AGRO SPY

A Farmer's Company

PROJECT REPORT
B. TECH CSE FINAL YEAR
(Academic Session 2016-20)

Group Members: **Shivank Agrahari (1602053007)**
Ashish Kumar Pandey (1602053022)
Vishnu (1602053005)

Project Guide: Mr. Mrinal Paliwal
(Assistant Professor, CSE Department)

Candidate’s Declaration

We hereby certify that the work which is being presented in the project report entitled “PROJECT AGROSPY ” by “Shivank Agrahari, Ashish Kumar Pandey, Vishnu Chaudhury” in partial fulfillment of requirements for the award of degree of B.Tech. (CSE) submitted in the Department of Computer Science and Engineering at Sanskriti University, Mathura, UP is an authentic record of my own work carried out during a period from 5th January to 25th May. We also certify that this assignment/report is our own work, based on our personal study and/or research and that We have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. We further certify that this assignment/report has not previously been submitted for assessment in any other unit, except where specific permission has been granted from all unit coordinators involved, or at any other time in this unit, and that We have not copied in part or whole or otherwise plagiarized the work of other students and/or persons.

Student Name:

Roll No.

Signature of the Student

Date

Certificate

This is to certify that this project report entitled “AGROSPY” by Shivank Agrahari ,Roll no.- 1602053007,Ashish Kumar Pandey, Roll no.- 1602053022,Vishnu Chaudhury, Roll no.-1602053005 , submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering of the Sanskriti University, Mathura, UP during the academic year 2016-20, is a bonafide record of work carried out in a satisfactory way.

EXTERNAL EXAMINER	PROJECT GUIDE	HOD
	Mr. Mrinal Paliwal (Assist. Prof)	

Date:

Acknowledgements

We would like to place on record our deep sense of gratitude to **Mr Mrinal Palliwal** and **Mr Mohit Mishra** ,Dept. of Computer Science and Engineering, Sanskriti University, Mathura, UP for there generous guidance, help and useful suggestions. We express our sincere gratitude to (Name of HOD), HoD in Dept. of Computer Science and Engineering, Sanskriti University, Mathura, UP for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We also wish to extend our gratitude to **Google Cloud Platform Ltd.** for providing us free credits for deploying Virtual Machines on their cloud infrastructure.

We also thanks to **Mr. Fahad Hussain** for providing us quality tutorial on Deep Learning and Convolutional Neural Network.

We are extremely thankful to Dr. Suresh Kaswan, Dean Engg., Sanskriti University, Mathura, UP for providing me infrastructural facilities to work in, without which this work would not have been possible. We further certify that this assignment/report is my own work, based on our personal study and/or research and that We have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication.

Student Name

Roll No.

Signature of the Student

Date:

Abstract

Our Project Agrospy is designed with the view to enhance agricultural techniques with technological advancement. Agriculture is one of the primary occupations in all over the world. The demand of agricultural productivity increases day by day with the increase in world population. Due to lack of sophisticated techniques, diseases,infections and scarcity of proper knowledge to farmers for controlling and monitoring their fields, productivity suffers. Our project takes this problem as a challenge and makes uses of Deep Learning technology to monitor crop health by capturing image samples of crop leaves.

During several past years, Deep Learning has been emerged as one of the most widely used Artificial Intelligence technique. It is widely used for binary and multi class classification problems. This project also utilizes the ability of Deep Learning Classification to solve a problem that is related to agriculture. This project uses AI technique to make a health scan of the crops, it provides health report card by testing the crop leaves.

This project consists of 48 different classes of healthy and unhealthy crop samples for predicting the diseases. We also studied different crops species and their diseases to provide symptoms and control measures information to farmers when AgroSpy finds crop image sample not well.

Table of Contents

1. Candidate’s Declaration……………………………………………………2
2. Certificate………………………………………………………………….3
3. Acknowledgement…………………………………………………………4
4. Abstract……………………………………………………………………5
5. List of Tables………………………………………………………………7
6. List of Figures……………………………………………………………..8
7. Chapter 1: Introduction……………………………………………………9
8. Chapter 2: Literature Review……………………………………………..10
9. Chapter 3: Problem Definition and Requirement Analysis…………….11
10. Chapter 4: Design and Implementation…………………………………13
11. Chapter 5: Testing and Deployment…………………………………….67
12. Chapter 6: Future Enhancements………………………………………71
13. Summary…………………………………………………………………...72
14. References…………………………………………………………………73

List of Tables

1. Task to be Completed with Selected Technology……………………….7
2. Datasets Used……………………………………………………………13
3. Crop Species Classes and Samples Collected……………………………14
4. Neural Network used in Project………………………………………….15
5. Testing and Training Accuracy in Several Epochs………………………67

List of Figures

1. Flow Chart of Web Server Module………………………20
2. Application Flow Chart…………………………………..21
3. Layout Design Prototype…………………………………22
4. AgroSpy version 1.0 Landing Page………………………34
5. AgroSpy version 1.0 Details Page………………………..34
6. Landing Page……………………………………………..42
7. Gallery Upload Page……………………………………...42
8. Result Page………………………………………………..42

Chapter 1: Introduction

The Project deals with one of the major problems that resists agricultural productivity, that is improper and inadequate monitoring of crop health that results in ruin of whole crop fields.
We took this as a challenge into our project and used our engineering skills to solve this problem. In this Project, we are using **Deep Convolutional Neural Network** technique to solve the problem of crop disease identification and classification.
Our idea behind this project is very simple but effective. We have developed an application **AgroSpy** that a user (usually farmer or Agriculture Professionals) can use to scan their crops, this app uses Artificial Intelligent techniques to identify diseases in their crops. Users can either use their phone camera to capture live samples or use an old picture of their crop saved in gallery. Once the user uploads the picture of crop sample, AgroSpy tries to find the name of crop and also predicts that whether crop sample is suffering from any disease or not.If it finds occurrence of any disease in the sample it provides detailed symptoms and treatment information to the user in the report card.
In this way, it helps to predict the disease that the crops might be suffering from.
We have started with preparation of 48 different classes of crop diseases that has been seen in various crops. These classes can be increased when one collects more samples.

Chapter 2: Literature Review

A Crop/Weed Field Image Dataset for the Evaluation of Computer Vision Based Precision Agriculture Tasks Sebastian Haug1(B) and J¨orn Ostermann2

This paper has given us enough insights about how computer vision can be applied to agriculture related tasks. This paper takes a dataset and uses machine learning approach for solving classification problems related to agriculture.

Using Deep Learning for Image-Based Plant Disease Detection
Sharada P. Mohanty 1, 2, 3, David P. Hughes 4, 5, 6 and Marcel Salathé1, 2, 3

This paper was related more closely to our problem. Here Deep Convolutional Neural Network is used to build classifiers for crop disease identification. We come to know about approach for CNN models and we also lernt about various standard CNN models like AlexNet, Inception Net etc. that suits for our problem.

Deep Learning with Python
Francois Collet

This book gives enough concepts about Deep Learning and Use of Keras API over Tensorflow

Problem Definition & Requirement Analysis

Problem Statement

The problem statement of our project is very straightforward:

> **“Given an agricultural crop leaf image sample as input we need to predict its crop species and also detect whether crop is in healthy condition or it is suffering from any disease.**
> **If crop is suffering from any disease predict its name along with the symptoms and control management strategies for that disease.”**

Requirement Analysis

While doing the task of requirement analysis, we thought over different approaches that we can follow to solve our problem.
By our research and analysis ,we found two major approaches to solve our problem, these are as following:

A. **Naïve Image Processing** :- Since it was an image related problem, our first approach was to use image processing algorithms to solve this problem. Our possible solving approach was to detect some sort of pattern in the crop image sample by performing various image processing operations such as color space change, image contouring etc.
   By using such techniques we had also developed a program for detecting wheat leaf rust into the crop sample.
   After a long hassling with the algorithms we were successful in our event and ended with such conclusion:
   i. Program worked fine after a long effort, but it was just for a single disease of a crop and doing this sort of manual feature grasping will be very tedious and time consuming for more species.
   ii. Program gives false result when the orientation and exposure in the sample varies.

   With this conclusion we omitted this approach and looked for some better alternatives.

B. **Deep Convolutional Neural Network**: - Artificial Neural Network is one of the emerging AI technique which makes a simulation of human brain structure to learn and adopt new things, it consists of a network of perceptron, a digital equivalent to human brain neuron. This network has been used widely to solve various real-world regression and classification problems.

But our research says that in order to solve problems that are related to Computer Vision a slightly different variant of Neural Network is being used. This is known as **Convolutional Neural Network.** It is a variant of Deep Neural Network in which we use Convolution and Max Pooling operations, followed by a ReLU(Rectified Linear Unit) activation function operations. The result obtained from these operations is called Feature Map. It is a collection of 2D Matrices. This Feature Map is converted into a single one-dimensional vector by performing flattening operation. Now this 1D vector is fed into the fully connected neural network for classification.

It is one of the most promising technique for Computer Vision Problems. After trying some mini projects similar to our problem, we found this technique most suitable for our task with following conclusions:

i. This AI technique use self-learning approach, that is one of the necessary approaches for feature identification. Neural network uses back propagation to identify the classifying features on its own.

ii. It is suitable for multi class classification, and many big projects like Face App, Google Lens etc. uses this technology.

So, we selected Deep Convolutional Network as our applicable approach.

Task Division

After selecting CNN as applicable approach, our next step was to find different major tasks that needs to be done for accomplishing our project. These are as follows:

Tasks	**Technology Used**
Neural Network Modeling	Keras API ,backend with Tensorflow on Python 3.6 environment
Data set Collection	PlantVillage, ImageNet datasets and web scrapped images
Web Application Server	Flask framework on Python 3.6
Neural Network Training	Google Cloud VM Instance with Nvidia GPU
Application Development	Android Studio
Application UI Designing	Adobe XD
Web App Deployment	Google App Engine

Table 1 Tasks to be Completed with Selected Technology

Chapter 3: Design and Implementation

We have followed a top down approach to create our design. We start our design by taking combined approach as a whole and further we tear down each module into detail.

Design: Combined Approach

Algorithm for Project:

- **I. Collect Dataset.**
- **II. Develop CNN model.**
- **III. Train CNN model**
- **IV. Deploy Model on Web Server**
- **V. Develop Android Application**
- **VI. Test Infrastructure**

I. Collect Dataset

We studied during requirement analysis that before deploying a deep learning model it is good to check for available data related to our project. Fortunately, we got much datasets available publicly that was very helpful for our research.

Dataset used in our project is given in the table:

Serial Number	Dataset Name	No of Classes	Quantity
1	Plant Village	38	40298
2	UCI Rice Dataset	3	120
3.	Digi pathos	10	10218
4.	Web Scrapped	48	Around 4000 images

Table 2. Dataset Used

Table structure for table crop_info

Column	Type	Null	Default
crop_id	int(11)	No	
crop_name	Text	No	
crop_disease	Text	No	
crop_symtom	Text	No	
crop_treatment	Text	No	
crop_reference	Text	No	

Data for table crop_info

1	Apple	Apple Scab
2	Apple	Black Rot
3	Apple	Cedar Rust
4	Apple	Healthy
5	Blueberry	Healthy
6	Cherry	Healthy
7	Cherry	Powder Mildow
8	Corn	Grey Leaf Spot
9	Corn	Common Rust
10	Corn	Healthy
11	Corn	Northern Leaf Blight
12	Grape	Black Rot
13	Grape	Esca
14	Grape	Healthy
15	Grape	Leaf Blight
16	Orange	Haunlongbling
17	Peach	Bacterial Spot
18	Peach	Healthy
19	Pepper	Bacterial Spot
20	Pepper	Healthy
21	Potato	Early Blight
23	Potato	Late Blight
24	Raspberry	Healthy
25	Rice	Bacterial Leaf Blight
26	Rice	Brown Spot
27	Rice	Healthy
28	Rice	Hispa
29	Rice	Leaf Blast
30	Rice	Leaf Smut
31	Syabean	Healthy
32	Squash	Powdery Mildow
33	Strawberry	Healthy
34	Strawberry	Leaf Scorch
35	Tomato	Bacterial Spot
36	Tomato	Early Blight
38	Tomato	Late Blight
37	Tomato	Healthy

22	Potato	Healthy
39	Tomato	Leaf Mold
40	Tomato	Leaf Spot
41	Tomato	Spider Mites
42	Tomato	Target Spot
43	Tomato	Mosaic Virus
44	Tomato	Yellow Leaf Curl Virus
45	Wheat	Heathy
46	Wheat	Leaf Rust
47	Wheat	Stem Rust
48	Non Crop Sample	None

Table 3. Crop Species Classes Collected

II. Develop CNN Model

By reading various literatures on Deep Learning, we found that there are some standard CNN models that can be designed for achieving good accuracy.
We also saw in various research papers related to our project where these standard convolutional neural networks are used as reference to develop neural network model. We also used these standard models as a reference to develop five neural models.

These models are given into table:

Name	**Convolution Layers**	**Max Pooling Layers**	**Fully Connected Layer**	**Total No of Classes**
Alex Net	5	3	4	48
Inception Net	12	2	3	48
VGG Net	13	5	3	48
Res Net	6	0	2	48
Custom Model	2	2	2	48

Table 4. Neural Networks used in Project

Model Summary

We are giving structural design of models that we used for training :

i. **Alex Net**

```
Layer (type)                Output Shape            Param #
=================================================================
input_1 (InputLayer)        [(None, 256, 256, 3)]     0
_________________________________________________________________
conv2d (Conv2D)             (None, 62, 62, 96)        34944
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 30, 30, 96)        0
_________________________________________________________________
conv2d_1 (Conv2D)           (None, 7, 7, 256)         614656
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 3, 3, 256)         0
_________________________________________________________________
conv2d_2 (Conv2D)           (None, 3, 3, 384)         885120
_________________________________________________________________
conv2d_3 (Conv2D)           (None, 3, 3, 384)         1327488
_________________________________________________________________
conv2d_4 (Conv2D)           (None, 3, 3, 256)         884992
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 1, 1, 256)         0
_________________________________________________________________
flatten (Flatten)           (None, 256)               0
_________________________________________________________________
dense (Dense)               (None, 9216)              2368512
_________________________________________________________________
dense_1 (Dense)             (None, 4096)              37752832
_________________________________________________________________
dense_2 (Dense)             (None, 4096)              16781312
_________________________________________________________________
dense_3 (Dense)             (None, 48)                196656
=================================================================
Total params: 60,846,512
Trainable params: 60,846,512
```

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

Table 4 Alex Net Model Design

ii. VGG Net

```
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 256, 256, 64)      1792
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 256, 256, 64)      36928
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 128, 64)      0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 128, 128, 128)     73856
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 128, 128, 128)     147584
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 64, 64, 128)       0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 64, 64, 256)       295168
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 64, 64, 256)       590080
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 64, 64, 256)       590080
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 32, 256)       0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 32, 32, 512)       1180160
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 32, 32, 512)       2359808
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 32, 32, 512)       2359808
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 16, 16, 512)       0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 16, 16, 512)       2359808
```

```
_________________________________________________________________
conv2d_13 (Conv2D)          (None, 16, 16, 512)      2359808
_________________________________________________________________
conv2d_14 (Conv2D)          (None, 16, 16, 512)      2359808
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 8, 8, 512)        0
_________________________________________________________________
flatten (Flatten)           (None, 32768)            0
_________________________________________________________________
dense (Dense)               (None, 4096)             134221824
_________________________________________________________________
dense_1 (Dense)             (None, 4096)             16781312
_________________________________________________________________
dense_2 (Dense)             (None, 48)               196656
=================================================================
Total params: 165,914,480
Trainable params: 165,914,480
Non-trainable params: 0
_________________________________________________________________
```

iii. Custom Net

```
Layer (type)                Output Shape             Param #
=================================================================
input_1 (InputLayer)        [(None, 256, 256, 3)]    0
_________________________________________________________________
conv2d (Conv2D)             (None, 254, 254, 16)     448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 127, 127, 16)     0
_________________________________________________________________
conv2d_1 (Conv2D)           (None, 125, 125, 32)     4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 62, 62, 32)       0
_________________________________________________________________
flatten (Flatten)           (None, 123008)           0
_________________________________________________________________
dense (Dense)               (None, 512)              62980608
_________________________________________________________________
dense_1 (Dense)             (None, 48)               24624
=================================================================
Total params: 63,010,320
Trainable params: 63,010,320
```

Non-trainable params: 0

III. Train CNN Model

For Training CNN model, we have created Augmented Dataset from Main Dataset and used 80% of the total images for training purpose.

Design for Creating Augmented Dataset

Scale	**1-100%**
Shear	**1-20%**
Zoom	**1-20%**
Flip	**Horizontal**

Training logistics is given in implementation section.

IV. Deployment on Web Server

After training neural model we save it along with trained weights for final deployment on web server. Since our application will communicate to this server. Web server is chosen in such a way so that it gives dynamic response to the client application.
Web server will be responsible for

- Taking input image from AgroSpy application
- Execution of Neural Network Prediction Module
- Fetching relevant data from Database Server
- Sending the response detail in suitable format

For better illustration we have provided flow chart for the underlying web server module is given in next image:

CNNProcessOnServer() Flow chart:

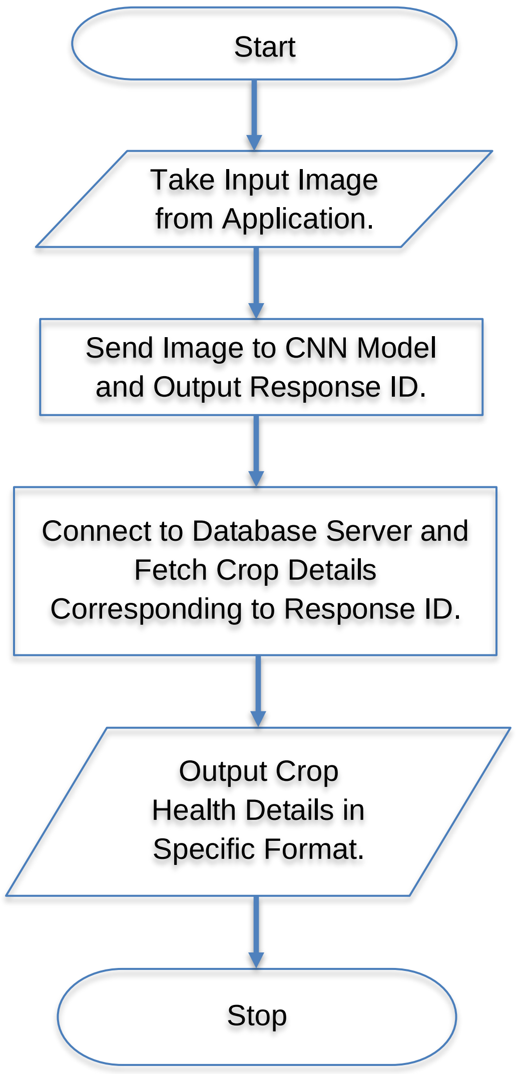

Start
Take Input Image from Application.
Send Image to CNN Model and Output Response ID.
Connect to Database Server and Fetch Crop Details Corresponding to Response ID.
Output Crop Health Details in Specific Format.
Stop

Figure 1. Flow Chart of Web Server Module

V. Development of Android Application

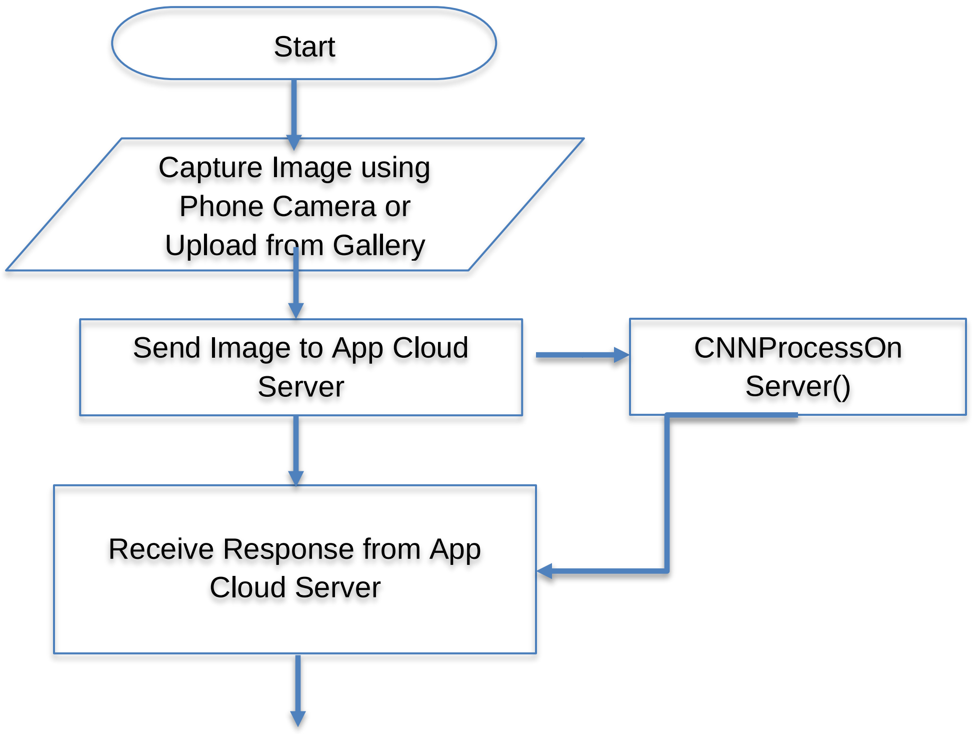
Android is a multidevice Operating System widely used in smartphones. It support lots of application support with robust development environment.
It covers more than 75% of the smartphone market. Hence we have chosen this platform as our primary application development environment.

Our purpose behind development of this application is to provide a comfortable user experience for indirect interaction with Neural Network.

Various responsibilities of this application are:

- Provide a rich UI experience to User.
- Take Image samples from Smartphone Camera or Gallery.
- Send Request to Application Cloud Web Server.
- Receive and Parse the Response from Web Server.
- Display information to user in specific format.

Flow Chart for Client Application


Start
Capture Image using Phone Camera or Upload from Gallery
Send Image to App Cloud Server
CNNProcessOn Server()
Receive Response from App Cloud Server

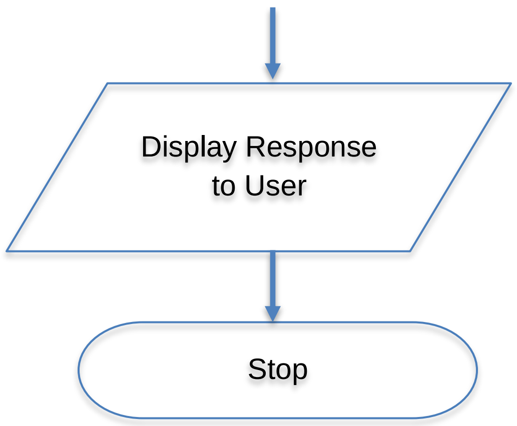


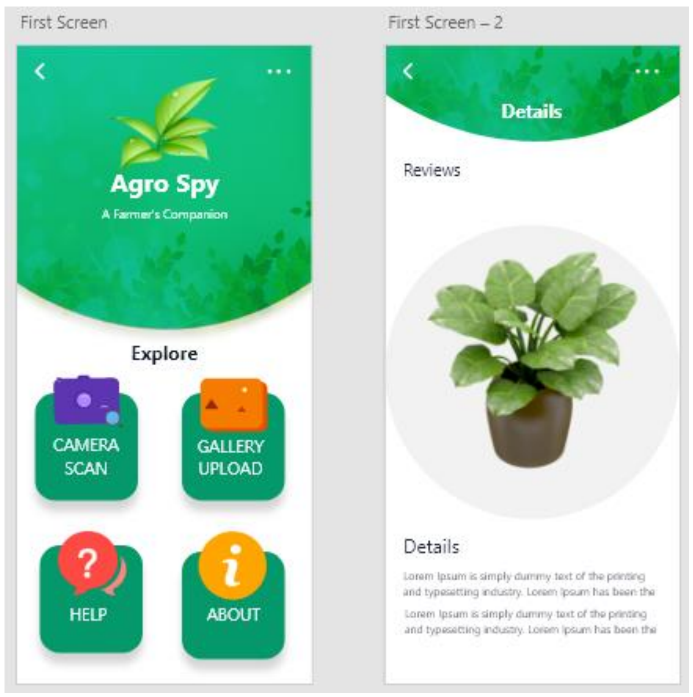
Figure 2. Application Flow Chart

Application User Interface Design

User Interface designing is done using Adobe XD Software. We tried our best to provide a rich user experience.

We faced several difficulties while designing layout that suits for smartphones of all sizes but we overcome it using continuous testing and analysis.

Design Prototype is shown in Figure 3



Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the

Figure 3. Layout Design Prototype

VI. Testing Infrastructure

Whole infrastructure is tested against different environments and exposure for getting accuracy.
This process was limited to web images only due to lockdown at that time.

Implementation

As per our study and decision made in Requirement Analysis phase implementation of whole project is done using suitable technology.

During implementation, we worked on different technology for completing different requirements.

We are showing our whole implementation related programming work by breaking it into 4 modules.

A. Convolutional Neural Network Training Code

Technologies Used for Implementation of CNN Model

- **Python 3.6:** High level language with wide support of libraries for Machine Learning and Deep Learning
- **Keras API :** Keras is a high level neural network API , written in Python and capable of running on top TensorFlow , CNTK or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.
  It allows users for easy and fast prototyping (through user friendliness, modularity and extensibility). It runs seamlessly on CPU, GPU and TPU.
- **TensorFlow:** TensorFlow is free and open source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural network.
- **Flask Micro Framework:** It is a popular web server microframework implemented for deploying web application on Python. It can be extended with SQL alchemy, form validation libraries as per our requirement into the project.

i. Implementation Code for Alex Net

```
import os
base_dir = 'D:/Main_Database'
train_dir = os.path.join(base_dir,'train')
val_dir = os.path.join(base_dir,'val')

from tensorflow.keras import layers
from tensorflow.keras import Model

img_input = layers.Input(shape = (256,256,3))
x = layers.Conv2D(96,11,activation = 'relu',strides = (4,4))(img_input)
x = layers.MaxPooling2D(3,strides = (2,2))(x)
x = layers.Conv2D(256,5,activation = 'relu',strides = (4,4))(x)
x = layers.MaxPooling2D(3,strides = (2,2))(x)
x = layers.Conv2D(384,3,activation = 'relu',padding = 'same')(x)
x = layers.Conv2D(384,3,activation = 'relu',padding = 'same')(x)
x = layers.Conv2D(256,3,activation = 'relu',padding = 'same')(x)
x = layers.MaxPooling2D(3,strides = (2,2))(x)
x = layers.Flatten()(x)
x = layers.Dense(9216,activation = 'relu')(x)
x = layers.Dense(4096,activation = 'relu')(x)
x = layers.Dense(4096,activation = 'relu')(x)
output = layers.Dense(48,activation = 'softmax')(x)
model = Model(img_input, output)
model.summary()
model.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics = ['acc'])

from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen =ImageDataGenerator(
rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
training_set =
train_datagen.flow_from_directory(train_dir,target_size=(256,256),batch_size =
256,class_mode='categorical')

test_datagen =ImageDataGenerator(
rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
testing_set = test_datagen.flow_from_directory(val_dir,target_size=(256,256),batch_size
= 256,class_mode='categorical')

model.fit_generator(training_set,steps_per_epoch = 500,epochs = 20,validation_data =
testing_set,validation_steps = 350)
from tensorflow.keras.models import model_from_json
json_model = model.to_json()
with open("crop_disease_model_alex.json",'w') as json_file:
    json_file.write(json_model)
model.save_weights('model_alex.h5')
```

ii. Implementation Code for Inception Net

```
import os

base_dir = 'C:/Users/smartshiwank_agr/Downloads/SourceCode/Main_Database'
train_dir = os.path.join(base_dir,'train')
val_dir = os.path.join(base_dir,'val')

from tensorflow.keras import layers
from tensorflow.keras import Model
from keras.layers.merge import concatenate

def inception_module(layer_in,f1,f2_in,f2_out,f3_in,f3_out,f4_out):
   conv1 = Conv2D(f1,(1,1),padding='same', activation = 'relu')(layer_in)
   conv3 = Conc2D(f2_in,(1,1),padding = 'same',activation = 'relu')(layer_in)
   conv3 = Conv2D(f2_out,(3,3),padding = 'same',activation = 'relu')(layer_in)
   conv5 = Conv2D(f3_in,(1,1),paddding = 'same',activation = 'relu')(layer_in)
   conv5 = Conv2D(f3_out,(5,5),paddding = 'same',activation = 'relu')(layer_in)
   pool = MaxPooling((3,3),strides =(1,1),padding = 'same')(layer_in)
   pool = Conv2D(f4_out,(1,1),padding = 'same',activation = 'relu')(pool)
   #concatenate
   layer_out = concatenate([conv1,conv3,conv5,pool],axis = -1)
   return layer_out
img_input = layers.Input(shape = (256,256,3))
x = inception_module(img_input,64,96,128,16,32,32)
x = inception_module(img_input,128,128,64,128,32,32)
x = layers.Flatten()(x)
x = layers.Dense(4096,activation = 'relu')(x)
output = layers.Dense(48,activation = 'softmax')(x)
model = Model(img_input, output)
model.summary()
model.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics =
['acc'])
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen =ImageDataGenerator(
rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
```

```
training_set =
train_datagen.flow_from_directory(train_dir,target_size=(256,256),batch_size =
256,class_mode='categorical')

test_datagen =ImageDataGenerator(
rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
testing_set =
test_datagen.flow_from_directory(val_dir,target_size=(256,256),batch_size =
256,class_mode='categorical')
model.fit_generator(training_set,steps_per_epoch = 1000,epochs =
15,validation_data = testing_set,validation_steps = 700)
from keras.models import model_from_json
json_model = model.to_json()
with open("crop_disease_model_inception_net.json",'w') as json_file:
    json_file.write(json_model)
model.save_weights('model_inceptionnet.h5')
```

iii. Implementation Code for VGG Net

```
import os

base_dir = 'C:/Users/smartshiwank_agr/Downloads/Main_Database'
train_dir = os.path.join(base_dir,'train')
val_dir = os.path.join(base_dir,'val')
from tensorflow.keras import layers
from tensorflow.keras import Model
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten

def vgg_block(layer_in,n_filters,n_conv):
    for _ in range(n_conv):
        layer_in = Conv2D(n_filters,(3,3),padding='same' ,activation = 
'relu')(layer_in)
    layer_in = MaxPooling2D((2,2),strides =(2,2))(layer_in)
    return layer_in
img_input = layers.Input(shape = (256,256,3))
x = vgg_block(img_input,64,2)
x = vgg_block(x,128,2)
x = vgg_block(x,256,3)
x = vgg_block(x,512,3)
x = vgg_block(x,512,3)
x = layers.Flatten()(x)
x = layers.Dense(4096,activation = 'relu')(x)
x = layers.Dense(4096,activation = 'relu')(x)
output = layers.Dense(48,activation = 'softmax')(x)
model = Model(img_input, output)
model.summary()
model.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics = 
['acc'])

from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen =ImageDataGenerator(
rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
training_set =
train_datagen.flow_from_directory(train_dir,target_size=(256,256),batch_size =
256,class_mode='categorical')

test_datagen =ImageDataGenerator(
rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
testing_set =
test_datagen.flow_from_directory(val_dir,target_size=(256,256),batch_size =
256,class_mode='categorical')
model.fit_generator(training_set,steps_per_epoch = 1000,epochs =
15,validation_data = testing_set,validation_steps = 700)
from keras.models import model_from_json
json_model = model.to_json()
with open("crop_disease_model_vggnet.json",'w') as json_file:
    json_file.write(json_model)
model.save_weights('model_vggnet.h5')
```

iv. Implementation of ResNet

```
import os

base_dir = 'C:/Users/smartshiwank_agr/Downloads/Main_Database'
train_dir = os.path.join(base_dir,'train')
val_dir = os.path.join(base_dir,'val')

from tensorflow.keras import layers
from tensorflow.keras import Model
from keras.layers import add

def residual_module(layer_in,n_filters):
    merge_input = layer_in
    if layer_in.shape[-1] != n_filters:
        merge_input = Conv2D(n_filters,(1,1),padding = 'same',activation =
'relu',kernel_initailizer = 'he_normal')(layer_in)
    conv1 = Conv2D(n_filters,(3,3), padding = 'same', activation ='relu',
kernel_initializer = 'he_normal')(layer_in)
    conv2 = Conv2D(n_filters , (3,3), padding = 'same',activation = 'linear',
kernel_initializer = 'he_normal')(conv1)
    layer_out = add([conv2,merge_input])
    layer_out = Activation('relu')(layer_out)
    return layer_out
img_input = layers.Input(shape = (256,256,3))
x = residual_module(img_input,64)
x = residual_module(x,128)
x = layers.Flatten()(x)
x = layers.Dense(4096,activation = 'relu')(x)
output = layers.Dense(48,activation = 'softmax')(x)
model = Model(img_input, output)
model.summary()
model.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics =
['acc'])
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen =ImageDataGenerator(
rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
```

```
training_set =
train_datagen.flow_from_directory(train_dir,target_size=(256,256),batch_size =
256,class_mode='categorical')

test_datagen =ImageDataGenerator(
rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)
testing_set =
test_datagen.flow_from_directory(val_dir,target_size=(256,256),batch_size =
256,class_mode='categorical')
model.fit_generator(training_set,steps_per_epoch = 1000,epochs =
15,validation_data = testing_set,validation_steps = 700)
from keras.models import model_from_json
json_model = model.to_json()
with open("crop_disease_model_res_net.json",'w') as json_file:
    json_file.write(json_model)
model.save_weights('model_resnet.h5')
```

B. Implementation on Web Server

Web server is implemented using flask microweb framework on Python. Implementation Code is given below :

```
import os
import flask
import werkzeug as wz
import numpy as np
import sqlite3
import json
UPLOAD_FOLDER = 'uploads'
ALLOWED_EXTENSIONS = set(['txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'])
app = flask.Flask(__name__)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
host = '127.0.0.1'
port = 5000
@app.route('/', methods=['GET', 'POST'])
def handle_request():
    return flask.render_template("upload.html", host=host, port=port)


@app.route('/uploaded', methods=['GET', 'POST'])
def show_data():
    if flask.request.method == 'POST':
        img = flask.request.files['image']
        if img.filename == '':
            print('No selected file')
        if img:
            filename = wz.utils.secure_filename(img.filename)
            img_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
            img.save(img_path)
        json_file = open('models/crop_disease_model_alex.json', 'r')
        loaded_model_json = json_file.read()
        json_file.close()
        from tensorflow.keras.models import model_from_json
        loaded_model = model_from_json(loaded_model_json)
        loaded_model.load_weights('models/model_alex.h5')
        from tensorflow.keras.preprocessing import image
        test_image = image.load_img(img_path, target_size=(256, 256))
        test_image = image.img_to_array(test_image)
```

```
            test_image = np.expand_dims(test_image, axis=0)
            category = loaded_model.predict(test_image)
            result = category.argmax() + 1
            result = int(result)
            conn = sqlite3.connect('crop_info.db')
            cursor = conn.cursor()
            cursor.execute("SELECT * FROM  crop_info WHERE crop_id=?",(result,))
            data = cursor.fetchall()
            conn.close()

data_dic={'crop_id':data[0][0],'crop_name':data[0][1],'crop_disese':data[0][2],'crop_symp
tom':data[0][3],'crop_treatment':data[0][4],'crop_reference':data[0][5]}
            json_detail = json.dumps(data_dic)
            return json_detail
    else:
            return "Some Error Occured"


if __name__ == "__main__":
    app.run(host=host, port=port, debug=True)
```

Python libraries used while implementation

Flask==0.12.3
click==6.7
gunicorn==19.7.1
itsdangerous==0.24
Jinja2==2.9.6
requests==2.13.0
Werkzeug==0.12.1
tensorflow
keras
pandas==0.22.0
scipy
scikit-learn

C. Android Application Development Code

Application development Code is written on Android Studio Platform. While doing our project, we have developed two versions of AgroSpy application.

AgroSpy 1.0

This Application is developed for Basic Testing Purpose. It has a very basic UI and main intention while developing this project to test all the functionality of backend.

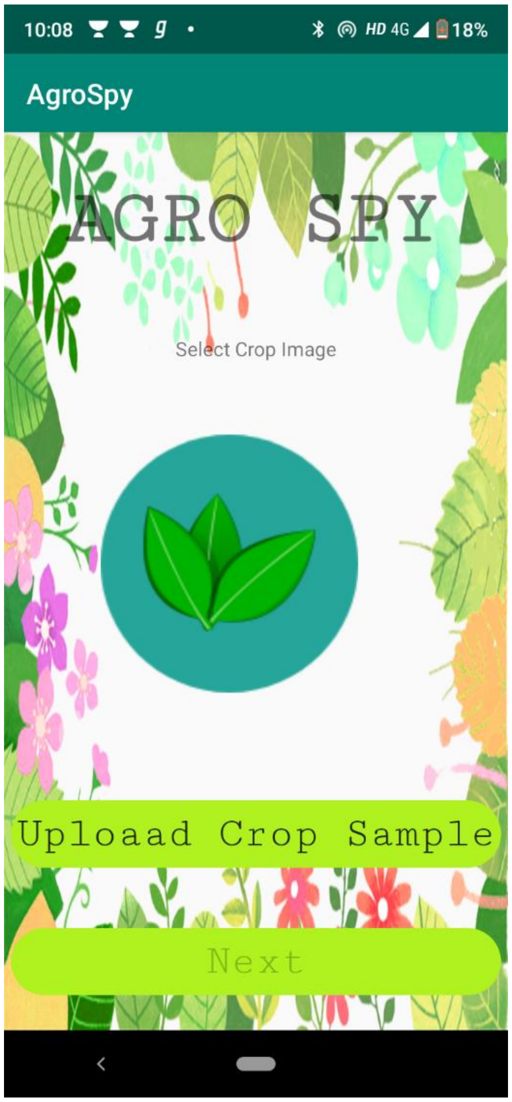


Fig. 4 AgroSpy ver 1.0 Landing Page

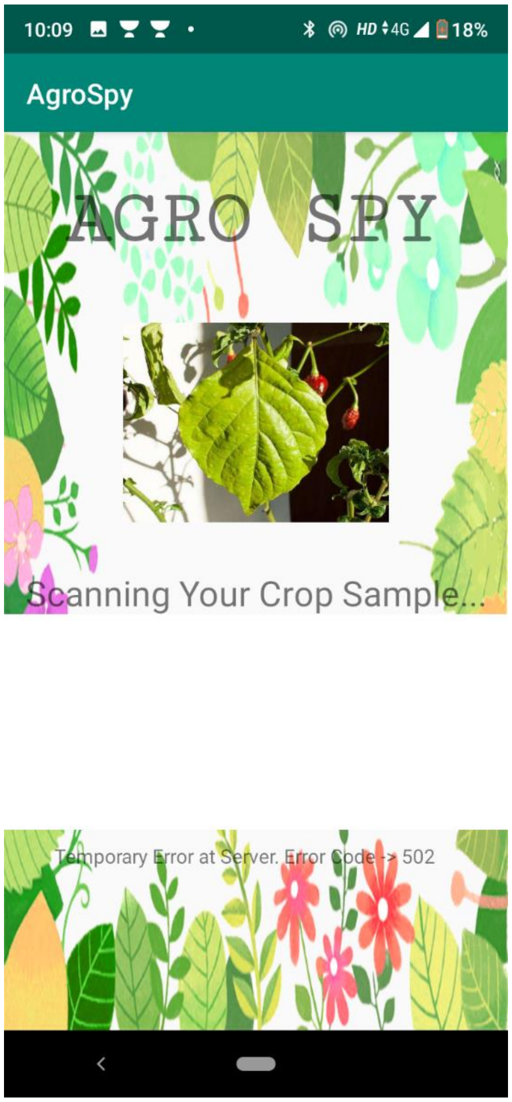

10:09
HD 4G 18%
AgroSpy
AGRO SPY
Scanning Your Crop Sample...
Temporary Error at Server. Error Code -> 502

Fig. 5 AgrSpy ver 1.0 Details Page

Code for Version 1.0

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/hiclipart"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView2"
        style="@android:style/Widget.DeviceDefault.TextView.SpinnerItem"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="serif-monospace"
        android:text="Agro Spy"
        android:textAllCaps="true"
        android:textAppearance="@style/TextAppearance.AppCompat.Display2"
        android:textSize="55dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.463"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.055" />

    <ImageView
        android:id="@+id/cropImage"
        android:layout_width="400dp"
        android:layout_height="184dp"
        android:layout_marginTop="216dp"
        android:background="@mipmap/ic_launcher_foreground"
        android:src="@mipmap/ic_launcher"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:srcCompat="@mipmap/ic_launcher" />

    <Button
        android:id="@+id/uploadButton"
        style="@style/Widget.AppCompat.Button.Borderless.Colored"
        android:layout_width="350dp"
        android:layout_height="wrap_content"
        android:background="@drawable/rounded_button"
        android:ellipsize="marquee"
        android:fontFamily="serif-monospace"
        android:text="Uploaad Crop Sample"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textSize="30sp"
        app:layout_constraintBottom_toTopOf="@+id/uploadButton2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/cropImage"
        app:layout_constraintVertical_bias="0.642" />
    <Button
        android:id="@+id/uploadButton2"
        style="@style/Widget.AppCompat.Button.Borderless.Colored"
        android:layout_width="350dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="143dp"
        android:background="@drawable/rounded_button"
        android:ellipsize="marquee"
        android:enabled="false"
        android:fontFamily="serif-monospace"
        android:text="Next"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textSize="30sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/cropImage" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Crop Image"
        app:layout_constraintBottom_toTopOf="@+id/cropImage"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/hiclipart"
    tools:context=".ScanActivity">

    <TextView
        android:id="@+id/textView22"
        style="@android:style/Widget.DeviceDefault.TextView.SpinnerItem"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="serif-monospace"
        android:text="Agro Spy"
        android:textAllCaps="true"
        android:textAppearance="@style/TextAppearance.AppCompat.Display2"
        android:textSize="55dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.463"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.055" />

    <ImageView
        android:id="@+id/cropScan"
        android:layout_width="190dp"
        android:layout_height="175dp"
        android:layout_marginTop="120dp"
        android:background="@mipmap/ic_launcher_foreground"
        android:src="@mipmap/ic_launcher"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:srcCompat="@mipmap/ic_launcher" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="220dp"
        android:text="Scanning Your Crop Sample..."
        android:textSize="25dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/cropScan"
        app:layout_constraintVertical_bias="0.183" />

    <ScrollView
        android:layout_width="409dp"
        android:layout_height="166dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/cropScan"
        app:layout_constraintVertical_bias="0.232">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

        </LinearLayout>
    </ScrollView>

    <WebView
        android:id="@+id/webresponse"
        android:layout_width="match_parent"
        android:layout_height="154dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/cropScan"
        app:layout_constraintVertical_bias="0.254" >

    </WebView>
    <TextView
        android:id="@+id/response"
        android:layout_width="282dp"
        android:layout_height="127dp"
        android:text="Result"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.459"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/webresponse"
        app:layout_constraintVertical_bias="0.583" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.appdev.shinovi.agrospy;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

class PHOTO {
    public static Bitmap CropSample = null;
}

public class MainActivity extends AppCompatActivity {
    ImageView crop_image;
    Button uploadButton;
    Button nextButton;

    private static final int IMAGE_PICK_CODE =1000;
    private static final int PERMISSION_CODE1 = 1001;
    private static  final int Permission_CODE2 = 1002;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        crop_image =(ImageView)findViewById(R.id.cropImage);
        uploadButton = (Button)findViewById(R.id.uploadButton);
        nextButton = (Button)findViewById(R.id.uploadButton2);
        //Set Button Click Event Rules
        uploadButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Check if Version is greater than or equal to Android M
                if(Build.VERSION.SDK_INT >=Build.VERSION_CODES.M){
                    //Check permission if not granted,request it
                    if(checkSelfPermission(Manifest.permission.READ_EXTERNAL_STORAGE)
== PackageManager.PERMISSION_DENIED){
                        String[] permissions =
{Manifest.permission.READ_EXTERNAL_STORAGE};
                        //open a permission request popup
                        requestPermissions(permissions,PERMISSION_CODE1);
                    }
                    if(checkSelfPermission(Manifest.permission.INTERNET ) ==
PackageManager.PERMISSION_DENIED){
                        String[] permissions = {Manifest.permission.INTERNET};
                        //open a permission request popup
                        requestPermissions(permissions,Permission_CODE2);
                    }
                    else {
                        pickImage();
                    }
                }
                else {
                    pickImage();
                }
            }
        });
        nextButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent i = new Intent(MainActivity.this,ScanActivity.class);
                startActivity(i);
            }
        });
    }

    private void pickImage(){
        Intent intent = new Intent(Intent.ACTION_PICK);
        intent.setType("image/*");
        startActivityForResult(intent,IMAGE_PICK_CODE);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults){
        switch(requestCode){
            case PERMISSION_CODE1:{
                if(grantResults.length >0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED){
                    pickImage();
                }
                else{
                    Toast.makeText(this,"Permission
Denied...",Toast.LENGTH_LONG).show();
                }
            }break;
            case Permission_CODE2:{
                if(grantResults.length >0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED){
                    pickImage();
                }
                else{
                    Toast.makeText(this,"Permission
Denied...",Toast.LENGTH_LONG).show();
                }
            }
        }
    }
    @Override
    protected void onActivityResult(int requestCode,int resultCode,Intent data){
        if(resultCode==RESULT_OK && requestCode ==IMAGE_PICK_CODE ){
            crop_image.setImageURI(data.getData());
            BitmapDrawable drawable = (BitmapDrawable)crop_image.getDrawable();
            Bitmap b = drawable.getBitmap();
            if (b != null) {
                PHOTO.CropSample = b;
            }
            nextButton = (Button)findViewById(R.id.uploadButton2);
            nextButton.setEnabled(true);
        }
    }
}
```

ScanActivity.java

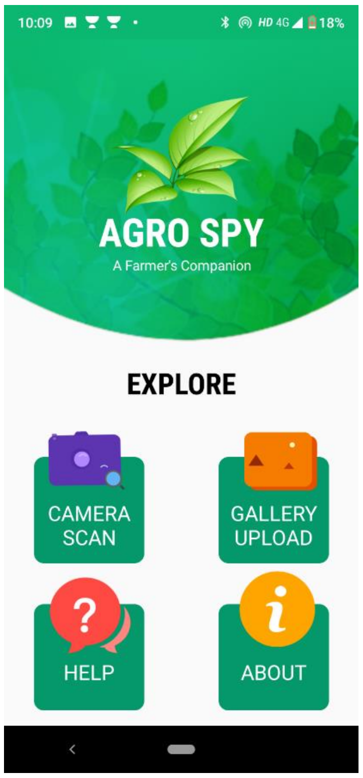
```java
package com.appdev.shinovi.agrospy;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.concurrent.TimeUnit;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class ScanActivity extends AppCompatActivity {

    ImageView cropScan;
    WebView webView;
    int request_count = 2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scan);
        cropScan = (ImageView)findViewById(R.id.cropScan);
        if(PHOTO.CropSample != null){
            cropScan.setImageBitmap(PHOTO.CropSample);
            View v = new View(this);
            connectServer(v);
        }
    }
    void connectServer(View v){

        String postUrl= "http://cropmodel.df.r.appspot.com/uploaded";

        //String postBodyText="Hello";
        //MediaType mediaType = MediaType.parse("text/plain; charset=utf-8");
        //RequestBody postBody = RequestBody.create(mediaType, postBodyText);

        //postRequest(postUrl, postBody);

        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        BitmapFactory.Options options = new BitmapFactory.Options();
        options.inPreferredConfig = Bitmap.Config.RGB_565;
        // Read BitMap by file path
        Bitmap bitmap = PHOTO.CropSample;
        bitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
        byte[] byteArray = stream.toByteArray();
        RequestBody postBodyImage = new MultipartBody.Builder()
                .setType(MultipartBody.FORM)
                .addFormDataPart("image", "cropImage.jpg",
RequestBody.create(MediaType.parse("image/*jpg"), byteArray))
                .build();

        TextView responseText = findViewById(R.id.response);
        responseText.setText("Please wait ...");

        postRequest(postUrl, postBodyImage);
    }

    void postRequest(String postUrl, RequestBody postBody) {

        OkHttpClient client = new OkHttpClient.Builder().readTimeout(30,
TimeUnit.SECONDS).writeTimeout(30, TimeUnit.SECONDS).connectTimeout(30,
TimeUnit.SECONDS).build();

        Request request = new Request.Builder()
                .url(postUrl)
                .post(postBody)
                .build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                // Cancel the post on failure.
                call.cancel();
                e.printStackTrace();
                final String exc = e.toString();
                // In order to access the TextView inside the UI thread, the code is
executed inside runOnUiThread()
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        TextView responseText = findViewById(R.id.response);
                        responseText.setText("Failed to Connect to Server"+exc);
                    }
                });
            }

            @Override
            public void onResponse(Call call, final Response response) throws
IOException {
                // In order to access the TextView inside the UI thread, the code is
executed inside runOnUiThread()
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        if(response.code()==200){
                            TextView responseText = findViewById(R.id.response);
                            try {
                                String res = response.body().string();
                                responseText.setText(res);
                                webView = (WebView)findViewById(R.id.webresponse);
                                WebSettings webSettings = webView.getSettings();
                                webSettings.setJavaScriptEnabled(true);
                                Log.d("msg",res);
                                webView.loadData(res,"text/html","UTF-8");
                            } catch (IOException e) {
                                e.printStackTrace();
                            }
                        }
```

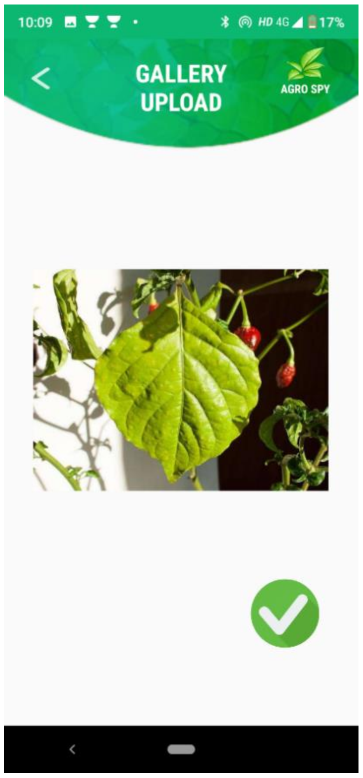
```
                                    else if(response.code()!=200 && request_count>0){
                                        View v = new View(getApplicationContext());
                                        request_count--;
                                        connectServer(v);

                                    }
                                    else{
                                        TextView responseText = findViewById(R.id.response);
                                        String res ="Temporary Error at Server. Error Code -> 
"+response.code();

                                        responseText.setText(res);

                                    }
                                }
                            });
                        }
                    });
                }
            }
```

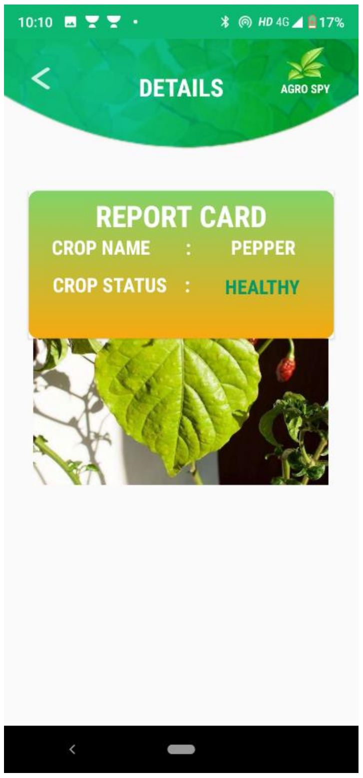
AgroSpy Version 2.0

After checking server workings and mitigating all the challenges we have developed next version of our application i.e. AgroSpy version 2.0.

This is the final version of our app having rich UI.
Camera Scan, Gallery Upload features, Help instructions and Developer Information Working Screenshots of apps has been shown in Figure 6, 7 and 8.







REPORT CARD
CROP NAME : PEPPER
CROP STATUS : HEALTHY

Fig.6 Landing Page **Fig.7 Gallery Upload Page** **Fig.8 Result page**

Implementation Code for Ver 2.0

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <include
        layout="@layout/include"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageView
        android:id="@+id/whiteTile2"
        android:layout_width="47dp"
        android:layout_height="46dp"
        android:elevation="-1dp"
        app:layout_constraintBottom_toBottomOf="@+id/aboutIcon"
        app:layout_constraintEnd_toEndOf="@+id/aboutIcon"
        app:layout_constraintHorizontal_bias="0.512"
        app:layout_constraintStart_toStartOf="@+id/aboutIcon"
        app:layout_constraintTop_toTopOf="@+id/aboutIcon"
        app:layout_constraintVertical_bias="0.548"
        app:srcCompat="@android:color/background_light" />

    <TextView
        android:id="@+id/menubox"
        android:layout_width="112dp"
        android:layout_height="108dp"
        android:background="@drawable/landing_background"
        android:gravity="center"
        android:paddingHorizontal="10dp"
        android:paddingTop="35dp"
        android:paddingBottom="5dp"
        android:text="CAMERA\nSCAN"
        android:textSize="21dp"
        android:elevation="-3dp"
        android:textColor="#FFFFFF"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.123"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.72" />

    <TextView
        android:id="@+id/menubox2"
        android:layout_width="112dp"
        android:layout_height="108dp"
        android:background="@drawable/landing_background"
        android:gravity="center"
        android:paddingHorizontal="10dp"
        android:paddingTop="35dp"
        android:paddingBottom="5dp"
        android:text="GALLERY\nUPLOAD"
        android:textColor="#FFFFFF"
        android:elevation="-3dp"
        android:textSize="21dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.879"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.72" />

    <TextView
        android:id="@+id/menubox3"
        android:layout_width="112dp"
        android:layout_height="108dp"
        android:background="@drawable/landing_background"
        android:gravity="center"
        android:textSize="21dp"
        android:paddingHorizontal="10dp"
        android:paddingTop="35dp"
        android:paddingBottom="5dp"
        android:text="ABOUT"
        android:textColor="#FFFFFF"
        android:elevation="-3dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.879"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.974" />


    <TextView
        android:id="@+id/subtitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="A Farmer's Companion"
        android:textColor="#FFFFFF"
        android:elevation="1dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.503"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.324" />


    <TextView
        android:id="@+id/menubox1"
        android:layout_width="112dp"
        android:layout_height="108dp"
        android:textSize="21dp"
        android:background="@drawable/landing_background"
        android:gravity="center"
        android:paddingHorizontal="10dp"
        android:paddingTop="35dp"
        android:paddingBottom="5dp"
        android:text="HELP"
        android:textColor="#FFFFFF"
        android:elevation="-3dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.123"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.974" />

    <ImageView
        android:id="@+id/cameraIcon"
        android:layout_width="78dp"
        android:layout_height="60dp"
        android:elevation="-1dp"
        app:layout_constraintBottom_toBottomOf="@+id/menubox"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.159"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.843"
        app:srcCompat="@drawable/camera_identification" />

    <ImageView
        android:id="@+id/galleryIcon"
        android:layout_width="78dp"
        android:layout_height="59dp"
        android:elevation="-1dp"
        app:layout_constraintBottom_toBottomOf="@+id/menubox2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.858"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.843"
        app:srcCompat="@drawable/gallery" />

    <ImageView
        android:id="@+id/helpIcon"
        android:layout_width="88dp"
        android:layout_height="77dp"
        android:elevation="-1dp"
        app:layout_constraintBottom_toBottomOf="@+id/menubox1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.16"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.904"
        app:srcCompat="@drawable/help" />

    <ImageView
        android:id="@+id/aboutIcon"
        android:layout_width="86dp"
        android:layout_height="77dp"
        android:elevation="-1dp"
        app:layout_constraintBottom_toBottomOf="@+id/menubox3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.855"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.893"
        app:srcCompat="@drawable/about_512" />

    <ImageView
        android:id="@+id/whiteTile1"
        android:layout_width="47dp"
        android:layout_height="46dp"
        android:elevation="-2dp"
        app:layout_constraintBottom_toBottomOf="@+id/helpIcon"
        app:layout_constraintEnd_toEndOf="@+id/helpIcon"
        app:layout_constraintHorizontal_bias="0.341"
        app:layout_constraintStart_toStartOf="@+id/helpIcon"
        app:layout_constraintTop_toTopOf="@+id/helpIcon"
        app:layout_constraintVertical_bias="0.322"
        app:srcCompat="@android:color/background_light" />


    <TextureView
        android:id="@+id/cameraFrame"
        android:layout_width="300dp"
        android:layout_height="400dp"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="@+id/mainTile"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.493"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.492" />

    <ImageView
        android:id="@+id/cropImage"
        android:layout_width="300dp"
        android:layout_height="400dp"
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="@+id/mainTile"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.493"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.492" />

    <Button
        android:id="@+id/capture_button"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:background="@drawable/camera_shutter"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="@+id/mainTile"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.875" />

    <Button
        android:id="@+id/checkButton"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:background="@drawable/green_check_button"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="@+id/mainTile"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.863"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.874" />

    <ScrollView
        android:id="@+id/scroller"
        android:layout_width="357dp"
        android:layout_height="594dp"
        android:layout_marginStart="16dp"
        android:elevation="-3dp"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="500dp"
            android:elevation="-2dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/helpBox"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="STEPS TO USE APP"
                android:textColor="@color/colorBlack"

                android:visibility="visible"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintVertical_bias="0.002" />

            <TextView
                android:id="@+id/helpSubbox"
                android:layout_width="359dp"
                android:layout_height="669dp"
                android:layout_margin="20dp"
                android:lineHeight="25dp"
                android:text="AgroSpy is an AI project that helps farmer by monitoring
their crops and check for the health of their crops.\nTo Use it follow these
steps:\n\nA.Camera Scan\n1.Click on CAMERA SCAN menu.\n2.Move your phone camera closer
towards the leaf of crop and focus on the infected part.\n3.Click on the Capture
Button given below to take picture.\n4.Click on Scan Button to send image sample for
scanning.\n5.Scanned Details of Your Crop will be shown to you.\n\nB.Gallery
Upload\n1.You can also use images from gallery for health scanning.Select GALLERY
UPLOAD menu.\n2.Your Gallery will be opened,select the picture of the crop.\n3.After
selecting image click on scan button to send the image for scanning.\n4.You will get
the details in next page."
                android:textColor="@color/colorBlack"
                android:textSize="18dp"
                android:visibility="visible"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintHorizontal_bias="0.493"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintVertical_bias="1.0" />
        </androidx.constraintlayout.widget.ConstraintLayout>
    </ScrollView>

    <ScrollView
        android:id="@+id/scrollView2"
        android:layout_width="348dp"
        android:layout_height="310dp"
        android:layout_marginTop="52dp"
        android:visibility="gone"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="500dp"
            android:elevation="-2dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/aboutBox"
                android:layout_width="359dp"
                android:layout_height="669dp"
                android:layout_margin="20dp"
                android:lineHeight="25dp"
                android:text="AgroSpy\nVersion 2.0\nDesigned and Developed By
Shinovi\nFor suggestion and bug report contact agr.shivank@gmail.com"
                android:textColor="@color/colorBlack"
                android:textSize="18dp"
                android:textAlignment="center"
                android:visibility="invisible"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintHorizontal_bias="0.493"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent"
                app:layout_constraintVertical_bias="1.0" />
        </androidx.constraintlayout.widget.ConstraintLayout>
    </ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <include
        android:id="@+id/include"
        layout="@layout/include"
        app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent" />

    <ScrollView
        android:id="@+id/scrollView4"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:elevation="-1dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
tools:context=".DetailsActivity">


<ImageView
    android:id="@+id/cropImage"
    android:layout_width="300dp"
    android:layout_height="400dp"
    android:layout_marginTop="140dp"
    android:gravity="center"
    android:visibility="visible"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.495"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<ImageView
    android:id="@+id/scanLineOrange"
    android:layout_width="300dp"
    android:layout_height="3dp"
    android:layout_marginTop="5dp"
    android:background="@color/colorOrange"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="@+id/cropImage"
    app:layout_constraintEnd_toEndOf="@+id/cropImage"
    app:layout_constraintHorizontal_bias="0.342"
    app:layout_constraintStart_toStartOf="@+id/cropImage"
    app:layout_constraintTop_toTopOf="@+id/cropImage"
    app:layout_constraintVertical_bias="0.16" />

<ImageView
    android:id="@+id/scanLineBlack"
    android:layout_width="300dp"
    android:layout_height="3dp"
    android:background="@color/colorBlack"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="@+id/cropImage"
    app:layout_constraintEnd_toEndOf="@+id/cropImage"
    app:layout_constraintHorizontal_bias="0.342"
    app:layout_constraintStart_toStartOf="@+id/cropImage"
    app:layout_constraintTop_toTopOf="@+id/cropImage"
    app:layout_constraintVertical_bias="0.14" />

<ImageView
    android:id="@+id/scanLineSkyblue"
    android:layout_width="300dp"
    android:layout_height="3dp"
    android:background="@color/colorSkyblue"
    android:visibility="visible"
    app:layout_constraintBottom_toBottomOf="@+id/cropImage"
    app:layout_constraintEnd_toEndOf="@+id/cropImage"
    app:layout_constraintHorizontal_bias="0.342"
    app:layout_constraintStart_toStartOf="@+id/cropImage"
    app:layout_constraintTop_toTopOf="@+id/cropImage"
    app:layout_constraintVertical_bias="0.12" />

<androidx.cardview.widget.CardView
    android:id="@+id/reportCard"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_margin="25dp"
    android:background="@drawable/card_background"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.576"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.380">

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/card_background"
    tools:layout_editor_absoluteX="58dp"
    tools:layout_editor_absoluteY="566dp">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-condensed"
        android:text="REPORT CARD"
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="@color/colorWhite"
        android:textSize="30dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.047" />


    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:fontFamily="sans-serif-condensed"
        android:text="CROP NAME        :"
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="@color/colorWhite"
        android:textSize="20dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.118"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.341" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:layout_marginTop="13dp"
        android:fontFamily="sans-serif-condensed"
        android:text="CROP Status    :"
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="@color/colorWhite"
        android:textSize="20dp"
        android:textStyle="bold"
```

```xml
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.123"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView4"
        app:layout_constraintVertical_bias="0.142" />

    <TextView
        android:id="@+id/cropDiseaseTag"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:fontFamily="sans-serif-condensed"
        android:text="CROP DISEASE    :"
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="@color/colorWhite"
        android:textSize="20dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.13"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView5"
        app:layout_constraintVertical_bias="0.347" />

    <TextView
        android:id="@+id/cropDisease"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:fontFamily="sans-serif-condensed"
        android:text="CROP Disease"
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="@color/colorWhite"
        android:textSize="20dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.478"
        app:layout_constraintStart_toEndOf="@+id/cropDiseaseTag"
        app:layout_constraintTop_toBottomOf="@+id/cropStatus"
        app:layout_constraintVertical_bias="1.0" />

    <TextView
        android:id="@+id/cropStatus"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:fontFamily="sans-serif-condensed"
        android:text="CROP Status"
        android:textAlignment="center"
        android:textAllCaps="true"
        android:textColor="@color/colorWhite"
        android:textSize="20dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.486"
        app:layout_constraintStart_toEndOf="@+id/textView5"
        app:layout_constraintTop_toBottomOf="@+id/cropName"
        app:layout_constraintVertical_bias="0.192" />
        <TextView
            android:id="@+id/cropName"
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:layout_margin="5dp"
            android:fontFamily="sans-serif-condensed"
            android:text="CROP Name"
            android:textAlignment="center"
            android:textAllCaps="true"
            android:textColor="@color/colorWhite"
            android:textSize="20dp"
            android:textStyle="bold"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.475"
            app:layout_constraintStart_toEndOf="@+id/textView4"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.341" />

    </androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/symptomCard"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/reportCard"
    app:layout_constraintVertical_bias="0.0">
    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/card_background"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/reportCard">

        <TextView
            android:id="@+id/textBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="15dp"
            android:fontFamily="sans-serif-condensed"
            android:text="SYMPTOMS"
            android:textAlignment="center"
            android:textAllCaps="true"
            android:textColor="@color/colorWhite"
            android:textSize="15dp"
            android:textStyle="bold"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/titleBox"
            app:layout_constraintVertical_bias="0.0" />

        <TextView
```

```
            android:id="@+id/titleBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="15dp"
            android:layout_marginTop="20dp"
            android:fontFamily="sans-serif-condensed"
            android:text="SYMPTOMS"
            android:textAlignment="center"
            android:textAllCaps="true"
            android:textColor="@color/colorWhite"
            android:textSize="30dp"
            android:textStyle="bold"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.501"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.0" />
    </androidx.constraintlayout.widget.ConstraintLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id = "@+id/treatmentCard"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    app:layout_constraintTop_toBottomOf="@+id/symptomCard"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    tools:layout_editor_absoluteX="20dp">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/card_background">

        <TextView
            android:id="@+id/treatmentBox"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="15dp"
            android:fontFamily="sans-serif-condensed"
            android:text="treatment"
            android:textAlignment="center"
            android:textAllCaps="true"
            android:textColor="@color/colorWhite"
            android:textSize="15dp"
            android:textStyle="bold"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.533"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/treatmentTitle"
            app:layout_constraintVertical_bias="0.0" />

        <TextView
            android:id="@+id/treatmentTitle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="15dp"
            android:fontFamily="sans-serif-condensed"
```

```
                        android:text="TREATMENT"
                        android:textAlignment="center"
                        android:textAllCaps="true"
                        android:textColor="@color/colorWhite"
                        android:textSize="30dp"
                        android:textStyle="bold"
                        app:layout_constraintBottom_toBottomOf="parent"
                        app:layout_constraintEnd_toEndOf="parent"
                        app:layout_constraintHorizontal_bias="0.501"
                        app:layout_constraintStart_toStartOf="parent"
                        app:layout_constraintTop_toTopOf="parent"
                        app:layout_constraintVertical_bias="0.0" />
                </androidx.constraintlayout.widget.ConstraintLayout>
            </androidx.cardview.widget.CardView>
            </androidx.constraintlayout.widget.ConstraintLayout>
    </ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```java
package com.appdev.shinovi.agrospyv20;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.camera.core.CameraX;
import androidx.camera.core.ImageAnalysis;
import androidx.camera.core.ImageAnalysisConfig;
import androidx.camera.core.ImageCapture;
import androidx.camera.core.ImageCaptureConfig;
import androidx.camera.core.ImageProxy;
import androidx.camera.core.Preview;
import androidx.camera.core.PreviewConfig;
import androidx.camera.core.AspectRatio;
import androidx.lifecycle.LifecycleOwner;

import android.app.Application;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.Matrix;
import android.graphics.Typeface;
import android.graphics.drawable.BitmapDrawable;
import android.hardware.Camera;
import android.media.Image;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.util.Rational;
import android.util.Size;
import android.view.Surface;
import android.view.TextureView;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.view.animation.DecelerateInterpolator;
import android.view.animation.Interpolator;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import org.w3c.dom.Text;

import java.io.File;
import java.util.concurrent.Executor;

class PICTURE{
    public static Bitmap CropSample = null;
}
public class MainActivity extends AppCompatActivity {

    ImageView
mainTile,agroIcon,backButton,cameraIcon,galleryIcon,helpIcon,aboutIcon,whiteTile1,whit
eTile2,cropImage;
    TextView
title,subtitle,explore,menuBox1,menuBox2,menuBox3,menuBox4,helpBox,helpSubBox,aboutBox
;
    Animation
title_item_displace,translate3,title_item_displace_alt,alpha,translate,translate2,alph
a2,alpha3;
    TextureView tv;
    ScrollView scrollView,scrollView2;
    boolean stream_toggle = true;
    int stack_depth =0;
    Button captureButton,checkButton;
    private int PERMISSION_REQUEST_CODE=1908;
    private int IMAGE_PICK_CODE = 321;
    private final String[] PERMISSIONS =
{"android.permission.WRITE_EXTERNAL_STORAGE","android.permission.READ_EXTERNAL_STORAGE
","android.permission.CAMERA","android.permission.INTERNET"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        translate =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.translate);
        alpha = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.alpha);
        translate2 =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.translate2);
        alpha2 = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.alpha2);
        title_item_displace =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.title_item_displace);
        translate3 =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.translate3);
        title_item_displace_alt =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.title_item_displace_alt);
        alpha3 =AnimationUtils.loadAnimation(getApplicationContext(),R.anim.alpha3);

        mainTile =    findViewById(R.id.mainTile);
        agroIcon=     findViewById(R.id.agroIcon);
        backButton =  findViewById(R.id.back_button);
        cameraIcon =  findViewById(R.id.cameraIcon);
        helpIcon =    findViewById(R.id.helpIcon);
        galleryIcon = findViewById(R.id.galleryIcon);
        aboutIcon =   findViewById(R.id.aboutIcon);
        title =       findViewById(R.id.title);
        subtitle =    findViewById(R.id.subtitle);
        explore =     findViewById(R.id.explore);
        whiteTile1 =  findViewById(R.id.whiteTile1);
        whiteTile2 =  findViewById(R.id.whiteTile2);
        menuBox1 =    findViewById(R.id.menubox);
        menuBox2 =    findViewById(R.id.menubox1);
        menuBox3 =    findViewById(R.id.menubox2);
        menuBox4 =    findViewById(R.id.menubox3);
        tv   =        findViewById(R.id.cameraFrame);
        captureButton = findViewById(R.id.capture_button);
        cropImage =     findViewById(R.id.cropImage);
        checkButton = findViewById(R.id.checkButton);
        helpBox = findViewById(R.id.helpBox);
        helpSubBox = findViewById(R.id.helpSubbox);
        scrollView = findViewById(R.id.scroller);
        scrollView2 = findViewById(R.id.scrollView2);
        aboutBox =findViewById(R.id.aboutBox);

        explore.setAnimation(alpha2);
        title.setAnimation(translate2);
        subtitle.setAnimation(alpha);
        mainTile.setAnimation(translate);
        agroIcon.setAnimation(alpha);

        menuBox1.setOnClickListener(new View.OnClickListener() {
            @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
            @Override
            public void onClick(View view) {
                make_transition("CAMERA\nSCAN","forward");
                tv.setVisibility(View.VISIBLE);
                captureButton.setVisibility(View.VISIBLE);

                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    if(allPermissionsGranted()){
                        startCamera(); //start camera if permission has been granted
by user
                    } else{
                        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                            requestPermissions( PERMISSIONS, PERMISSION_REQUEST_CODE);
                        }
                    }
                }
                else{
                    startCamera();
                }
            }
        });
        menuBox2.setOnClickListener(new View.OnClickListener() {
            @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
            @Override
            public void onClick(View view) {
                make_transition("HELP","forward");
                scrollView.setVisibility(View.VISIBLE);
                helpBox.setVisibility(View.VISIBLE);
                helpSubBox.setVisibility(View.VISIBLE);
            }
        });
        menuBox3.setOnClickListener(new View.OnClickListener() {
            @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
            @Override
            public void onClick(View view) {
                make_transition("GALLERY\nUPLOAD","forward");

                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    if(allPermissionsGranted()){
                        scanGallery(); //start camera if permission has been granted
by user
```

```
                } else{
                    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                        requestPermissions( PERMISSIONS, PERMISSION_REQUEST_CODE);
                    }
                }
            }
            else{
                scanGallery();
            }
        }
    });
    menuBox4.setOnClickListener(new View.OnClickListener() {
        @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
        @Override
        public void onClick(View view) {
            make_transition("ABOUT","forward");
            scrollView2.setVisibility(View.VISIBLE);
            aboutBox.setVisibility(View.VISIBLE);
        }
    });
    backButton.setOnClickListener(new View.OnClickListener() {
        @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
        @Override
        public void onClick(View view) {
            tv.setVisibility(View.GONE);
            cropImage.setVisibility(View.GONE);
            captureButton.setVisibility(View.GONE);
            checkButton.setVisibility(View.GONE);
            scrollView.setVisibility(View.INVISIBLE);
            helpBox.setVisibility(View.INVISIBLE);
            helpSubBox.setVisibility(View.INVISIBLE);
            scrollView2.setVisibility(View.GONE);
            aboutBox.setVisibility(View.INVISIBLE);

            CameraX.unbindAll();
            make_transition("EXPLORE","reverse");
        }
    });
    checkButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(MainActivity.this,DetailsActivity.class);
            startActivity(i);
        }
    });

}
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
void make_transition(String head,String type){
    if(type.equals("reverse")){
        title_item_displace.setInterpolator(new ReverseInterpolator());
        title_item_displace_alt.setInterpolator(new ReverseInterpolator());
        translate3.setInterpolator(new ReverseInterpolator());

        explore.setText(head);
        explore.setTextColor(getResources().getColor(R.color.colorBlack));
        explore.setTranslationY(0);
        explore.setElevation(-3);
        explore.setTextSize(30);

        backButton.setVisibility(View.INVISIBLE);

        menuBox1.setVisibility(View.VISIBLE);
        menuBox2.setVisibility(View.VISIBLE);
        menuBox3.setVisibility(View.VISIBLE);
        menuBox4.setVisibility(View.VISIBLE);

        cameraIcon.setVisibility(View.VISIBLE);
        galleryIcon.setVisibility(View.VISIBLE);
        helpIcon.setVisibility(View.VISIBLE);
        aboutIcon.setVisibility(View.VISIBLE);
        whiteTile1.setVisibility(View.VISIBLE);
        whiteTile2.setVisibility(View.VISIBLE);
        subtitle.setVisibility(View.VISIBLE);
        subtitle.startAnimation(alpha);
        stack_depth--;

    }
    else {
        title_item_displace.setInterpolator(new DecelerateInterpolator());
        title_item_displace_alt.setInterpolator(new DecelerateInterpolator());
        translate3.setInterpolator(new DecelerateInterpolator());

        explore.setText(head);
        explore.setTextColor(getResources().getColor(R.color.colorWhite));
        explore.setTranslationY(-600);
        explore.setElevation(1);
        explore.setTextSize(25);

        backButton.setVisibility(View.VISIBLE);

        menuBox1.setVisibility(View.INVISIBLE);
        menuBox2.setVisibility(View.INVISIBLE);
        menuBox3.setVisibility(View.INVISIBLE);
        menuBox4.setVisibility(View.INVISIBLE);

        subtitle.setVisibility(View.GONE);
        subtitle.setAlpha(0);
        cameraIcon.setVisibility(View.INVISIBLE);
        galleryIcon.setVisibility(View.INVISIBLE);
        helpIcon.setVisibility(View.INVISIBLE);
        aboutIcon.setVisibility(View.INVISIBLE);
        whiteTile1.setVisibility(View.INVISIBLE);
        whiteTile2.setVisibility(View.INVISIBLE);
        stack_depth =1;
    }

    title.startAnimation(title_item_displace_alt);
    agroIcon.startAnimation(title_item_displace);
    mainTile.startAnimation(translate3);

}
@Override
public  void onBackPressed(){
    if (stack_depth==1){
        tv.setVisibility(View.GONE);
        cropImage.setVisibility(View.GONE);
        captureButton.setVisibility(View.GONE);
        checkButton.setVisibility(View.GONE);
        scrollView.setVisibility(View.INVISIBLE);
        helpBox.setVisibility(View.INVISIBLE);
        helpSubBox.setVisibility(View.INVISIBLE);
        scrollView2.setVisibility(View.GONE);
        aboutBox.setVisibility(View.INVISIBLE);

        CameraX.unbindAll();
```

```
            make_transition("EXPLORE","reverse");
        }
        else if(stack_depth == 0){
            Toast.makeText(this, "Press back again to exit.",
Toast.LENGTH_SHORT).show();
            stack_depth--;
        }
        else {
            super.onBackPressed();

        }
    }
    private void startCamera() {
        //make sure there isn't another camera instance running before starting
        CameraX.unbindAll();
        Toast.makeText(MainActivity.this,"Entered into start
Camera",Toast.LENGTH_LONG).show();
        /* start preview */
        int aspRatioW = tv.getWidth(); //get width of screen
        int aspRatioH = tv.getHeight(); //get height
        AspectRatio asp2 = AspectRatio.RATIO_16_9;
        //Rational asp = new Rational(aspRatioW, aspRatioH); //aspect ratio
        Size screen = new Size(aspRatioW, aspRatioH); //size of the screen

        //config obj for preview/viewfinder thingy.
        PreviewConfig pConfig = new
PreviewConfig.Builder().setTargetResolution(screen).build();
        Preview preview = new Preview(pConfig); //lets build it

        preview.setOnPreviewOutputUpdateListener(
                new Preview.OnPreviewOutputUpdateListener() {
                    //to update the surface texture we have to destroy it first, then
re-add it
                    @Override
                    public void onUpdated(Preview.PreviewOutput output){
                        ViewGroup parent = (ViewGroup) tv.getParent();
                        parent.removeView(tv);
                        parent.addView(tv, 0);

                        tv.setSurfaceTexture(output.getSurfaceTexture());
                        updateTransform();
                    }
                });

        /* image capture */

        //config obj, selected capture mode
        ImageCaptureConfig imgCapConfig = new
ImageCaptureConfig.Builder().setCaptureMode(ImageCapture.CaptureMode.MIN_LATENCY)

.setTargetRotation(getWindowManager().getDefaultDisplay().getRotation()).build();
        final ImageCapture imgCap = new ImageCapture(imgCapConfig);

        captureButton = findViewById(R.id.capture_button);
        captureButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (stream_toggle==true){
                    File folder = new File(Environment.getExternalStorageDirectory() +
                            File.separator + "AgroSpy/Samples");
                    boolean success = true;
                    if (!folder.exists()) {
                        success = folder.mkdirs();
                    }
                    if (success) {
                        // Do something on success
                    } else {
                        Toast.makeText(getBaseContext(),"Unable to Create Folder
AgroSpy.",Toast.LENGTH_LONG).show();
                    }
                    File file = new File(Environment.getExternalStorageDirectory() +
"/AgroSpy/Samples/" + System.currentTimeMillis() + ".jpg");
                    Executor executor = new Executor() {
                        @Override
                        public void execute(Runnable runnable) {

                        }
                    };
                    imgCap.takePicture(file,executor  ,new
ImageCapture.OnImageSavedListener() {
                        @Override
                        public void onImageSaved(@NonNull File file) {
                            String msg = "Photo capture succeeded: " +
file.getAbsolutePath();
                            Toast.makeText(MainActivity.this,
msg,Toast.LENGTH_LONG).show();

                        }

                        @Override
                        public void onError(@NonNull ImageCapture.ImageCaptureError
useCaseError, @NonNull String message, @Nullable Throwable cause) {
                            String msg = "Photo capture failed: " + message;
                            Toast.makeText(getApplicationContext(),
msg,Toast.LENGTH_LONG).show();
                            if(cause != null){
                                cause.printStackTrace();
                            }
                        }
                    });
                    CameraX.unbindAll();
                    Bitmap cropBmpImage = tv.getBitmap();
                    cropImage.setVisibility(View.VISIBLE);
                    tv.setVisibility(View.INVISIBLE);
                    cropImage.setImageBitmap(cropBmpImage);
                    PICTURE.CropSample = cropBmpImage;

captureButton.setBackground(getResources().getDrawable(R.drawable.restart_button));
                    stream_toggle = !(stream_toggle);
                    checkButton.setVisibility(View.VISIBLE);

                }
                else{

captureButton.setBackground(getResources().getDrawable(R.drawable.camera_shutter));
                    stream_toggle = !(stream_toggle);
                    startCamera();
                    tv.setVisibility(View.VISIBLE);
                    cropImage.setVisibility(View.INVISIBLE);
                    checkButton.setVisibility(View.INVISIBLE);

                }

            }
        });
        /* image analyser */

        ImageAnalysisConfig imgAConfig = new
ImageAnalysisConfig.Builder().setImageReaderMode(ImageAnalysis.ImageReaderMode.ACQUIRE
_LATEST_IMAGE).build();
        ImageAnalysis analysis = new ImageAnalysis(imgAConfig);
        Executor ex = new Executor() {
            @Override
            public void execute(Runnable runnable) {

            }
        };
        analysis.setAnalyzer(ex,
                new ImageAnalysis.Analyzer(){
                    @Override
                    public void analyze(ImageProxy image, int rotationDegrees){
                        //Code for analyzation
                    }
                });

        //bind to lifecycle:
        CameraX.bindToLifecycle((LifecycleOwner)this, analysis, imgCap, preview);
    }

    private void scanGallery(){
        Intent intent = new Intent(Intent.ACTION_PICK);
        intent.setType("image/*");
        startActivityForResult(intent,IMAGE_PICK_CODE);
    }
    private void updateTransform(){
        /*
         * compensates the changes in orientation for the viewfinder, bc the rest of
the layout stays in portrait mode.
         * methinks :thonk:
         * imgCap does this already, this class can be commented out or be used to
optimise the preview
         */
        Matrix mx = new Matrix();
        float w = tv.getMeasuredWidth();
        float h = tv.getMeasuredHeight();

        float centreX = w / 2f; //calc centre of the viewfinder
        float centreY = h / 2f;

        int rotationDgr;
        int rotation = (int)tv.getRotation(); //cast to int bc switches don't like
floats

        switch(rotation){ //correct output to account for display rotation
            case Surface.ROTATION_0:
                rotationDgr = 0;
                break;
            case Surface.ROTATION_90:
                rotationDgr = 90;
                break;
            case Surface.ROTATION_180:
                rotationDgr = 180;
                break;
            case Surface.ROTATION_270:
                rotationDgr = 270;
                break;
            default:
                return;
        }

        mx.postRotate((float)rotationDgr, centreX, centreY);
        tv.setTransform(mx); //apply transformations to textureview
    }

    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        //start camera when permissions have been granted otherwise exit app
        if(requestCode == PERMISSION_REQUEST_CODE){
            if(allPermissionsGranted()){
                startCamera();
            } else{
                Toast.makeText(this, "Permissions not granted by the user.",
Toast.LENGTH_SHORT).show();
                finish();
            }
        }
    }
    @RequiresApi(api = Build.VERSION_CODES.M)
    private boolean allPermissionsGranted(){
        //check if req permissions have been granted
        for(String permission : PERMISSIONS){
            if(checkSelfPermission(permission) != PackageManager.PERMISSION_GRANTED){
                return false;
            }
        }
        return true;
    }

    protected void onActivityResult(int requestCode,int resultCode,Intent data){
        if(resultCode==RESULT_OK && requestCode == IMAGE_PICK_CODE ){
            cropImage.setImageURI(data.getData());
            BitmapDrawable drawable = (BitmapDrawable)cropImage.getDrawable();
            Bitmap b = drawable.getBitmap();
            if (b != null) {
                PICTURE.CropSample = b;
            }
            cropImage.setVisibility(View.VISIBLE);
            checkButton.setVisibility(View.VISIBLE);

        }
    }
}
```

DetailsActivity.java

```
package com.appdev.shinovi.agrospyv20;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.ImageView;
import android.widget.TextView;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.concurrent.TimeUnit;

import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

import static android.view.View.GONE;

public class DetailsActivity extends AppCompatActivity {

    ImageView
mainTile,agroIcon,backButton,cropImage,scanLineBlack,scanLineOrange,scanLineSkyblue;
    TextView
title,explore,cropName,cropDisease,cropStatus,cropDiseaseTag,symptomBox,treatmentBox;
    Animation
title_item_displace,translate3,title_item_displace_alt,alpha,translate,translate2,alph
a2,alpha3,scanLineTranslate;
    int request_count =1;
    CardView reportCard,symptomCard,treatmentCard;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_details);
        mainTile = findViewById(R.id.mainTile);
        agroIcon = findViewById(R.id.agroIcon);
        backButton = findViewById(R.id.back_button);
        cropImage = findViewById(R.id.cropImage);
        title = findViewById(R.id.title);
        explore = findViewById(R.id.explore);
        symptomBox = findViewById(R.id.textBox);
        scanLineBlack = findViewById(R.id.scanLineBlack);
        scanLineOrange = findViewById(R.id.scanLineOrange);
        scanLineSkyblue = findViewById(R.id.scanLineSkyblue);
        reportCard = findViewById(R.id.reportCard);
        treatmentBox = findViewById(R.id.treatmentBox);
        cropName = findViewById(R.id.cropName);
        cropStatus = findViewById(R.id.cropStatus);
        cropDisease = findViewById(R.id.cropDisease);
        cropDiseaseTag = findViewById(R.id.cropDiseaseTag);
        symptomCard = findViewById(R.id.symptomCard);
        treatmentCard = findViewById(R.id.treatmentCard);
        backButton = findViewById(R.id.back_button);
        alpha = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.alpha);
        alpha2 = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.alpha2);
        alpha3 = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.alpha3);
        translate =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.translate);
        translate2 =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.translate2);
        translate3 =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.translate3);
        title_item_displace =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.title_item_displace);
        title_item_displace_alt =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.title_item_displace_alt);
        scanLineTranslate =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.scan_translate);
        alpha.setInterpolator(new ForwardInterpolator());
        alpha2.setInterpolator(new ForwardInterpolator());
        alpha3.setInterpolator(new ForwardInterpolator());
        translate.setInterpolator(new ForwardInterpolator());
        translate2.setInterpolator(new ForwardInterpolator());
        translate3.setInterpolator(new ForwardInterpolator());
        title_item_displace.setInterpolator(new ForwardInterpolator());
        title_item_displace_alt.setInterpolator(new ForwardInterpolator());
        reportCard.setVisibility(View.GONE);
        symptomCard.setVisibility(GONE);
        treatmentCard.setVisibility(GONE);
        mainTile.startAnimation(translate);
        mainTile.startAnimation(translate3);
        explore.startAnimation(alpha2);
        title.startAnimation(translate2);

        agroIcon.startAnimation(alpha);
        title.startAnimation(title_item_displace_alt);
        agroIcon.startAnimation(title_item_displace);

        explore.setText("DETAILS");
        explore.setTextColor(getResources().getColor(R.color.colorWhite));
        explore.setTranslationY(-600);
        explore.setElevation(1);
        explore.setTextSize(25);



        backButton.setVisibility(View.VISIBLE);
        cropImage.setVisibility(View.VISIBLE);
        backButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                DetailsActivity.super.onBackPressed();
            }
        });
        cropImage.setImageBitmap(PICTURE.CropSample);
        scanLineBlack.startAnimation(scanLineTranslate);
        scanLineOrange.startAnimation(scanLineTranslate);
        scanLineSkyblue.startAnimation(scanLineTranslate);
        View v =new View(this);
        connectServer(v);

    }
    void connectServer(View v){

        String postUrl= "http://agrospy.df.r.appspot.com/uploaded";
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        BitmapFactory.Options options = new BitmapFactory.Options();
        options.inPreferredConfig = Bitmap.Config.RGB_565;
        // Read Bitmap
        Bitmap bitmap = PICTURE.CropSample;
        bitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
        byte[] byteArray = stream.toByteArray();

        RequestBody postBodyImage = new MultipartBody.Builder()
                .setType(MultipartBody.FORM)
                .addFormDataPart("image", "cropImage.jpg",
RequestBody.create(MediaType.parse("image/*jpg"), byteArray))
                .build();


        postRequest(postUrl, postBodyImage);
    }

    void postRequest(String postUrl, RequestBody postBody) {

        OkHttpClient client = new OkHttpClient.Builder().readTimeout(30,
TimeUnit.SECONDS).writeTimeout(30, TimeUnit.SECONDS).connectTimeout(30,
TimeUnit.SECONDS).build();

        Request request = new Request.Builder()
                .url(postUrl)
                .post(postBody)
                .build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                // Cancel the post on failure.
                call.cancel();
                e.printStackTrace();
                final String exc = e.toString();
                // In order to access the TextView inside the UI thread, the code is
executed inside runOnUiThread()
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        cropName.setText("Failed to Connect to Server "+exc);
                    }
                });
            }

            @Override
            public void onResponse(Call call, final Response response) throws
IOException {
                // In order to access the TextView inside the UI thread, the code is
executed inside runOnUiThread()
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        if(response.code()==200){
                            String
crop_name="",crop_disease="",crop_symptom="",crop_treatment="",crop_reference="";
                            int crop_id =0;
                            try {
                                String res = response.body().string();
                                try {
                                    JSONObject result = new JSONObject(res);
                                    crop_id =result.getInt("crop_id");
                                    crop_name = result.getString("crop_name");
                                    crop_disease = result.getString("crop_disease");
                                    crop_symptom = result.getString("crop_symptom");
                                    crop_treatment =
result.getString("crop_treatment");
                                    crop_reference =
result.getString("crop_reference");
                                } catch (JSONException e) {
                                    e.printStackTrace();
                                }
                                reportCard.setVisibility(View.VISIBLE);
                                //reportCard.startAnimation(alpha);
                                cropName.setText(crop_name);
                                if(crop_id !=48){
                                    if(crop_disease.equals("Healthy")){
                                        cropStatus.setText(crop_disease);

cropStatus.setTextColor(getResources().getColor(R.color.colorBg));
                                        cropDisease.setVisibility(GONE);
                                        cropDiseaseTag.setVisibility(GONE);
                                    }
                                    else{
                                        cropDisease.setVisibility(View.VISIBLE);
                                        cropDiseaseTag.setVisibility(View.VISIBLE);
                                        cropDisease.setText(crop_disease);
                                        cropStatus.setText("Unhealthy");
                                        symptomBox.setText(crop_symptom);
                                        treatmentBox.setText(crop_treatment);
                                        symptomCard.setVisibility(View.VISIBLE);
                                        treatmentCard.setVisibility(View.VISIBLE);
                                    }
                                }
                                else{

                                    cropDisease.setVisibility(GONE);
                                    cropDiseaseTag.setVisibility(GONE);
                                    cropStatus.setText("No Crop Image Found");

                                }

                                scanLineBlack.clearAnimation();
                                scanLineOrange.clearAnimation();
                                scanLineSkyblue.clearAnimation();
                                scanLineBlack.setVisibility(View.INVISIBLE);
                                scanLineOrange.setVisibility(View.INVISIBLE);
                                scanLineSkyblue.setVisibility(View.INVISIBLE);

                            }
                            catch (IOException e) {
                                e.printStackTrace();
                            }
                        }

                        else if(response.code()!=200 && request_count>0){
                            View v = new View(getApplicationContext());
                            request_count--;
                            connectServer(v);

                        }
                        else{

                            String res ="Temporary Error at Server. Error Code -> 
"+response.code();

                            cropName.setText(res);

                        }
                    }
                });
            }
        });
    }
}
```

Chapter 4: Testing and Deployment

A. Testing

We have secured 20% of the dataset for testing purpose. We tested our dataset on every model we trained, we have recorded 89% of the validation accuracy in the final epoch on testing datasets on model based on AlexNet

The Table given here will describes training and testing accuracy on various epochs :

```
Epoch 1/20
500/500 [==============================] - 4863s 10s/step - loss: 2.6492 -
acc: 0.2909 - val_loss: 1.7570 - val_acc: 0.4876
Epoch 2/20
500/500 [==============================] - 4565s 9s/step - loss: 1.3160 -
acc: 0.5934 - val_loss: 1.0639 - val_acc: 0.6657
Epoch 3/20
500/500 [==============================] - 4483s 9s/step - loss: 0.7997 -
acc: 0.7412 - val_loss: 0.7213 - val_acc: 0.7730
Epoch 4/20
500/500 [==============================] - 4465s 9s/step - loss: 0.5668 -
acc: 0.8107 - val_loss: 0.5800 - val_acc: 0.8161
Epoch 5/20
500/500 [==============================] - 4494s 9s/step - loss: 0.4573 -
acc: 0.8470 - val_loss: 0.5171 - val_acc: 0.8376
Epoch 6/20
500/500 [==============================] - 4504s 9s/step - loss: 0.3788 -
acc: 0.8709 - val_loss: 0.4818 - val_acc: 0.8544
Epoch 7/20
500/500 [==============================] - 4469s 9s/step - loss: 0.3384 -
acc: 0.8842 - val_loss: 0.4453 - val_acc: 0.8641
Epoch 8/20
500/500 [==============================] - 4503s 9s/step - loss: 0.3016 -
acc: 0.8963 - val_loss: 0.4283 - val_acc: 0.8690
Epoch 9/20
500/500 [==============================] - 4547s 9s/step - loss: 0.2733 -
acc: 0.9057 - val_loss: 0.4494 - val_acc: 0.8693
Epoch 10/20
500/500 [==============================] - 4590s 9s/step - loss: 0.2445 -
acc: 0.9146 - val_loss: 0.3979 - val_acc: 0.8810
Epoch 11/20
500/500 [==============================] - 4468s 9s/step - loss: 0.2364 -
acc: 0.9174 - val_loss: 0.4511 - val_acc: 0.8678
Epoch 12/20
500/500 [==============================] - 4482s 9s/step - loss: 0.2272 -
acc: 0.9209 - val_loss: 0.4073 - val_acc: 0.8771
Epoch 13/20
500/500 [==============================] - 4489s 9s/step - loss: 0.2020 -
acc: 0.9278 - val_loss: 0.3913 - val_acc: 0.8883
Epoch 14/20
```

```
500/500 [==============================] - 4492s 9s/step - loss: 0.2021 -
acc: 0.9290 - val_loss: 0.3854 - val_acc: 0.8940
Epoch 15/20
500/500 [==============================] - 4475s 9s/step - loss: 0.1896 -
acc: 0.9342 - val_loss: 0.3922 - val_acc: 0.8898
Epoch 16/20
500/500 [==============================] - 4473s 9s/step - loss: 0.1775 -
acc: 0.9373 - val_loss: 0.4143 - val_acc: 0.8924
Epoch 17/20
500/500 [==============================] - 4481s 9s/step - loss: 0.1707 -
acc: 0.9383 - val_loss: 0.4084 - val_acc: 0.8880
Epoch 18/20
500/500 [==============================] - 4548s 9s/step - loss: 0.1725 -
acc: 0.9397 - val_loss: 0.3811 - val_acc: 0.8935
Epoch 19/20
500/500 [==============================] - 4640s 9s/step - loss: 0.1597 -
acc: 0.9429 - val_loss: 0.4248 - val_acc: 0.8892
Epoch 20/20
500/500 [==============================] - 4684s 9s/step - loss: 0.1565 -
acc: 0.9440 - val_loss: 0.4109 - val_acc: 0.8945
```

Table 5. Testing and Training Accuracy in several Epochs

B. Deployment

Neural Net Deployment

CNN model is trained on a powerful Google Virtual Instance combined with NVIDIA Tesla T4 GPU for accelerating training process.
Full system specification is given below:

Host Name:	INSTANCE-1
OS Name:	Microsoft Windows Server 2016 Datacenter
OS Version:	10.0.14393 N/A Build 14393
OS Manufacturer:	Microsoft Corporation
OS Configuration:	Standalone Server
OS Build Type:	Multiprocessor Free
Registered Owner:	N/A
Registered Organization:	N/A
Product ID:	00376-40000-00000-AA947
Original Install Date:	4/29/2020, 3:37:46 AM
System Boot Time:	5/20/2020, 9:19:57 AM
System Manufacturer:	Google
System Model:	Google Compute Engine
System Type:	x64-based PC
Processor(s):	1 Processor(s) Installed.

```
                         [01]: Intel64 Family 6 Model 79 Stepping 0 GenuineIntel ~2200 Mhz
BIOS Version:              Google Google, 1/1/2011
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume2
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+00:00) Monrovia, Reykjavik
Total Physical Memory:     15,359 MB
Available Physical Memory: 12,237 MB
Virtual Memory: Max Size:  16,383 MB
Virtual Memory: Available: 13,424 MB
Virtual Memory: In Use:    2,959 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\INSTANCE-1
Hotfix(s):                 7 Hotfix(s) Installed.
                         [01]: KB4537477
                         [02]: KB4049065
                         [03]: KB4486129
                         [04]: KB4494175
                         [05]: KB4524244
                         [06]: KB4550994
                         [07]: KB4550947
Network Card(s):           1 NIC(s) Installed.
                         [01]: Google VirtIO Ethernet Adapter
                               Connection Name: Ethernet
                               DHCP Enabled:    Yes
                               DHCP Server:     169.254.169.254
                               IP address(es)
                               [01]: 10.140.0.2
                               [02]: fe80::a464:96b0:76bb:1134
Hyper-V Requirements:      A hypervisor has been detected. Features required for Hyper-
V will not be displayed.
```

Web Server Deployment

Web server is deployed using Flask Micro framework on python combined with MySQLite3 database interface.
Server is deployed over Google App Engine PaaS Platform. Specification of App Engine is given in the app.yaml document given below:

App.yaml

```
runtime: python
env: flex
entrypoint: gunicorn -b :$PORT main:app

runtime_config:
  python_version: 3

manual_scaling:
  instances: 1
resources:
  cpu: 1
  memory_gb: 1
  disk_size_gb: 10
```

Chapter 6:Future Enhancements

While developing this project we got lots of new ideas that can enhance this project to further more extent, but due to limited time we are suggesting those ideas in Future Enhancements section so that any future work that would be done with this project take consideration from it. Some of the future enhancement ideas are:

A. **Embedded Drone System for Sample Collection**
   In our project, we put limits on sample image collection only through smartphone camera, but with more curated dataset and enhanced IoT technology we can also embed an automated drone camera with the cloud server. This will make the process automated and there will be no need to manually visit to crop field for health scanning. Drone will monitor the crops and inform automatically if any issue occurs.

B. **Automated Artificial Agriculture Growing Field**

   Combining this project resources with other agriculture based utility projects like Crop Weed Detection, Irrigation monitoring using thermal camera, Crop Disease treatment kits and robots, we can create an Artificial Crop Ripening Field where the only work needs to be done is to sow seeds. After that this Artificial Crop Field will take care of the growth and proper health of the crops.

Summary

Working on this project AgroSpy was very exciting and interesting. The main motivation behind this project was its relation with the real world. During development of this project we faced lots of little and big challenges that brought more maturity in our skills of problem solving. We found ourselves working in a more realistic manner.

During the development we also moved around lots of exciting and industry adopted technology stacks. Working over cutting edge technologies like Convolutional Neural Networks, Android

Development, Google Cloud Platforms etc. has given us much insights about their working concepts. We also learnt a lot about agriculture. We studied different crops patterns, crop species and their disease that was really enhanced our knowledge. We also come to know about different organizations working in the field of Agriculture Advancements.

At last, we ended with a real world applicable practical application. This project gave us enough confidence to showcase our engineering skills on further projects in upcoming future.

References

1.A Crop/Weed Field Image Dataset for the Evaluation of Computer Vision Based Precision Agriculture Tasks Sebastian Haug1(B) and J¨orn Ostermann2

2.Using Deep Learning for Image-Based Plant Disease Detection
Sharada P. Mohanty 1, 2, 3, David P. Hughes 4, 5, 6 and Marcel Salathé1, 2, 3

3.Deep Learning with Python
Francois Collet

4.https://www.youtube.com/watch?v=98sLySmptKY&list=PLtCBuHKmdxOdO88sgcFyDdL5AT g1WY-_U

5. https://plantvillage.psu.edu/

6. https://developer.android.com/guide/topics/media/camera

7. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6427818/