.5.0'

In [9]:  #model building using MobileNetV2

         from tensorflow.keras.applications import MobileNetV2
         basemodel=MobileNetV2(weights="imagenet",include_top=False,
                               input_tensor=Input(shape=(224,224,3)))
         model=basemodel.output
         model = tf.keras.models.Sequential([
             tf.keras.layers.Conv2D(100, (3,3), activation='relu', input_shape=(150, 150, 3)),
             tf.keras.layers.MaxPooling2D(2,2),

             tf.keras.layers.Conv2D(100, (3,3), activation='relu'),
             tf.keras.layers.MaxPooling2D(2,2),

             tf.keras.layers.Flatten(),
             tf.keras.layers.Dropout(0.5),
             tf.keras.layers.Dense(50, activation='relu'),
             tf.keras.layers.Dense(9, activation='softmax')

         ])
         #model summary
         print(basemodel.summary())
```

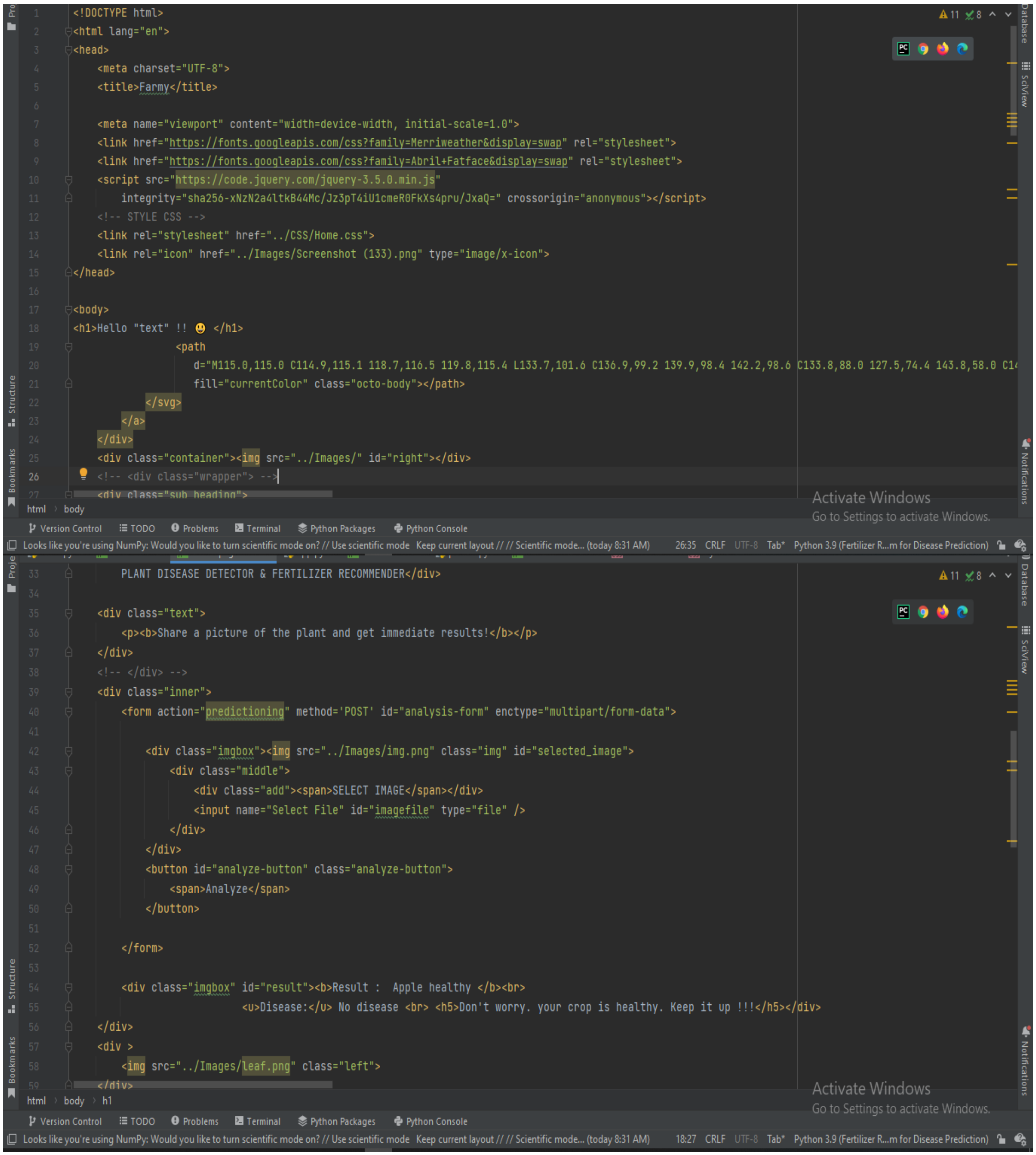
```
         Total params: 2,257,984
         Trainable params: 2,223,872
         Non-trainable params: 34,112
         ______________________________________________________________________________________________
         None

In [10]: (train_images, train_labels), _ = keras.datasets.fashion_mnist.load_data()
         train_images = train_images / 255.0

In [11]: logdir="logs/fit/" + datetime.now().strftime("%Y%m%d-%H%M%S")
         tensorboard_callback = keras.callbacks.TensorBoard(log_dir=logdir)

         # Train the model.
         opt=tf.keras.optimizers.Adam(learning_rate=0.001)

         model.compile(optimizer=opt,
                       loss='sparse_categorical_crossentropy',
                       metrics=['acc'])


         history = model.fit(
                     train_generator,
                     validation_data=validation_generator,
                     epochs=3,
                     verbose=1)
```


FIG[6]]

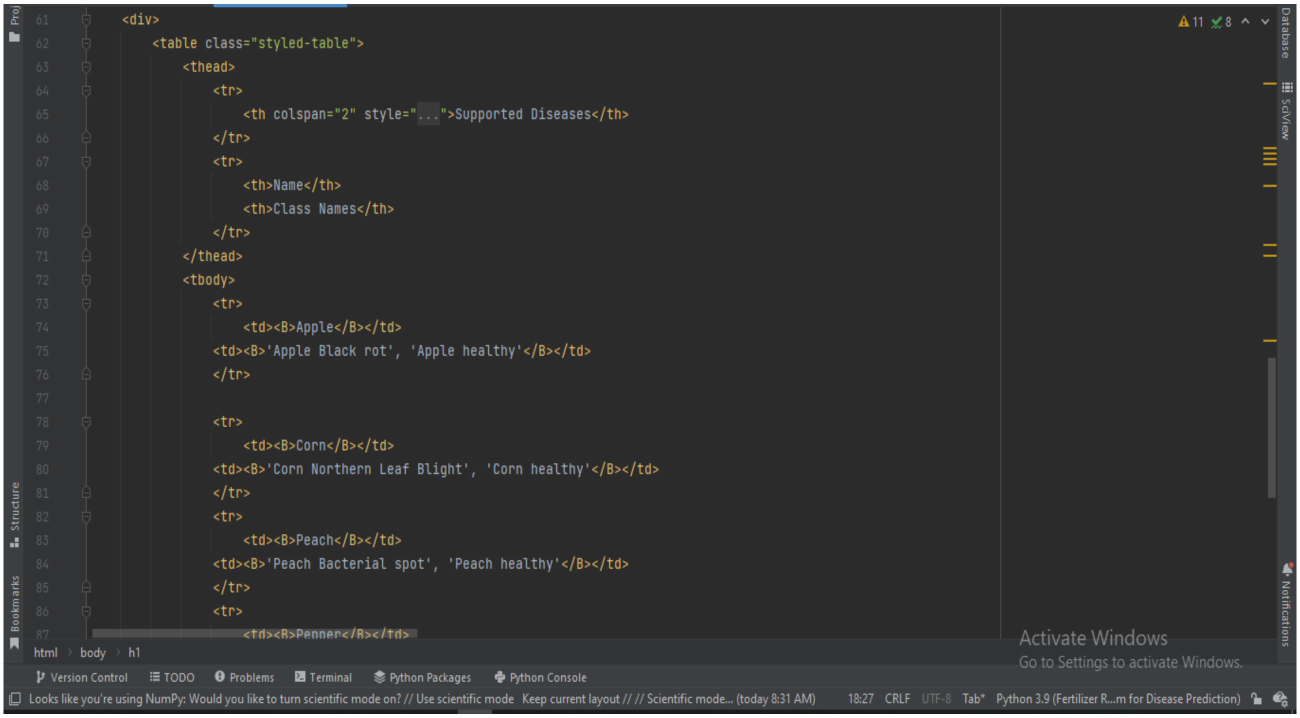
FEATURE 2:

- We are creating a Login page to the existing user and registration form for New User.
- Displaying the supported diseases and recommend the fertilizers for the disease.
- Easy to Check.
- Cost Efficient.

- It also provides environment friendly recovery measures of the identified disease.



FIG[7]



FIG[8]

8. TESTING:

8.1 Testing the fruit disease prediction and vegetable disease prediction

TESTED VEGETABLE DISEASE PREDICTION:

Predicted Output

In [26]:
```
img=image.load_img(r"/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/test_set/Pepper,_bell___healthy/b06117a8-6ca6-4b82-96cf-07604beb8f1b___JR_HL
img
```

Out[26]:

In [28]:
```
img=image.load_img(r"/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/test_set/Tomato___Late_blight/aff90d5c-a3f2-445a-b125-717a00b21fb8___RS_Late.
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=['Pepper,_bell___Bacterial_spot','Pepper,_bell___healthy','Potato___Early_blight','Potato___Late_blight','Potato___healthy','Tomato___Bacterial_spo
print("Predicted Output: ",index[y[0]])
```

```
1/1 [==============================] - 0s 52ms/step
Predicted Output:  Pepper,_bell___healthy
```

In [ ]:

In [29]:
```
img=r"C:\Users\HP\Dataset Plant Disease\Veg-dataset\Veg-dataset\test_set\Pepper,_bell___healthy\ba527e73-8be2-4c53-9e5b-cafeab987bba___JR_HL 8037.JPG"
s=load_img(img,target_size=(150,150,3))
s
```

Out[29]:

In [30]:
```
output(img)
```

```
** Pepper,_bell___healthy With probability 97.65
```

In [31]:
```
img=r"C:\Users\HP\Dataset Plant Disease\Veg-dataset\Veg-dataset\test_set\Potato___Early_blight\b7157976-61c2-4366-87c5-e3de23aa7c10___RS_Early.B 7227.
s=load_img(img,target_size=(150,150,3))
s
```

Out[31]:

In [32]:
```
output(img)
```

```
** Potato___Early_blight With probability 99.53
```

FIG[9]

TESTED FRUIT DISEASE PREDICTION:

PREDICTED OUTPUT:

```
img=r"C:\Users\HP\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Apple___healthy\0adc1c5b-8958-47c0-a152-f28078c214f1___RS_HL 7825.JPG"
s=load_img(img,target_size=(150,150,3))
s
```

Out[17]:

In [18]:

```
output(img)
```

```
** Apple___healthy With probability 80.21
```

In [19]:

```
img=r"C:\Users\HP\Dataset Plant Disease\fruit-dataset\fruit-dataset\test\Corn_(maize)___Northern_Leaf_Blight\5df7a420-aeab-4eb0-959a-290e4bb2a944___RS
s=load_img(img,target_size=(150,150,3))
s
```

Out[19]:

In [20]:

```
output(img)
```

```
** Corn_(maize)___Northern_Leaf_Blight With probability 98.21
```

FIG[10]

8.2 USER ACCEPTANCE TESTING

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
Yellow Leaves	10	4	5	15	34
Blights	1	5	2	4	12
Fruit rots	3	1	0	2	6
Leaf spots	9	2	4	18	33
Mosaic leaf pattern	3	9	6	6	24
Fruit Spots	3	1	5	1	10
Leaves misshapen	0	7	2	1	10
Totals	29	29	24	47	129

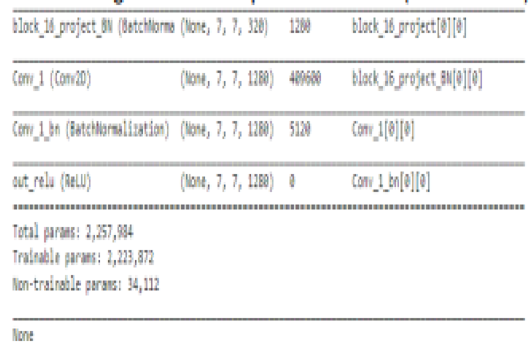
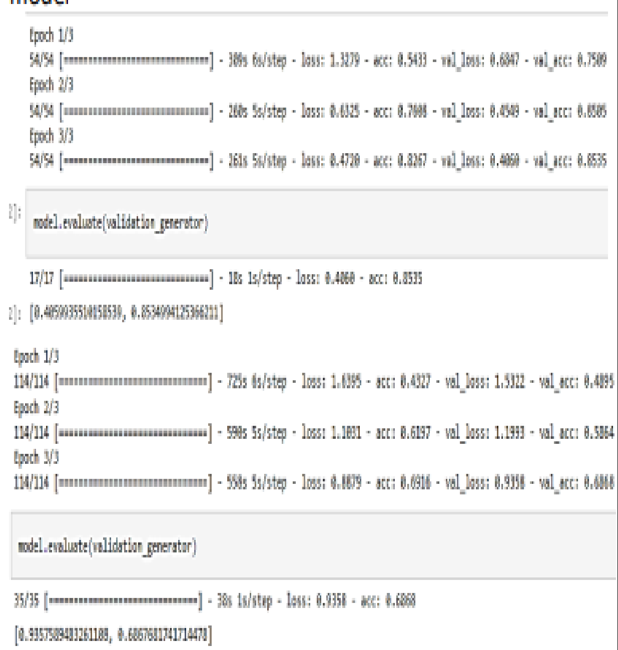
TEST CASES

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

Section	Total Cases	Not Tested	Fail	Pass
Yellow Leaves	20	0	0	20
Blights	43	0	0	43
Fruit rots	9	0	0	9
Leaf spots	5	0	0	5
Mosaic leaf pattern	19	0	0	19
Fruit Spots	2	0	0	2
Leaves misshapen	4	0	0	4

9. RESULT:

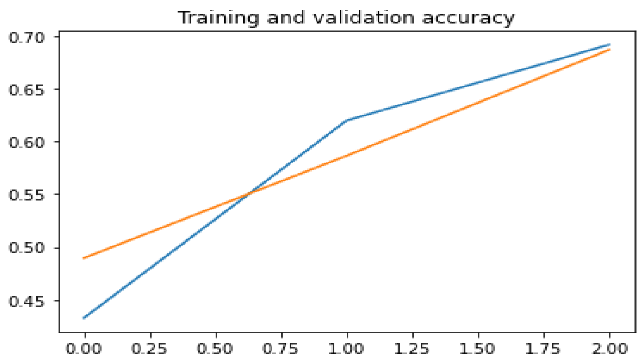
9.1 PERFORMANCE METRICS:

S.No	Parameter	Values	Screenshot
1.	Model Summary	-Total params: 2,257,984 -Trainable params: 2,223,872 -Non-trainable params: 34,112	Fruit and Vegetable disease prediction Model (MobileNetV2)  block_16_project_BN (BatchNorma (None, 7, 7, 320) 1280 block_16_project[0][0] Conv_1 (Conv2D) (None, 7, 7, 1280) 409600 block_16_project_BN[0][0] Conv_1_bn (BatchNormalization) (None, 7, 7, 1280) 5120 Conv_1[0][0] out_relu (ReLU) (None, 7, 7, 1280) 0 Conv_1_bn[0][0] Total params: 2,257,984 Trainable params: 2,223,872 Non-trainable params: 34,112 None
2.	Accuracy	**Training Accuracy -** -Fruit disease prediction 82.67% -Vegetable disease prediction 69.16% **Validation Accuracy –** -Fruit disease prediction 85.35% -Vegetable disease prediction 68.68%	Fruit disease prediction Model & vegetable disease prediction model  Epoch 1/3 54/54 [==============================] - 309s 6s/step - loss: 1.3279 - acc: 0.5433 - val_loss: 0.6847 - val_acc: 0.7509 Epoch 2/3 54/54 [==============================] - 260s 5s/step - loss: 0.6325 - acc: 0.7008 - val_loss: 0.4549 - val_acc: 0.8505 Epoch 3/3 54/54 [==============================] - 261s 5s/step - loss: 0.4720 - acc: 0.8267 - val_loss: 0.4060 - val_acc: 0.8535]: model.evaluate(validation_generator) 17/17 [==============================] - 18s 1s/step - loss: 0.4060 - acc: 0.8535]: [0.4059035510158539, 0.8534994125366211] Epoch 1/3 114/114 [==============================] - 725s 6s/step - loss: 1.6395 - acc: 0.4327 - val_loss: 1.5322 - val_acc: 0.4895 Epoch 2/3 114/114 [==============================] - 590s 5s/step - loss: 1.1031 - acc: 0.6197 - val_loss: 1.1993 - val_acc: 0.5864 Epoch 3/3 114/114 [==============================] - 550s 5s/step - loss: 0.8879 - acc: 0.6916 - val_loss: 0.9358 - val_acc: 0.6868 model.evaluate(validation_generator) 35/35 [==============================] - 38s 1s/step - loss: 0.9358 - acc: 0.6868 [0.9357589483261108, 0.6867681741714478]

In [34]:

```
acc      = history.history['acc' ]
val_acc  = history.history[ 'val_acc' ]
loss     = history.history[    'loss' ]
val_loss = history.history['val_loss' ]

epochs   = range(len(acc))

plt.plot  ( epochs,     acc )
plt.plot  ( epochs, val_acc )
plt.title ('Training and validation accuracy')
plt.figure()
```

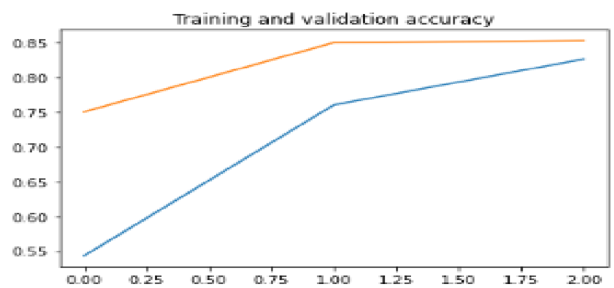
Out[34]:



In [21]:

```
acc      = history.history['acc' ]
val_acc  = history.history[ 'val_acc' ]
loss     = history.history[    'loss' ]
val_loss = history.history['val_loss' ]

epochs   = range(len(acc))

plt.plot  ( epochs,     acc )
plt.plot  ( epochs, val_acc )
plt.title ('Training and validation accuracy')
plt.figure()

plt.plot  ( epochs,     loss )
plt.plot  ( epochs, val_loss )
plt.title ('Training and validation loss'   )
```

Out[21]: Text(0.5, 1.0, 'Training and validation loss')



10.ADVANTAGES:

- ❖ Helps to predict disease at early stage which in turn helps customers to protect crops at early stages.
- ❖ User efficient portal with better accuracy
- ❖ Staking and trellising for soil prevention for reducing the disease.
- ❖ Display prediction with probability

DISADVANTAGES:

- ❖ For training and testing the proposed model requires very high computational.
- ❖ The neural network architecture used in this project 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,differentclassificationaccuraciesareobtained.

- 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:

This further research is implementing the proposed algorithm with the existing public datasets. Also, various segmentation algorithms can be implemented to improve accuracy. The proposed algorithm can be modified further to identify the disease that affects the various plant organs such as stems and fruits.

13. APPENDIX

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-20054-1659711520

DEMO LINK: