QIU
QUEST INTERNATIONAL UNIVERSITY

Bachelor in Computer Science (Hons)

Academic Year 2021-2022 (Year 3 Semester 1/Semester 2)

Study Intake: 202207
ASSIGNMENT

Submission Deadline:
8 October 2022

BCS3225
INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Student's Particulars:

Matric No	:	202004-003855	Intake/Semester :	202004 / Sem 8
Student's Name	:	Chung Yau Kit		
Lecturer's Name	:	Dr Lee Lam Hong		
Due Date	:	8 October 2022	Submission Date :	30 Sept 2022
Total number of pages including this cover page	:	76 pages		

Official Use:

Marker's Comment			
Marks Awarded :	/ 100	Overall Marks (20%) :	

Student declaration:

I declare that:

- *I understand what is meant by plagiarism*
- *The implication of plagiarism has been explained to me by my institution*
- *This assignment is all my own work and I have acknowledged any use of the published and unpublished works of other people.*

Student's signature: Chung Yau Kit Date: 30 September 2022

Marks Sheet:

Marks Table		
Questions	**Marks Allocated**	**Marks Awarded**
Intelligent Agent Program	70	
Technical Report	30	
Total	**100**	

Overall Total : /100

Table of Contents

Table of figures - figure

Table of figures - table

Chapter 1: Introduction

1.1 Introduction

Plant diseases have impacted society and it is defined as anything that prevents the healthy growth of a plant. It is usually caused by pathogens and environmental conditions. In order to know the health of a plant, Leaf Scanner is developed.

Leaf Scanner is the mobile application that able to show the plant health by scan the leaf of the plant. It shows the disease of the plant such as black rot, early bright, curl virus and more. It perceives the environment through camera, or any input images, then display the output through the screen of smartphones.

User can provide a correct and optimal treatment to the plant after knowing the disease that the plant suffering.

1.2 Problem Statement

People difficult to know the plant name and its health status by just looking at it if without any related knowledge and experience. Therefore, they couldn't identify the disease that the plant is suffering. Moreover, without a proper identification of the disease, the disease control measures can be waste of money and time. Eventually, it will lead to further plant losses.

1.3 Objectives

To develop a mobile application that capable to detect and identify the type of disease of various type of plant.

1.4 Scope

Leaf Scanner is a mobile application that use to get the health of a plant by input the image of leaf. It uses TensorFlow library to perform deep learning process which build and train the Convolutional Neuron Network (CNN). Then convert it to a TensorFlow Lite model that able to recognize the image of leaves, so that it can be use and load into the mobile application.

Chapter 2: Literature Review

2.1 Literature on intelligent agent.

2.1.1 Types of intelligent agents

An intelligent agent is a program that can make decisions or perform a service based on its environment, user input and experiences. These programs can be used to autonomously gather information on a regular, programmed schedule or when prompted by the user in real time. It perceives its environment through sensors and acts upon that environment through actuators.

There are four types of basic intelligent agent. First, simple reflex agent. It is simple but with limited intelligence. This is because it will need a fully observable environment to select actions based on the current percept. In other words, it uses condition-action rules.

Second, model-based reflex agent. It maintains some internal state that depends on percept history which can reflects some unobserved aspect of current state. Therefore, it can handle partial observable environments since it required knowledge of model of the world. It is also using condition-action rules.

Third, goal-based agent. It needs goal information that describe situation that are desirable to help in decision making since the current state information is not enough. This agent can alter its behaviour to suit new conditions.

Next, utility-based agent. The goals of this agent will provide binary distinction between happy and unhappy state which helps is decision making. This agent is useful when goal conflict, which means there are several goals that the agent can aim for.

Other than these four basic intelligent agents, there is also an agent called learning agent. It allows agent to operate in initially unknown environments. It takes feedback from whatever actions it has performed and adapting accordingly since this agent can gradually improve and become knowledgeable about the environment over time.

2.1.2 Intelligent agent of Leaf Scanner

The intelligent agent in Leaf Scanner is a combination of learning agent and simple reflex agent. Basically, it perceives the environment through camera, or any input images, then display the output through the screen of smartphones.

For the learning agent, it makes use of machine learning to let the program know what the image is about. The learning element is the program to read and examine the images whereas the performance element is the dataset of images. Critic will be the confidence value of an image and the problem generator will be the accuracy and loss during the training process.

Apart from this, it is also the simple reflex agent is because the model adopted in the Leaf Scanner apps will determine confidence from the trained model whenever an image is input. After that, it will fetch the related information of the highest confidence value then show it to the users.

The PEAS of Leaf Scanner agent are as follows:

Performance Measure	Accuracy of the output, processing time
Environment	Leaves, quality of images
Actuators	Display screen
Sensors	Camera, file manager

2.2 Literature on related technology, development platforms and tools.

In this project, Leaf Scanner, it uses some technologies which is under artificial intelligence to accomplish.

2.2.1 Machine Learning (CNN)

Machine learning plays a big role in this assignment. It is a branch of artificial intelligence which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Convolutional Neural Network (CNN) is a neuron network which under deep learning field and which deep learning is a subset of machine learning.

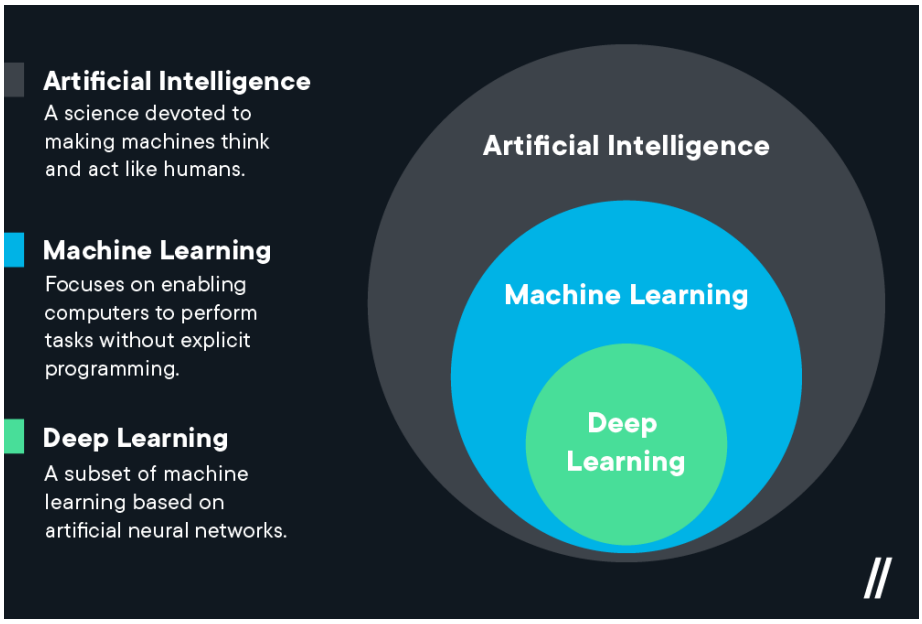


Figure 1: Relationship between Artificial Intelligence, Machine Learning, and Deep Learning

CNN is very helpful in image recognition and process; it has multilayer perceptron that purposely designed for reduce the processing requirements. The layers are consisting of input layer, hidden layer, and output layer. The hidden layer is including pre-processing layer, multiple convolutional layers, pooling layers, and normalization layers.

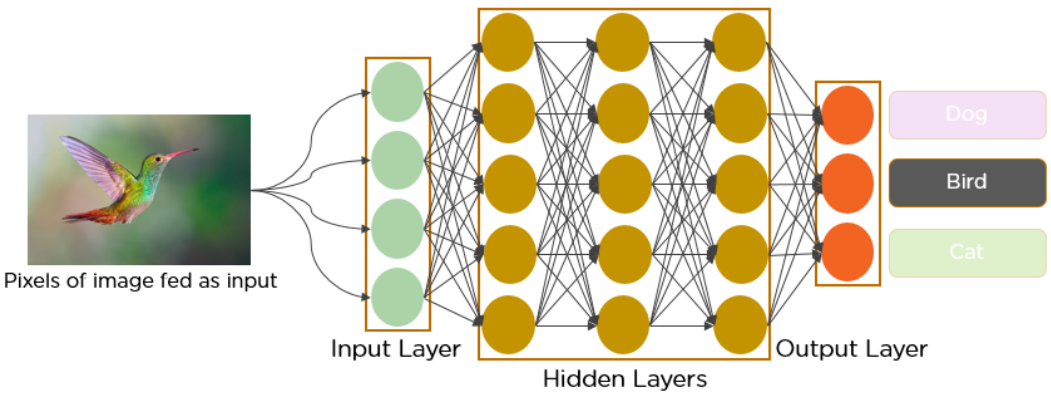


Figure 2: Layers of Convolutional Neuron Network

In pre-processing layer of CNN, with tend to have a smaller value for the network, it will rescale the pixel values of Red Green Blue (RGB) from 0 – 255 to 0 – 1. Next it will apply a convolutional layer which apply filters to images to learn. With several convolutional layer, it helps to extract features and distinguish the differences between different classes. In this project, there are 38 classes. Furthermore, it will have the max pooling 2D layer to decrease the feature map size by four times. It is because its only looks at every 2*2 patch of pixels in the feature map.

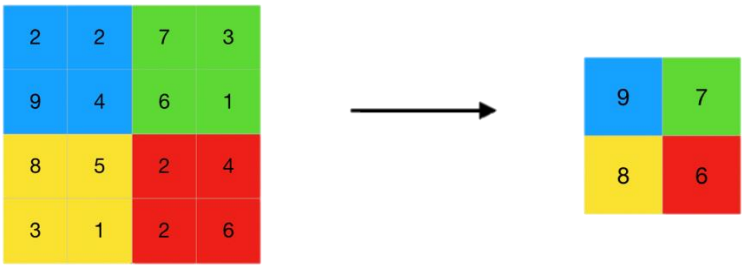


Figure 3: Max pooling 2D layer

After that, a dense layer with 128 neurons and 38 classes is created in this CNN.

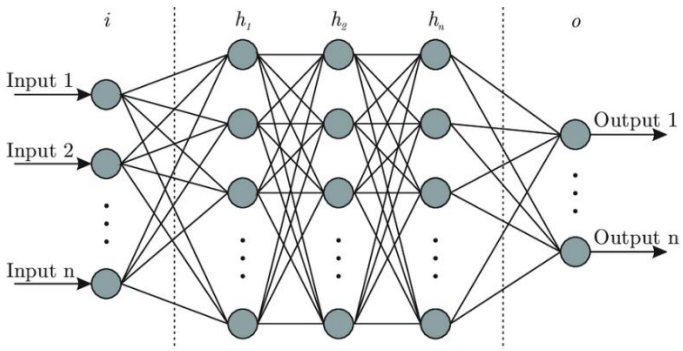


Figure 4: Neuron network diagram

Then, a model training process is needed to get an accurate deep learning neuron model that have losses will decrease consistently and accuracy increase steadily. The diagram below shows the training process and spot the graph of losses and accuracy in every epoch.

```
Epoch 1/10
3322/3322 [==============================] - 57s 17ms/step - loss: 1.6080 - accuracy: 0.5233 - val_loss: 0.9601 - val_accuracy: 0.7006
Epoch 2/10
3322/3322 [==============================] - 54s 16ms/step - loss: 0.8027 - accuracy: 0.7437 - val_loss: 0.7468 - val_accuracy: 0.7619
Epoch 3/10
3322/3322 [==============================] - 47s 14ms/step - loss: 0.6134 - accuracy: 0.8000 - val_loss: 0.7609 - val_accuracy: 0.7652
Epoch 4/10
3322/3322 [==============================] - 47s 14ms/step - loss: 0.5061 - accuracy: 0.8343 - val_loss: 0.6423 - val_accuracy: 0.7974
Epoch 5/10
3322/3322 [==============================] - 48s 14ms/step - loss: 0.4378 - accuracy: 0.8563 - val_loss: 0.5654 - val_accuracy: 0.8252
Epoch 6/10
3322/3322 [==============================] - 48s 14ms/step - loss: 0.3855 - accuracy: 0.8720 - val_loss: 0.4976 - val_accuracy: 0.8472
Epoch 7/10
3322/3322 [==============================] - 48s 14ms/step - loss: 0.3447 - accuracy: 0.8853 - val_loss: 0.3843 - val_accuracy: 0.8744
Epoch 8/10
3322/3322 [==============================] - 48s 14ms/step - loss: 0.3202 - accuracy: 0.8933 - val_loss: 0.3699 - val_accuracy: 0.8822
Epoch 9/10
3322/3322 [==============================] - 48s 14ms/step - loss: 0.2952 - accuracy: 0.9006 - val_loss: 0.3939 - val_accuracy: 0.8746
Epoch 10/10
3322/3322 [==============================] - 48s 14ms/step - loss: 0.2771 - accuracy: 0.9055 - val_loss: 0.3781 - val_accuracy: 0.8778
<keras.callbacks.History at 0x7f9778825510>
```

Figure 5: Neuron network training process

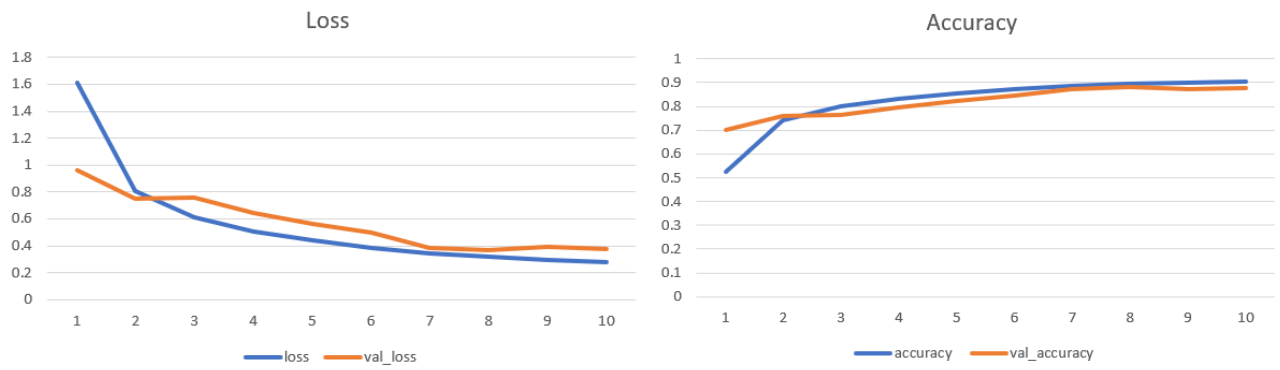


Figure 6: The graph of loss and accuracy during training

After the model is trained, a conversion to TensorFlow Lite (TFlite) model is needed because it enables on-device machine learning, which means it provide the availability to developers run the model on mobile, embedded and edge devices.

2.2.2 TensorFlow

TensorFlow is an end-to-end machine learning platform in which it is also known as a library. This library is important for the CNN network mentioned at above. All the capabilities to build and train the CNN network is provided by TensorFlow. It is flexible which it allows user to define own functionalities or services for the models. TensorFlow computational graphs represent the workflows that occur during deep learning model training. For a CNN model, the computational graph can be very complex. It is complex but TensorFlow provides an accessible and readable syntax which is essential for making these programming resources easier to use.

2.2.3 Google Colab

The platform or tool that use to create and train the neural network are Google Colab. The mentioned technology is used to build the model for Plant Scanner, which uses the platform Google Colab. It allows write and execute arbitrary python code through the browser and it is well suited to machine learning. In this assignment, it's been use for run the Interactive Python Notebook (.ipynb) to build the deep learning network and eventually convert it to a TFlite model. It is a hosted Jupyter notebook that requires no setup and has an excellent free version. Google Colab is chosen because it provided pre-installed libraries, cloud server, collaboration feature and free usage of GPU and TPU.

The pre-installed data libraries is helpful which reduce the burden of setting up the environment. The example libraries are Pandas, Numpy, Matplotlib, Keras, TensorFlow, and Pytorch and etc. In this project, a conversion to TensorFlow Lite model (TFlite) is needed because it enables on-device machine learning, which means it helps developers run the model on mobile, embedded and edge devices.

Google Colab allow users to save their data on cloud, which means user can access all the files by logging in their google account. It will no need to have all the related file on local machine. Thus, portability and flexibility is provided by using Google Colab.

The collaboration feature is available in Google Colab, which means developer can invite multiple developers to work together on a project by share the link or email invitation. This can increase the efficiency in terms of productivity.

Last but not least, the free GPU and TPU use is the most outstanding function they introduced. Developers will definitely decrease their efficiency if they faced any hardware issue, but with the GPU and TPU acceleration provided by them, it can make a huge difference for any machine learning projects. This is because it uses Google resources and will not mess with the processor or GPU of local machine.

2.2.4 Android Studio

On the other hand, a mobile app is needed to be developed in order to apply and adopt the model. The platform or tools that used to develop the Leaf Scanner is android studio. It is an integrated development environment (IDE) for Android application development. It enables user to design on the apps interface either using coding or drag-drop. The android emulator feature provided is also improve the debugging process which increase the efficient of developing the apps.

The Android Studio provide many features to ensure the productivity and effectiveness of building an android app. It is a flexible Gradle-based system. Gradle is a build automation tool for multi-language software development. It controls the development process in the tasks of compilation and packaging to testing, deployment, and publishing.

Apart from this, Android Studio features an Android emulator. The android emulator provides the availability for developers to debug and prototype their mobile applications. It will need no restarting the apps when any changes is applied to push code and resources.

It is also collab and integrate with GitHub, where users can get code templates to build common app features. This will help to increase the efficiency of development process where the developers no need to rebuild the wheel.

It is a unified environment where you can develop for all Android devices. It is compatible to all the android operating systems devices such as smartphones, tablets, Android Wear as well as Android TV devices.

2.3 Literature on similar systems available in the market.

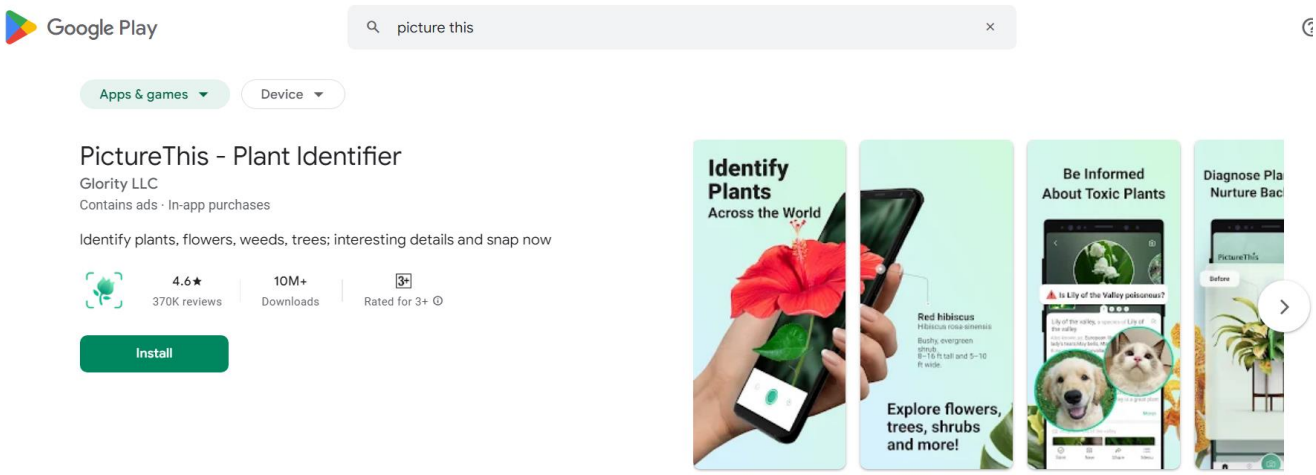


Figure 7: PictureThis - Plant Identifier

PictureThis is the similar system that is available in the market. This application is mainly focus on plant identifier which scan the image, then classify the type of plant. The processing time for classify the image is roughly 2 seconds. After the process, it will display the information of the plant, such as botanical name, description, related images, symbolism, interesting facts, characteristics, and care guide.

The diagrams below show one of the **positive** testing result from PictureThis. The testing is carried out by snapping the image of hibiscus.

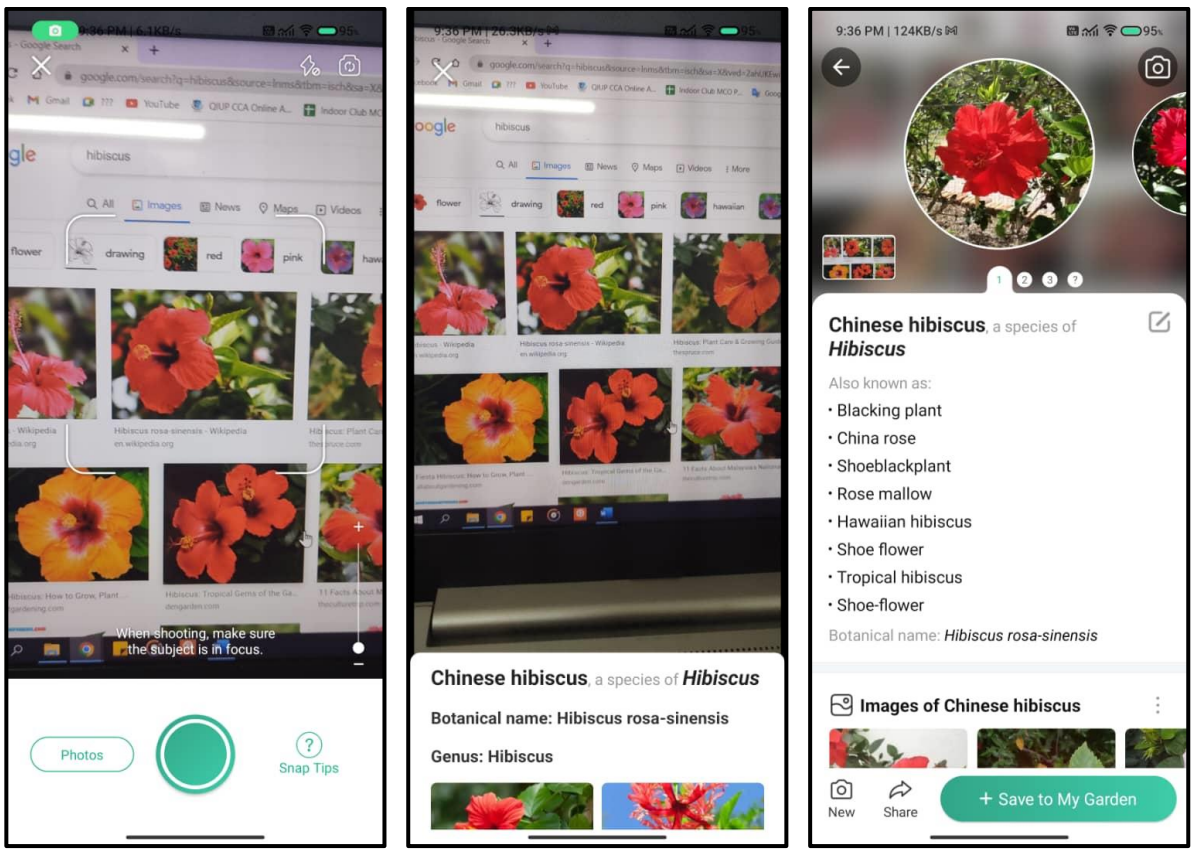


Figure 8: Process of positive testing

From the test result, it classified the image of hibiscus successfully and provide correct information of it to user.

The diagrams below show the **negative** testing result from PictureThis which the result is irrelevant. The testing is carried out by snapping the image of calculator.

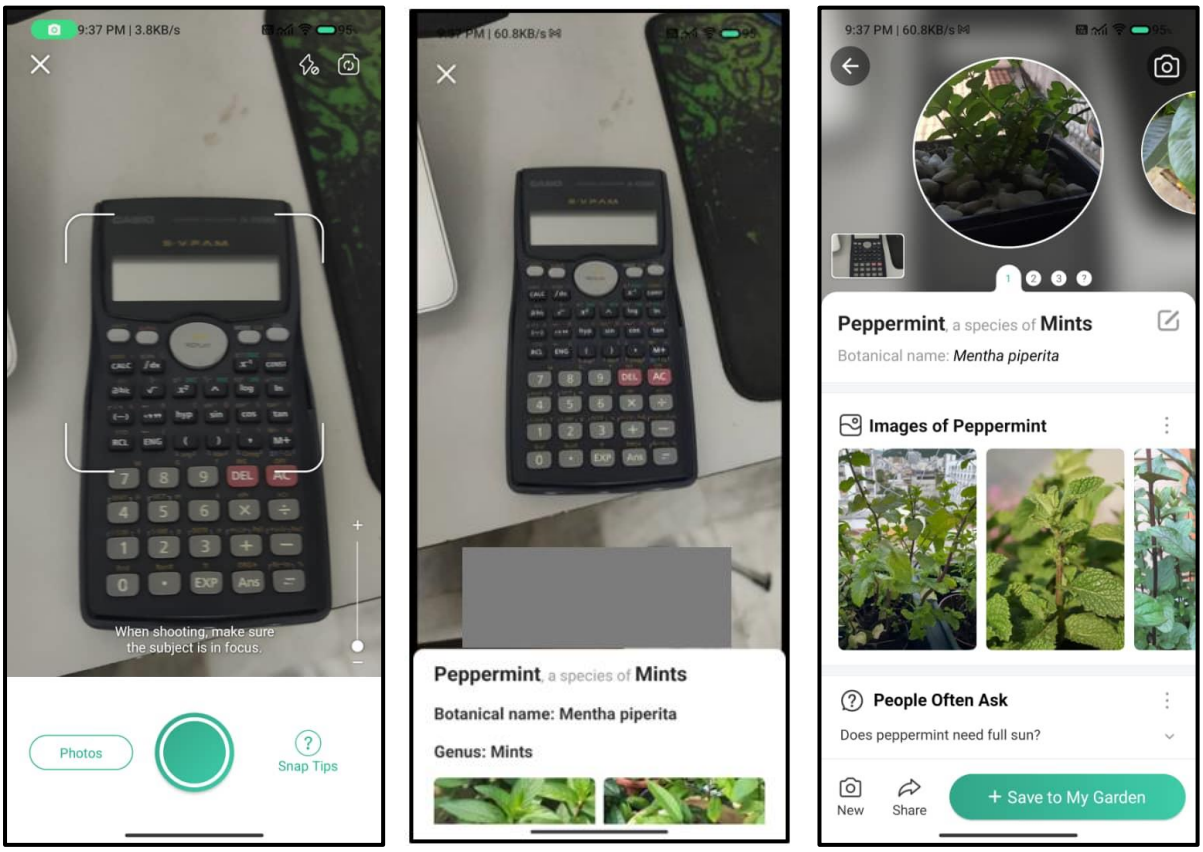


Figure 9: Process of negative testing

From the test result, it classified the image of calculator as peppermint and provide information of peppermint to user.

From both testing that it can be assumed the program examine the image and get the details of highest confidence value and take it as the output. Basically the concept used of PictureThis is similar to Leaf Scanner which is image classification, which the assumption of classified image is made based on the highest confidence value of input image to the loaded model.

Comparison between the Leaf Scanner and PictureThis

Table 1: Comparison between Leaf Scanner and PictureThis

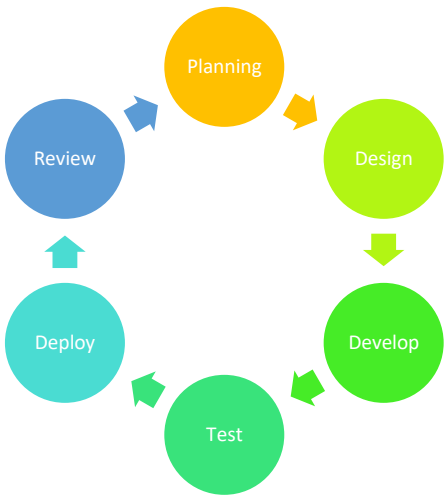
	Leaf Scanner	**PictureThis**
Different	Mainly focus on leaf images, and classify the plant name and it health status by input the image.	Mainly focus on plant image, and classify the plant name and all description and information that able to found online.
Image processing time	t <= 1 seconds	t >= 2 seconds
Internet connection	Not required	Required
Accuracy (positive tests)	78.94%	100%
Description of accuracy	The results is assumed from the first round testing at section 4.1.1. Which out of 38 classes, there are 8 classes is misclassified	The results is assumed from a user perspective of the mobile apps. So far there is no misclassification of images when positive test case is carried out. (Which means only input the images related to plant)

Chapter 3: Methodology

3.1 Discussion on the selected methodology in relation to the design and development of the intelligent agent.

3.1.1 Development methodology

The methodology use in this Leaf Scanner project is Agile model.



This project is managed by segmenting it into several phases using the agile technique. Continuous improvement is required at every level, as well as ongoing cooperation with stakeholders. Once the project begins, it will cycle the process planning, design, develop, test, deploy and review until the stage is no other issue.

There are many advantages by using agile model. First it can reduce the risks. Agile works in small sprint that focus on continuous output thus the project team will regularly access and assess the processes. Second, it increases flexibility. Agile divides the project in short sprints that are both manageable and flexible enough to allow the team to implement changes on short notice. Third, it can improve project predictability. Since project divided in short sprints, therefore the visibility of the project is increased so effective mitigation plans become easier to develop.

The works of project are break into 3 sprints; the work breakdown structure are as follows:

3.1.2 Work Breakdown Structure

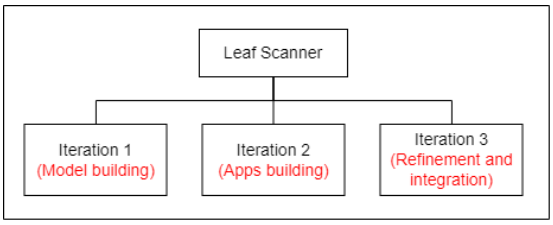


Figure 10: Work breakdown structure

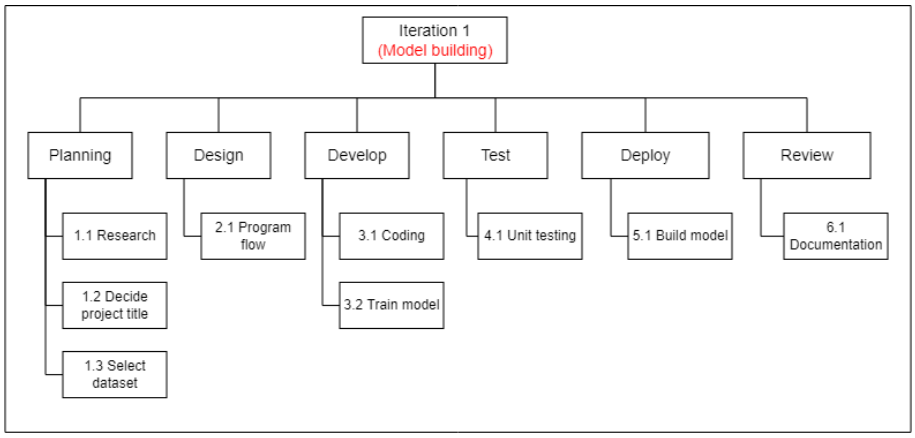


Figure 11: First iteration of work breakdown structure

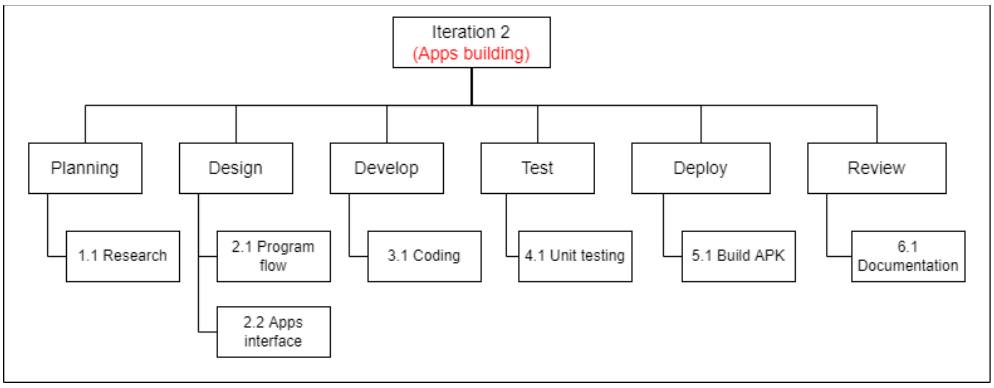


Figure 12: Second iteration of work breakdown structure

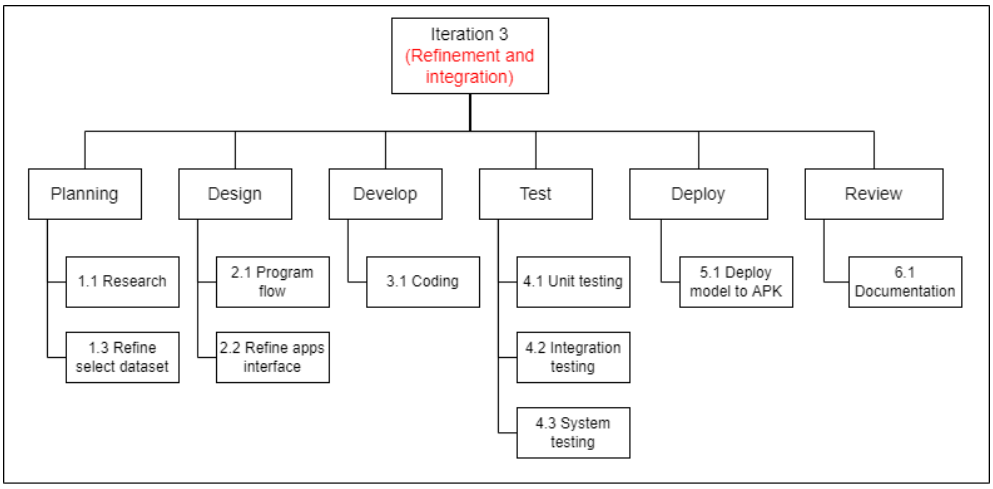


Figure 13: Third iteration of work breakdown structure

3.1.3 Functional and non-functional requirements

Functional requirement

1. User able to input the image by the camera function on phone.

2. User able to input the image by the select image of gallery on phone.

3. The application able to perform image classification after user input the image.

Non-functional requirement

1. Speed
    - Process time to perform image classification is within 1 second.
2. Compatibility
    - The mobile application compatible to any Android based operating system phone.

3.1.4 Journey Map

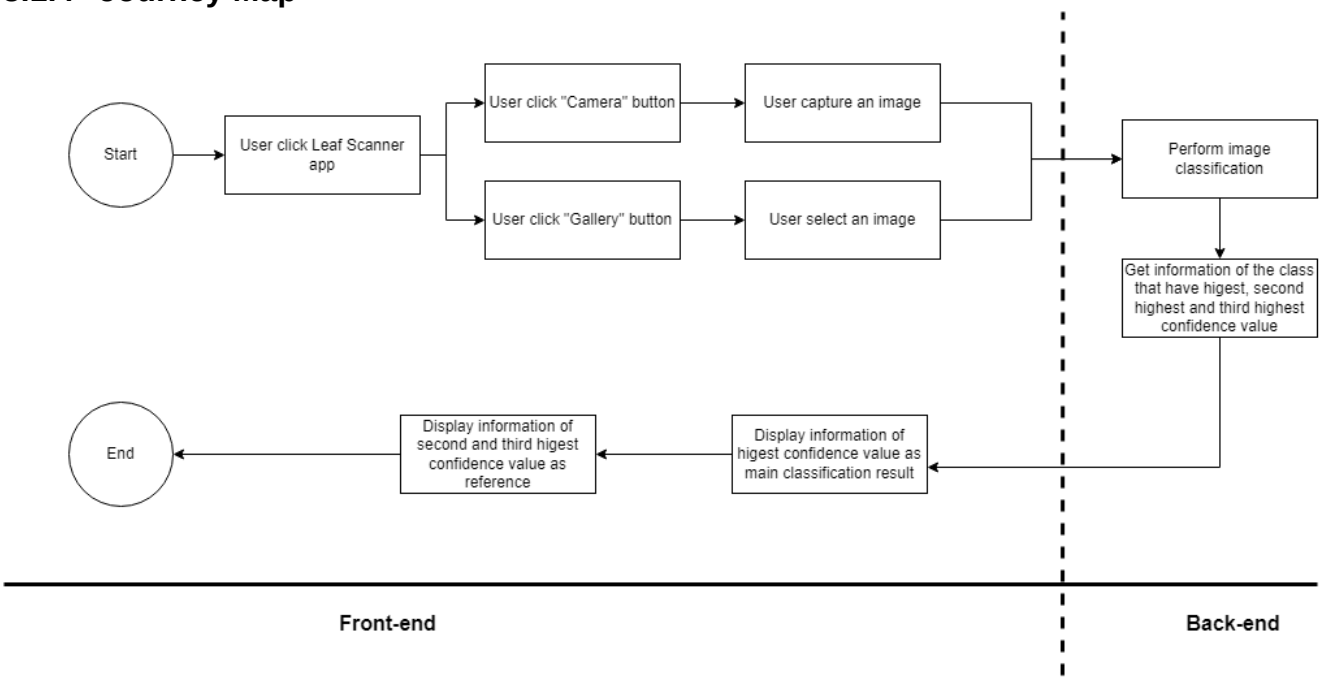


Figure 14: Journey map of Leaf Scanner

3.1.5 Use Case Diagram

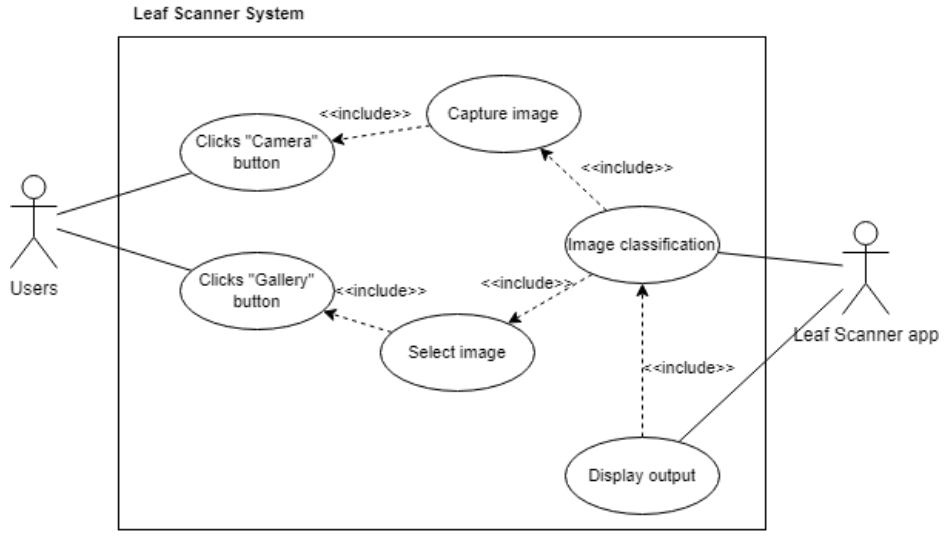


Figure 15: Use case diagram of Leaf Scanner system

3.1.6 Flow Chart Diagram

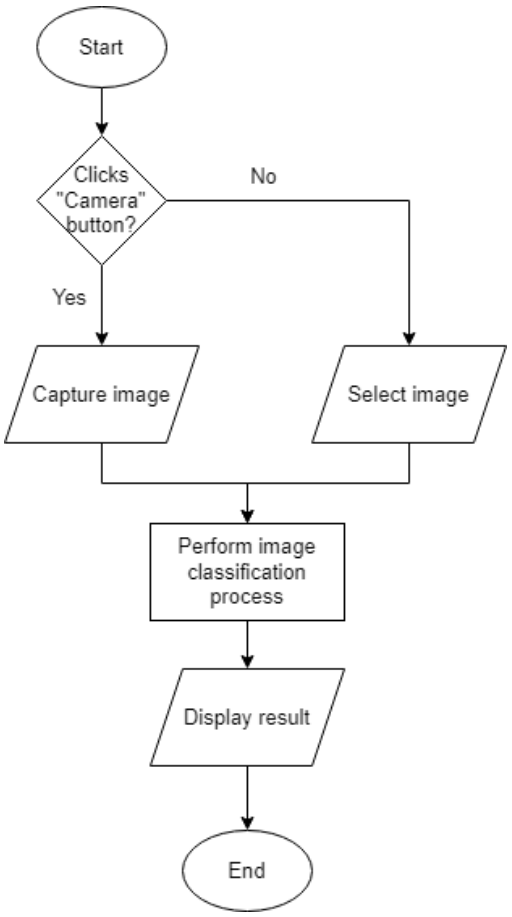


Figure 16: Flowchart of Leaf Scanner

3.1.7 User interface design

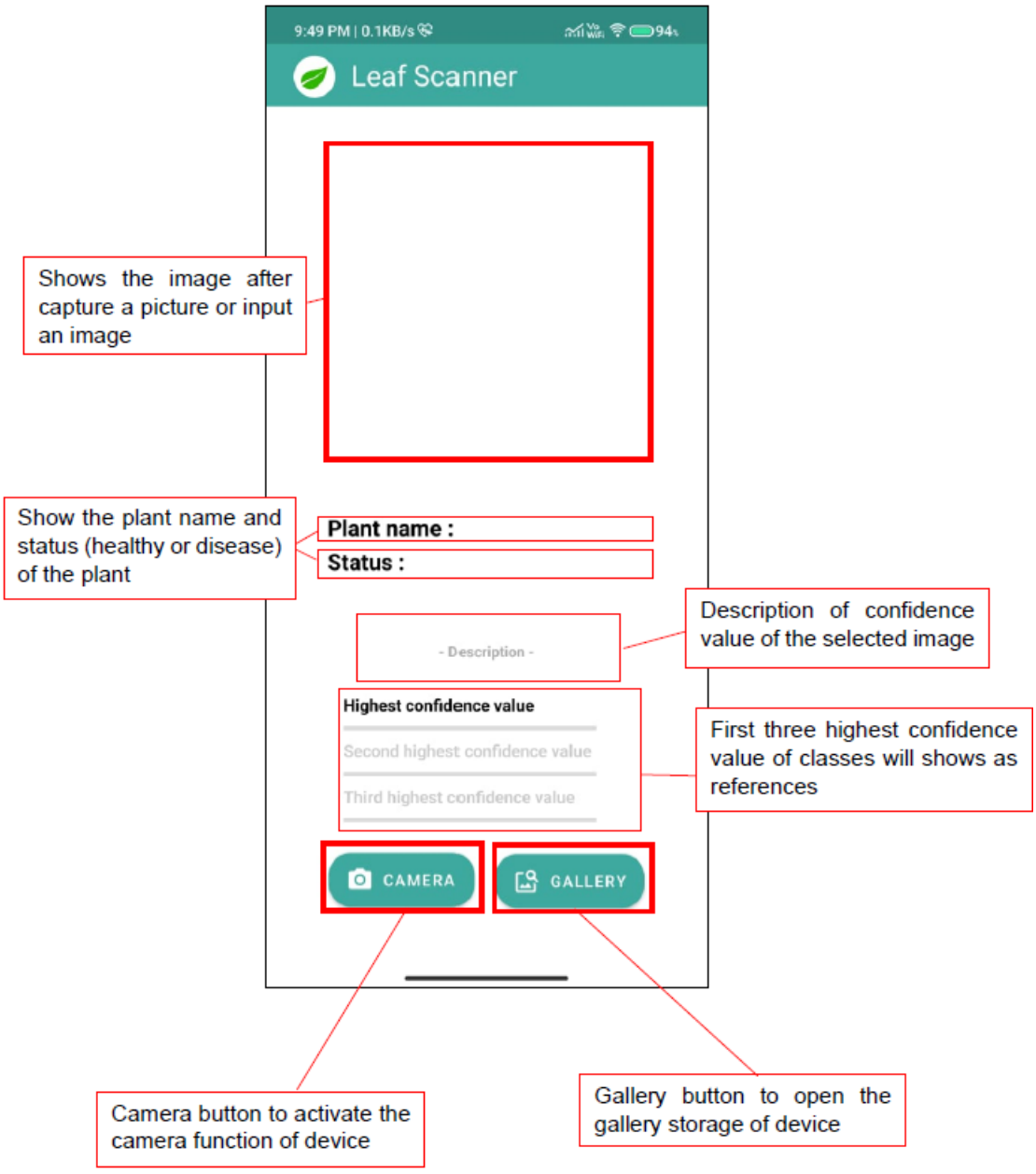


Figure 17: User interface design of Leaf Scanner mobile app

Chapter 4: Implementation, Results Analysis and Discussion

4.1 The detailed description and explanation for each of the stages of the intelligent agent's operation, and the results of the testing and implementation with detailed analysis and discussion.

Detailed description of testing procedures

Leaf Scanner is an app that able to scan on images and know the plant type and plant disease. The interface design of the application shows as follows:

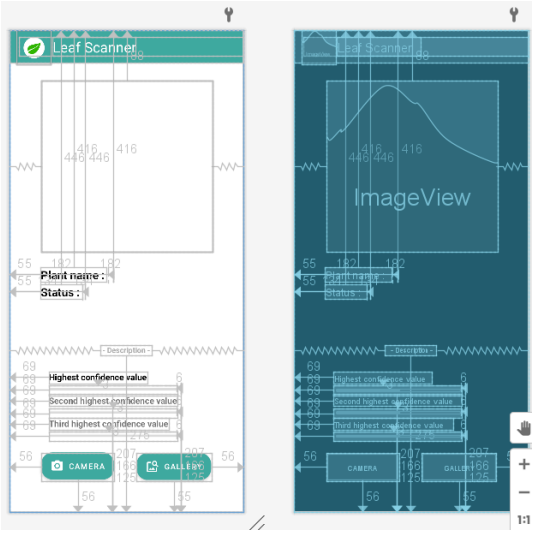


Figure 18: Interface design of Leaf Scanner

There are two buttons which are “Camera” and “Gallery”. The camera button will ask the permission to open camera to capture image.

Figure 19: User clicked camera button

Whereby the gallery button is featured to access the gallery storage and select images.

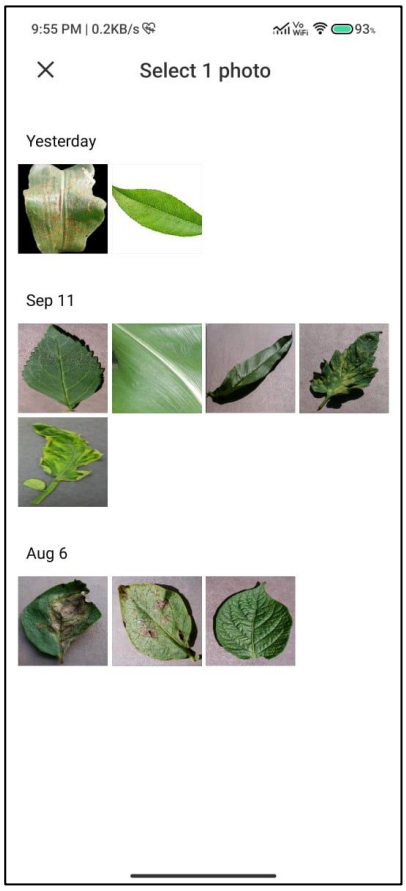


Figure 20: User clicked gallery button

After captured an image or selected an image, the program will classify the image by load the model that have trained.

With the confidences value of input image compared to all 38 classes, it will take the highest value and show the class details as output. Apart from this, it will also show the details of second highest and third highest confidence value as reference to users.

The diagram below shows the example of positive testing.

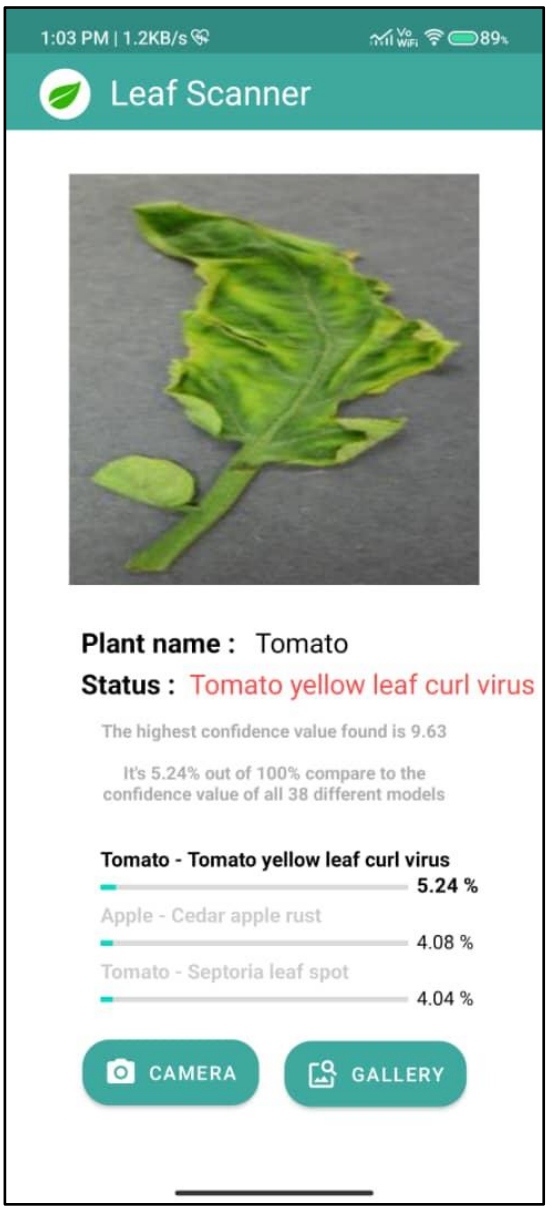


Figure 21: Example output of postive testing

It will display the picture of selected image, plant name as well as the status of the plant. The description at middle is showing the confidence value percentage. The percentage is low because it is the value compared to all classes. Even though there is only 5.24% but it is the highest confidence value which is 9.63. In this test case, the confidence value 9.63 = 5.24%, meanwhile 100% = 183.78, where 183.78 is the total confidence value of 38 classes. This also shown that 9.63/183.78 * 100 = 5.24%.

Basically, there is 3 rounds of testing will perform on next page onwards. Round 1 test all 38 classes by input related images. Round 2 will re-test on the negative result that input get from round 1 testing by input different related image. Round 3 will tests on irrelevant images.

4.1.1 Testing – Round 1

Round 1 test all 38 classes by input related images.

Apple - Apple scab

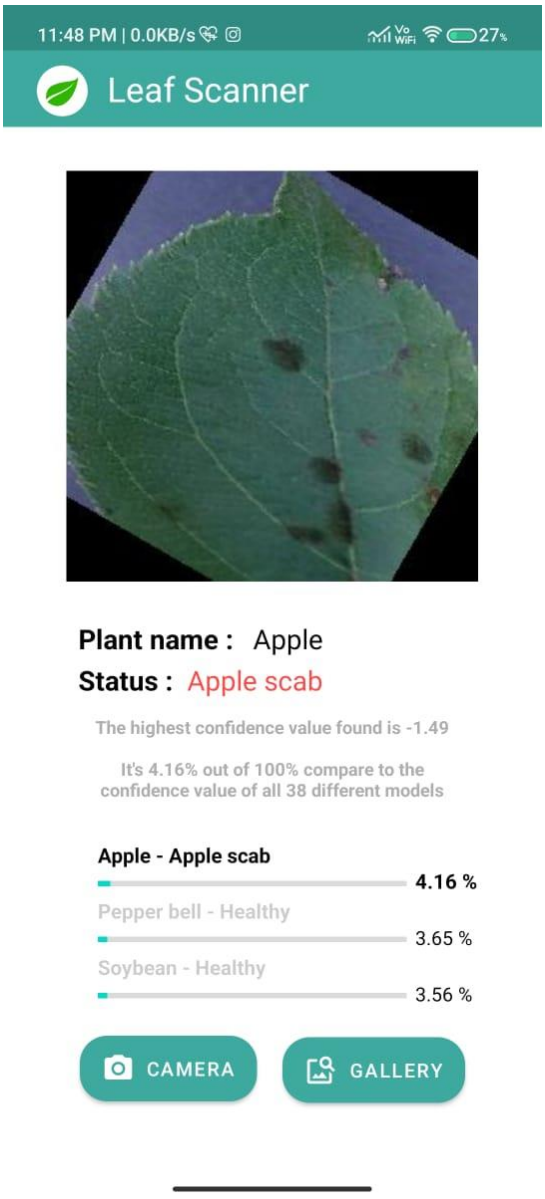


Figure 22: Test result of apple - scab

The image is identified correctly as apple – apple scab. Which the class has highest confidence value among all 38 classes.

Apple - Black rot

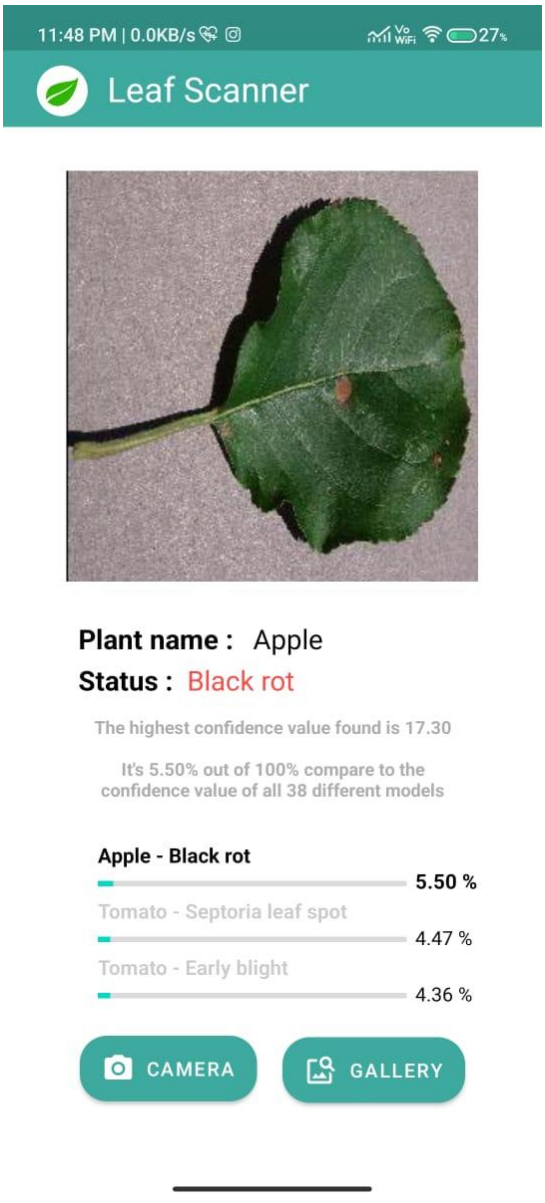


Figure 23: Test result of apple - black rot

The image is identified correctly as apple – black rot. Which the class has highest confidence value among all 38 classes.

Apple - Cedar apple rust

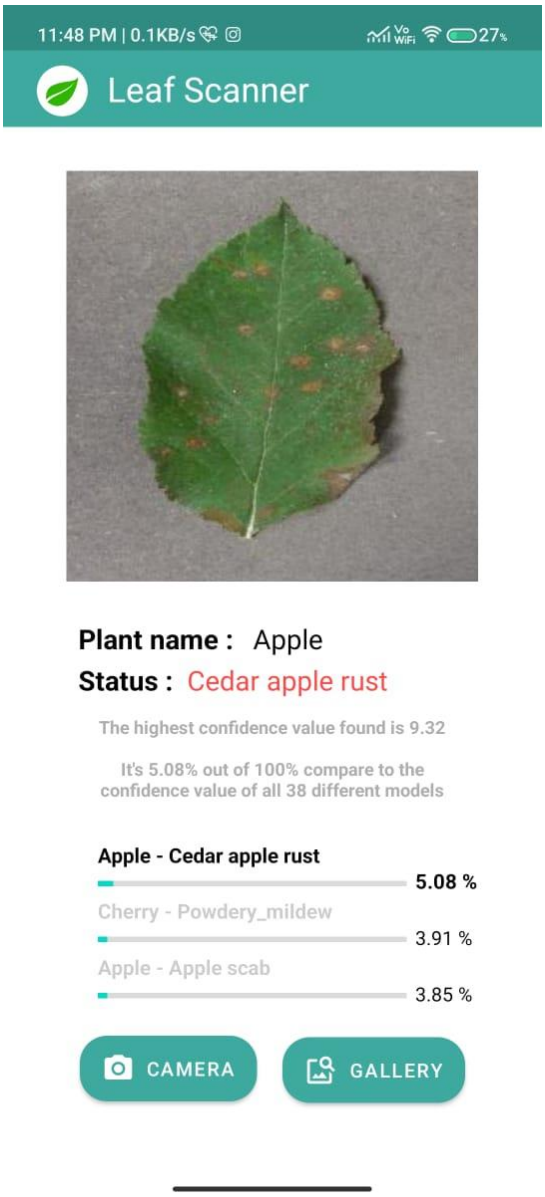


Figure 24: Test result of apple -cedar apple rust

The image is identified correctly as apple - cedar apple rust. Which the class has highest confidence value among all 38 classes.

Apple - Healthy

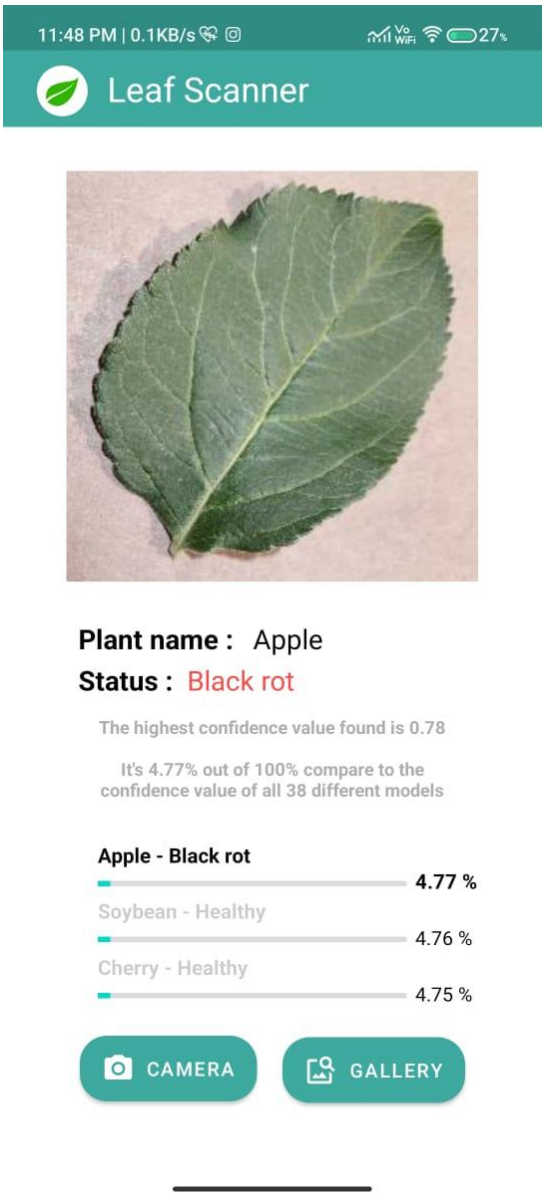


Figure 25: Test result of apple - healthy

The image identified wrongly as Apple – Black rot, which the image is showing an image of apple – healthy. This is because it is high similarity of the image pattern and cause the confidence value of apple – black rot is higher than apple - healthy.

Blueberry – Healthy

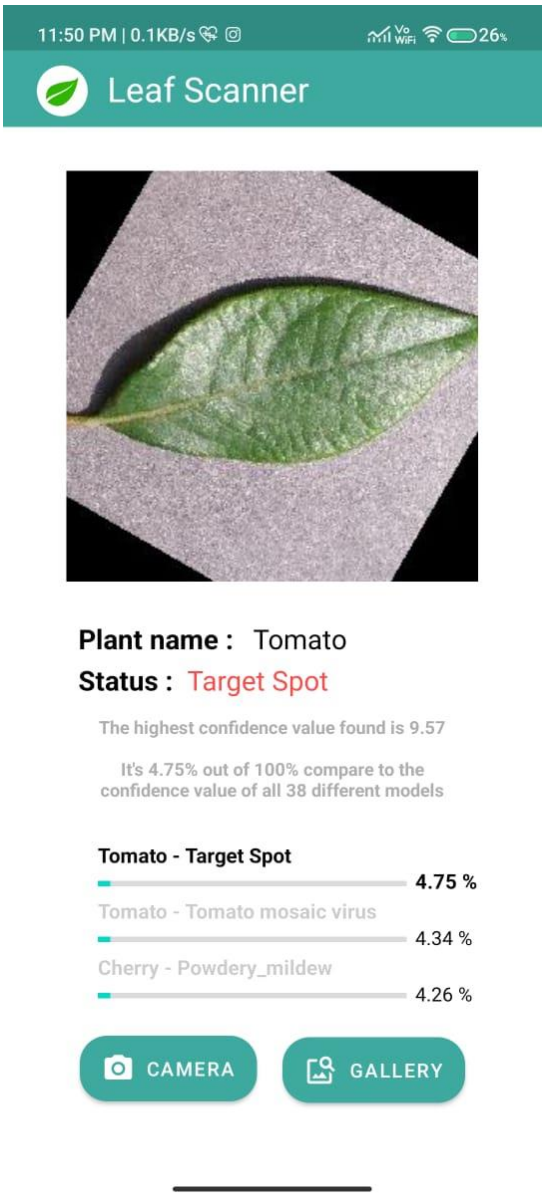


Figure 26: Test result of blueberry - healthy

The image identified wrongly as tomato – target spot, which the image is showing an image of blueberry – healthy. This is because it is high similarity of the image pattern and cause the confidence value of tomato – target spot is higher than blueberry – healthy.

Cherry – Healthy

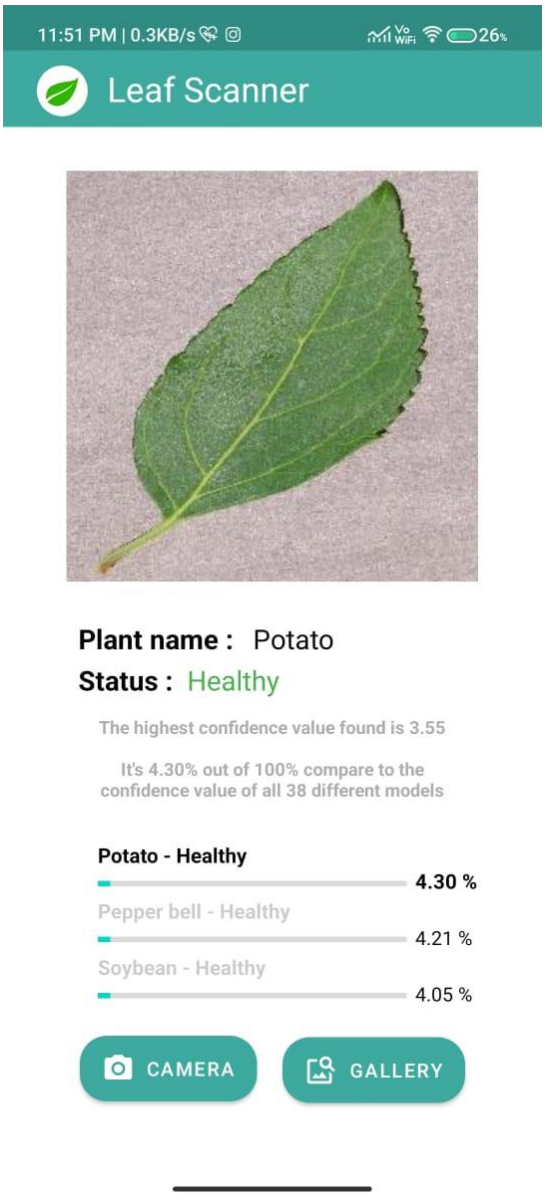


Figure 27: Test result of cherry - healthy

The image identified wrongly as potato - healthy, which the image is showing an image of cherry – healthy. This is because it is high similarity of the image pattern and cause the confidence value of potato - healthy is higher than cherry – healthy.

Cherry – Powdery mildew

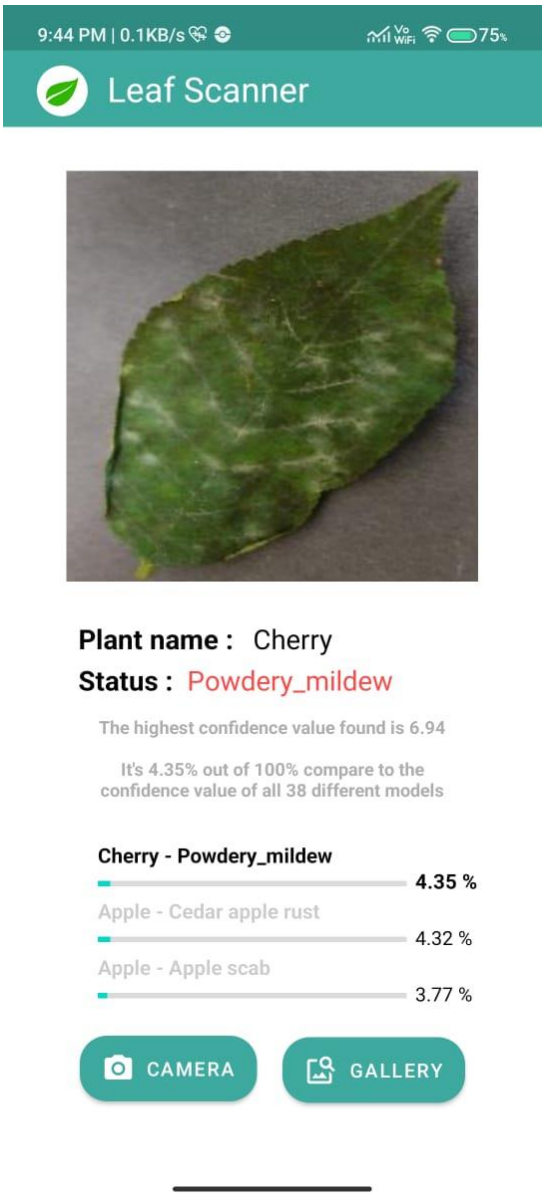


Figure 28: Test result of cherry - powdery mildew

The image is identified correctly as cherry – powdery mildew. Which the class has highest confidence value among all 38 classes.

Corn - Cercospora leaf spot / Grayleaf spot

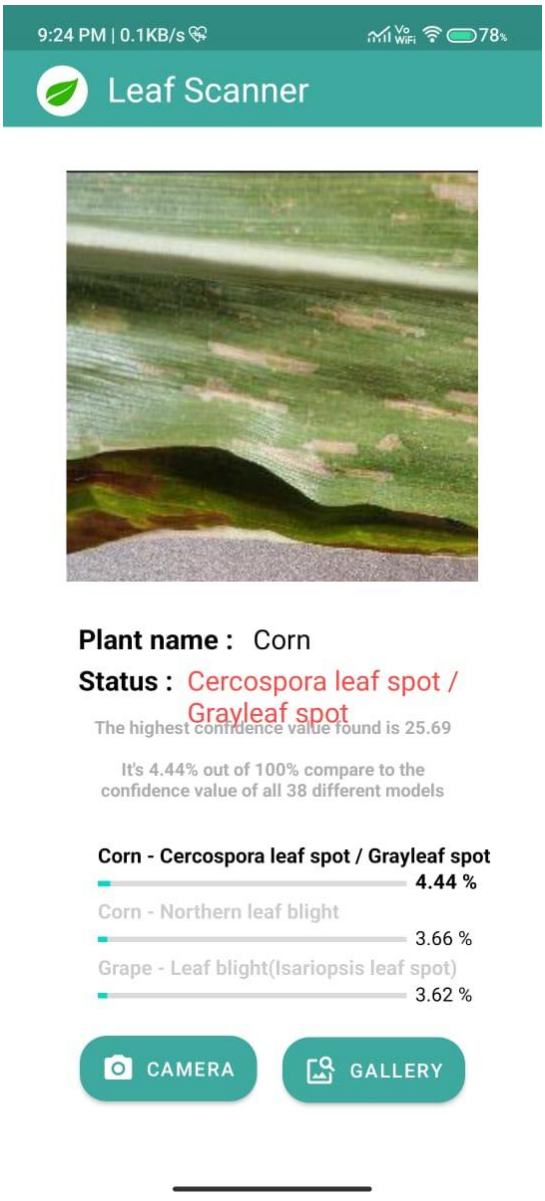


Figure 29: Test result of corn - cerospora leaf spot/ grayleaf spot

The image is identified correctly as corn - cercospora leaf spot / grayleaf spot. Which the class has highest confidence value among all 38 classes.

Corn - Common rust

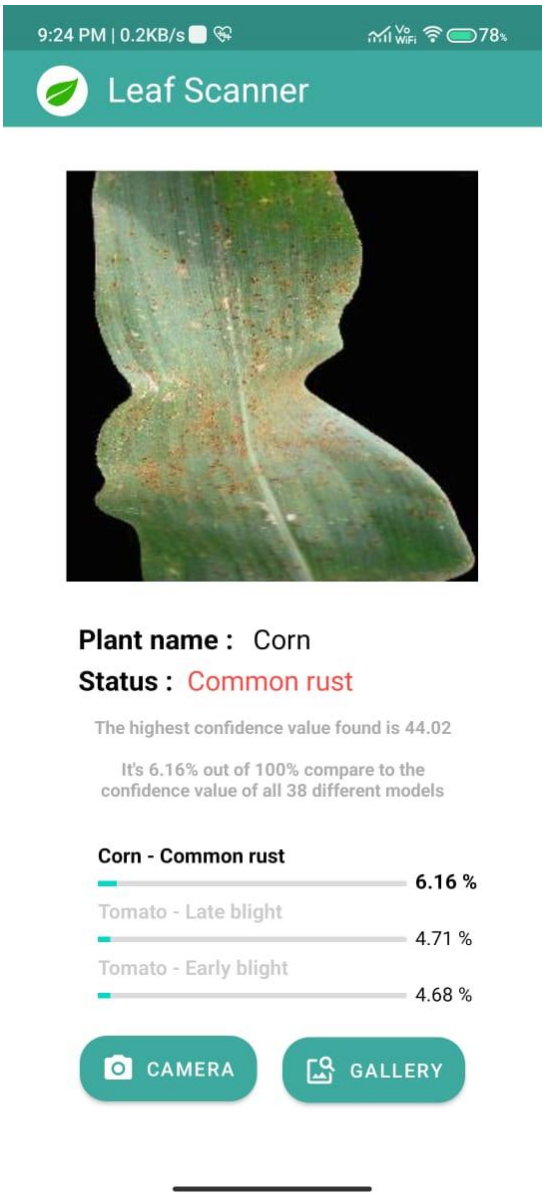


Figure 30: Test result of corn - common rust

The image is identified correctly as corn - common rust. Which the class has highest confidence value among all 38 classes.

Corn – Healthy

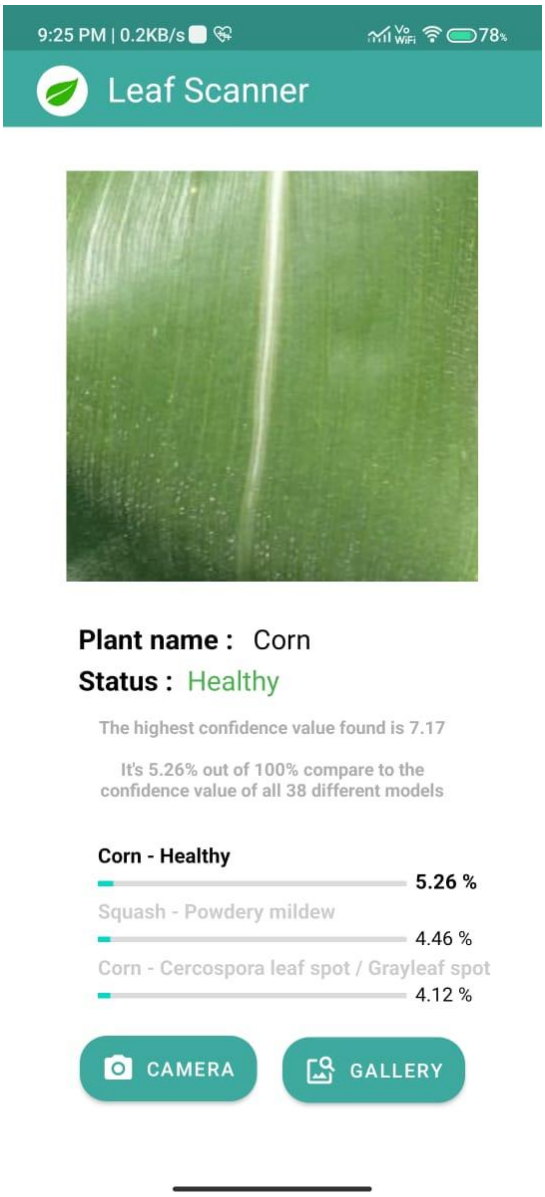


Figure 31: Test result of corn -healthy

The image is identified correctly as corn – healthy. Which the class has highest confidence value among all 38 classes.

Corn - Northern leaf blight

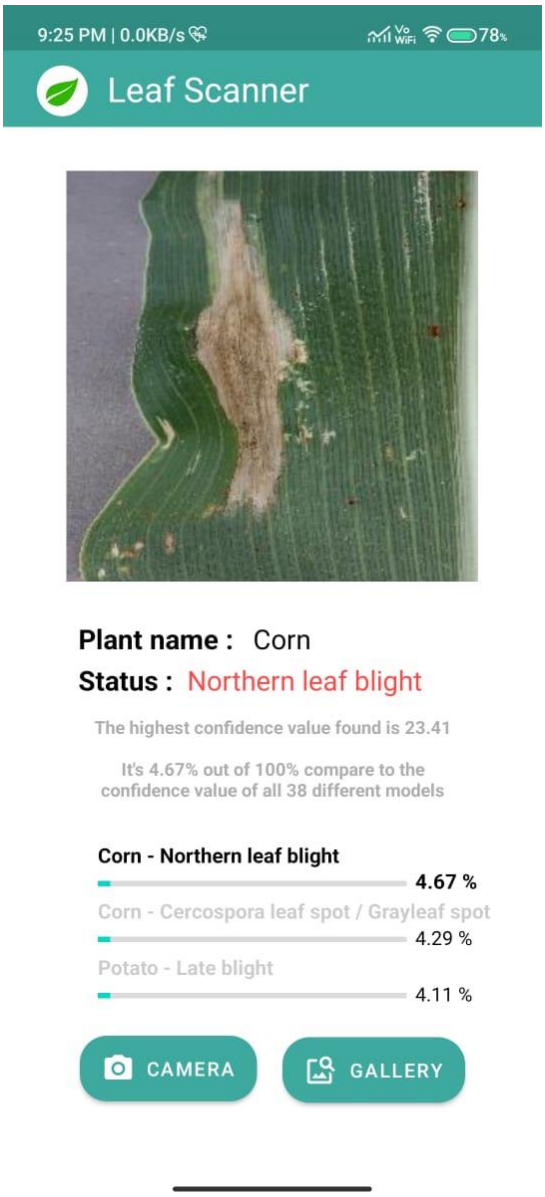


Figure 32: Test result of corn - northern leaf blight

The image is identified correctly as corn - northern leaf blight. Which the class has highest confidence value among all 38 classes.

Grape - Black rot

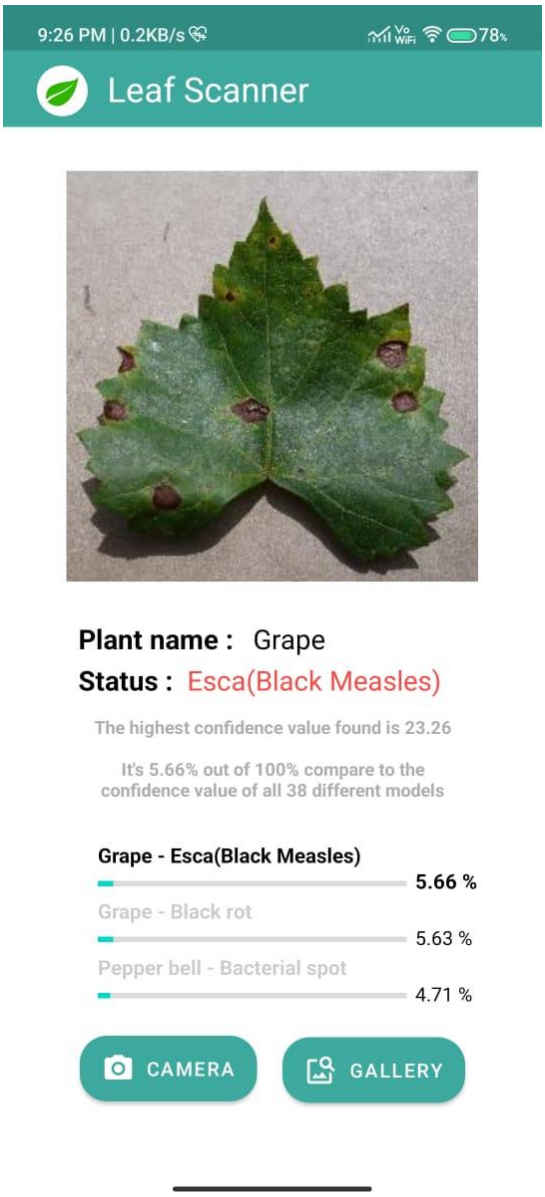


Figure 33: Test result of grape - black rot

The image identified wrongly as grape – esca (Black Measles), which the image is showing an image of grape - black rot. This is because it is high similarity of the image pattern and cause the confidence value of grape – esca (Black Measles) is higher than grape - black rot.

Grape - Esca(Black Measles)

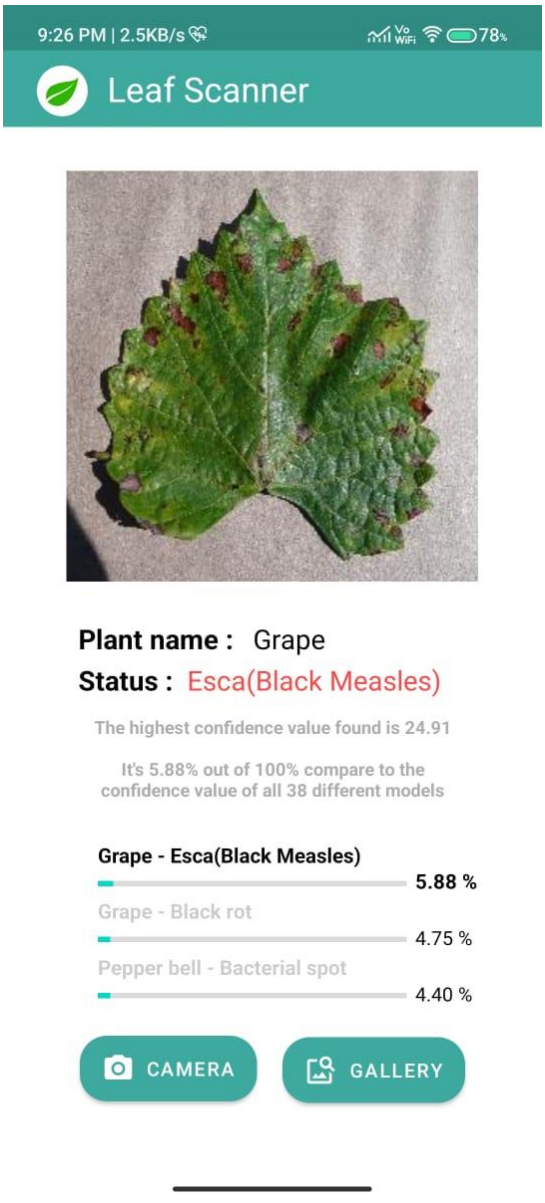


Figure 34: Test result of grape - esca (Black Measles)

The image is identified correctly as grape – esca (Black Measles). Which the class has highest confidence value among all 38 classes.

Grape – Healthy

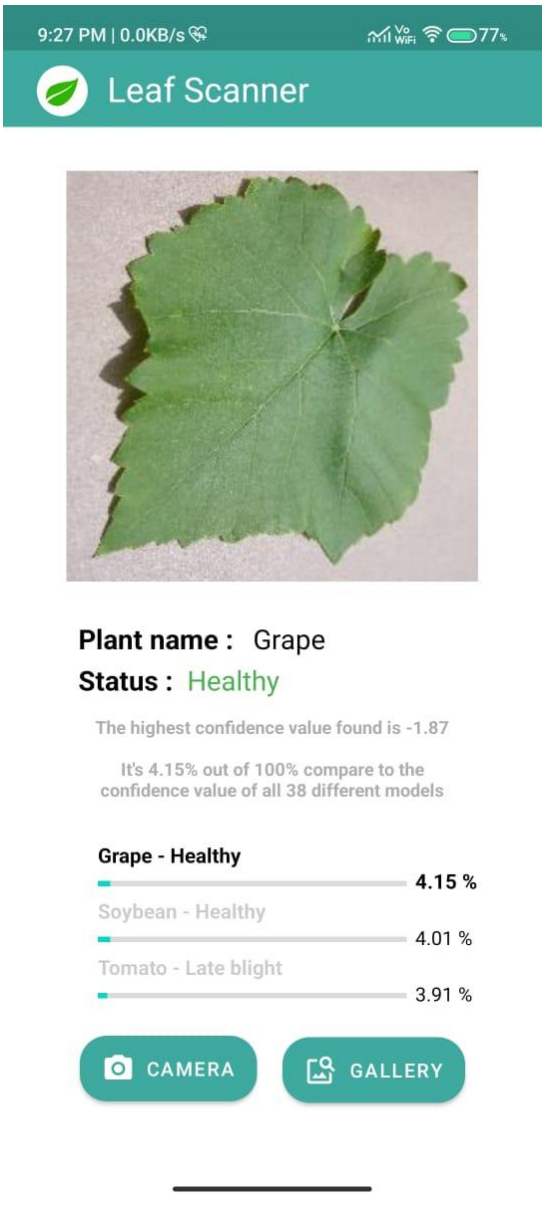


Figure 35: Test result of grape - healthy

The image is identified correctly as grape – healthy. Which the class has highest confidence value among all 38 classes.

Grape - Leaf blight (Isariopsis leaf spot)

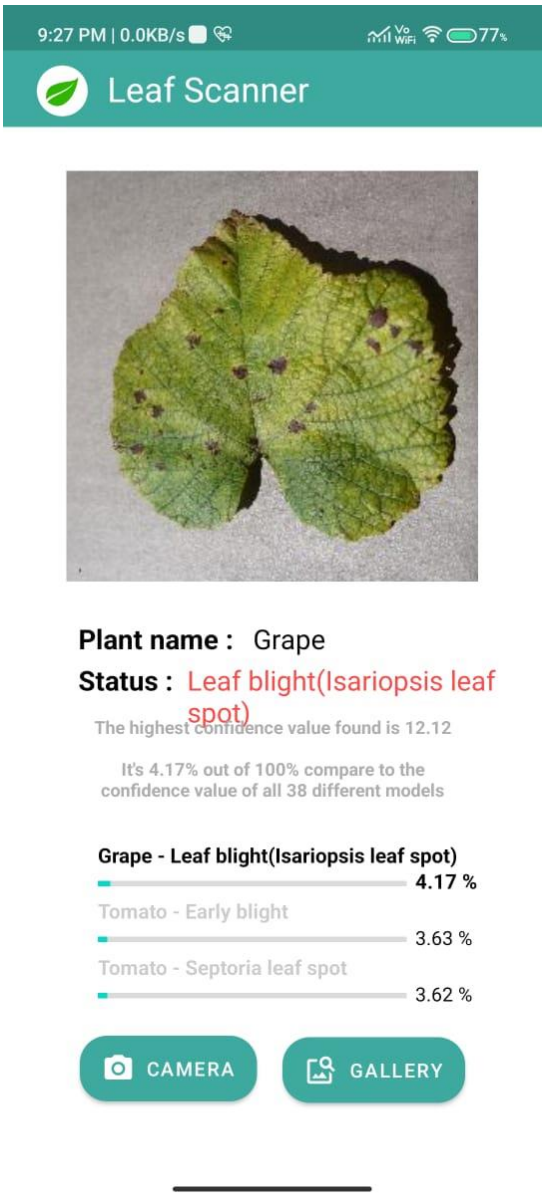


Figure 36: Test result of grape - leaf blight (Isariopsis leaf spot)

The image is identified correctly as grape - leaf blight (Isariopsis leaf spot). Which the class has highest confidence value among all 38 classes.

Orange - Haunglongbing(Citrus greening)

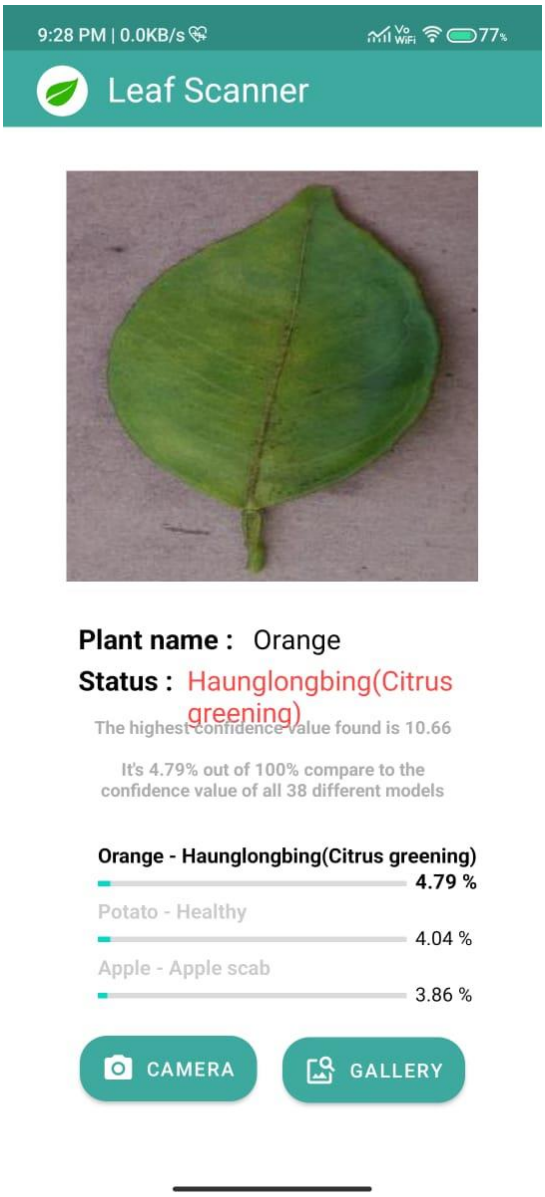


Figure 37: Test result of orange - Haunglongbing

The image is identified correctly as orange - Haunglongbing(Citrus greening). Which the class has highest confidence value among all 38 classes.

Peach - Bacterial spot

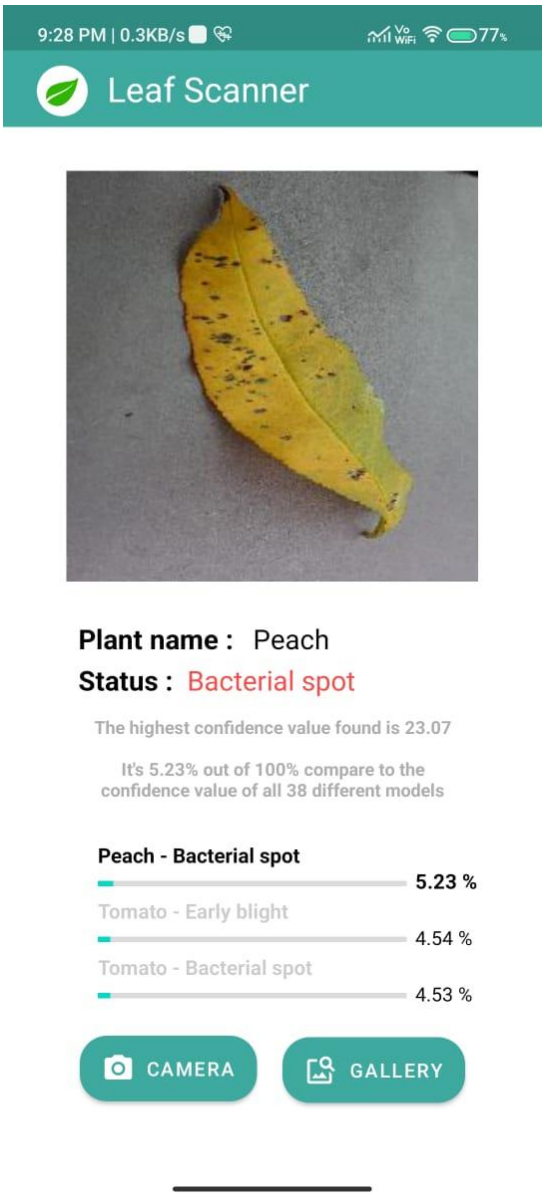


Figure 38: Test result of peach -bacterial spot

The image is identified correctly as peach - bacterial spot. Which the class has highest confidence value among all 38 classes.

Peach – Healthy

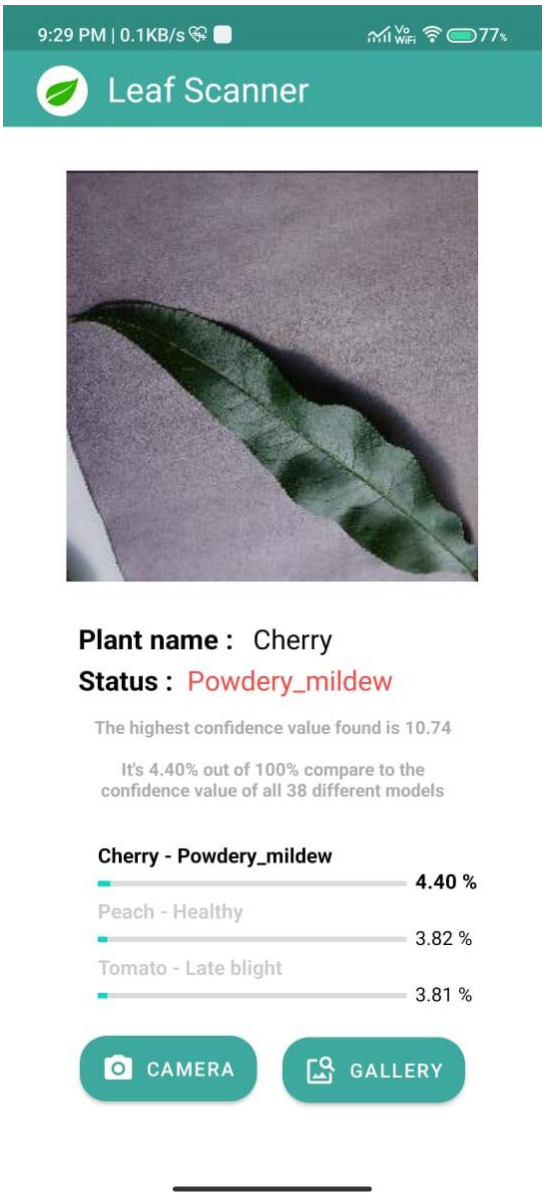


Figure 39: Test result of peach - healthy

The image identified wrongly as cherry – powdery mildew, which the image is showing an image of peach - healthy. This is because it is high similarity of the image pattern and cause the confidence value of cherry – powdery mildew is higher than peach - healthy.

Pepper bell - Bacterial spot

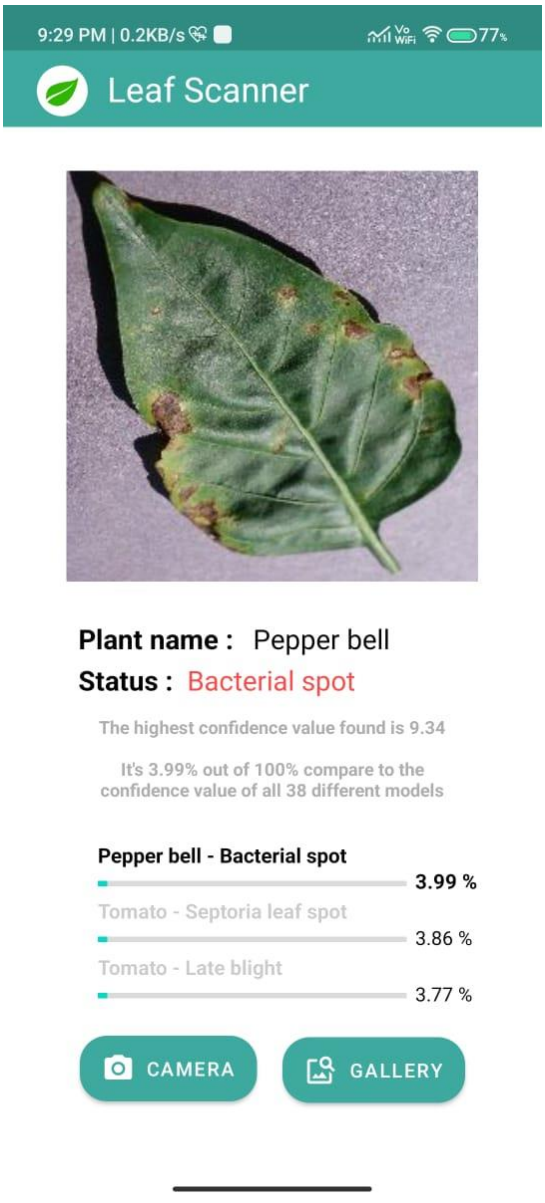


Figure 40: Test result of pepper bell - bacterial spot

The image is identified correctly as pepper bell - bacterial spot. Which the class has highest confidence value among all 38 classes.

Pepper bell – Healthy

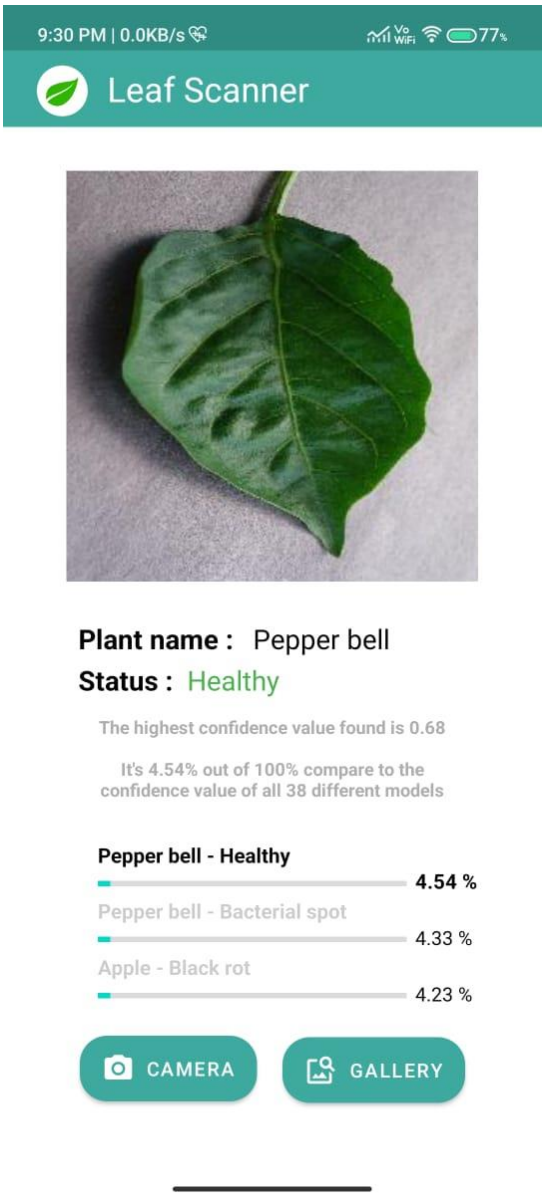


Figure 41: Test result of pepper bell - healthy

The image is identified correctly as pepper bell – healthy. Which the class has highest confidence value among all 38 classes.

Potato - Early blight

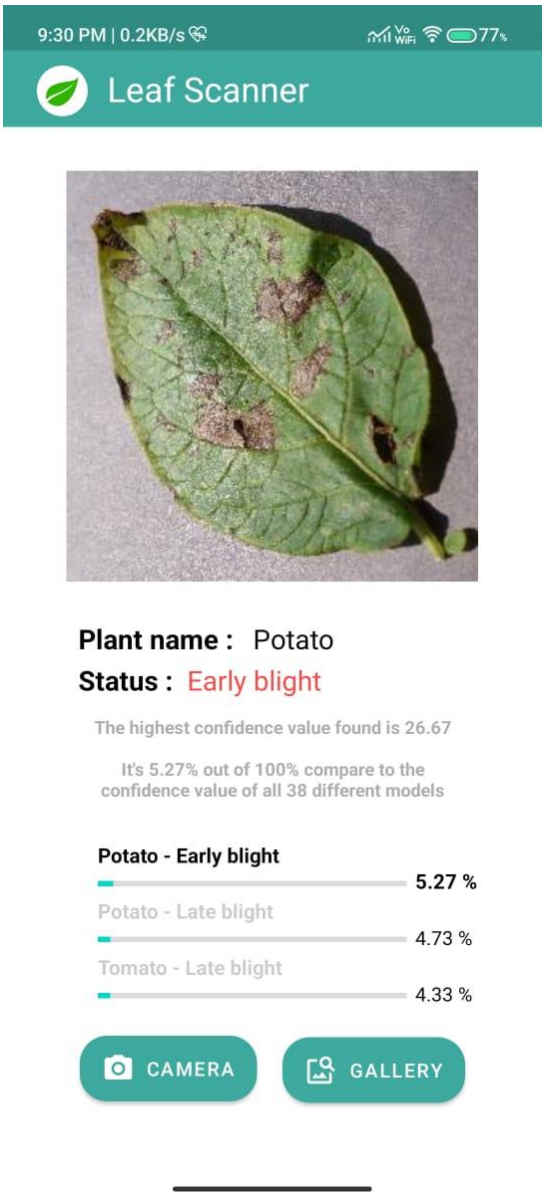


Figure 42: Test result of potato - early blight

The image is identified correctly as potato - early blight. Which the class has highest confidence value among all 38 classes.

Potato – Healthy

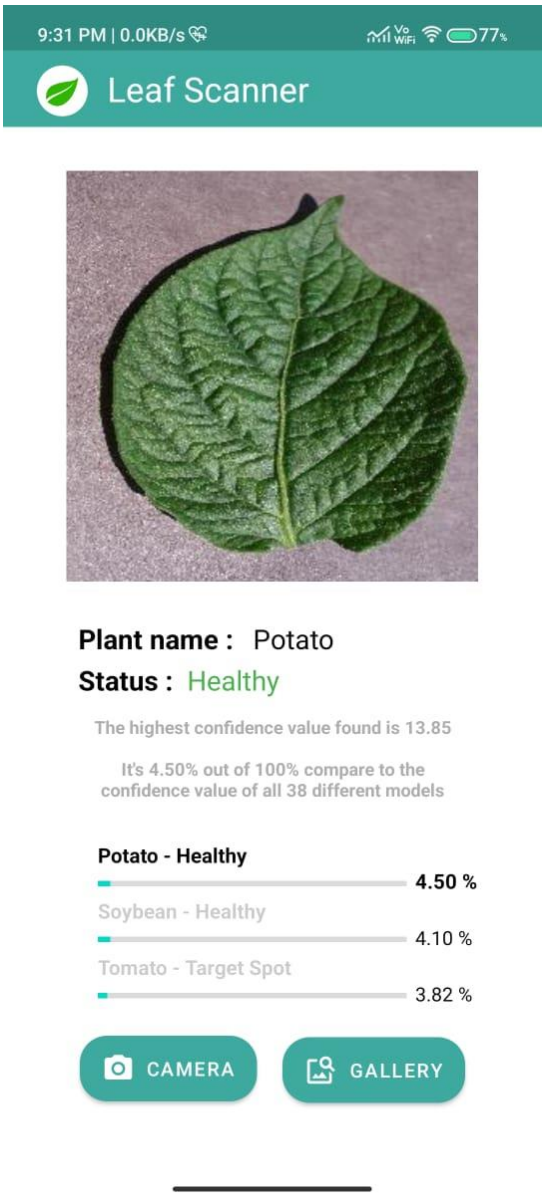


Figure 43: Test result of potato - healthy

The image is identified correctly as potato – healthy. Which the class has highest confidence value among all 38 classes.

Potato - Late blight

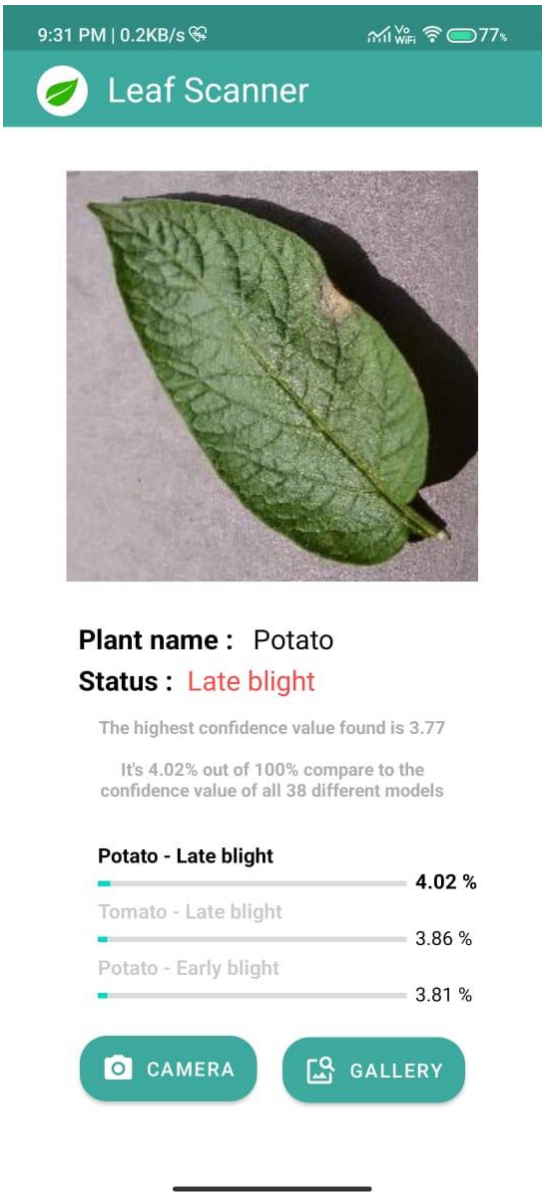


Figure 44: Test result of potato - late blight

The image is identified correctly as potato - late blight. Which the class has highest confidence value among all 38 classes.

Raspberry – Healthy

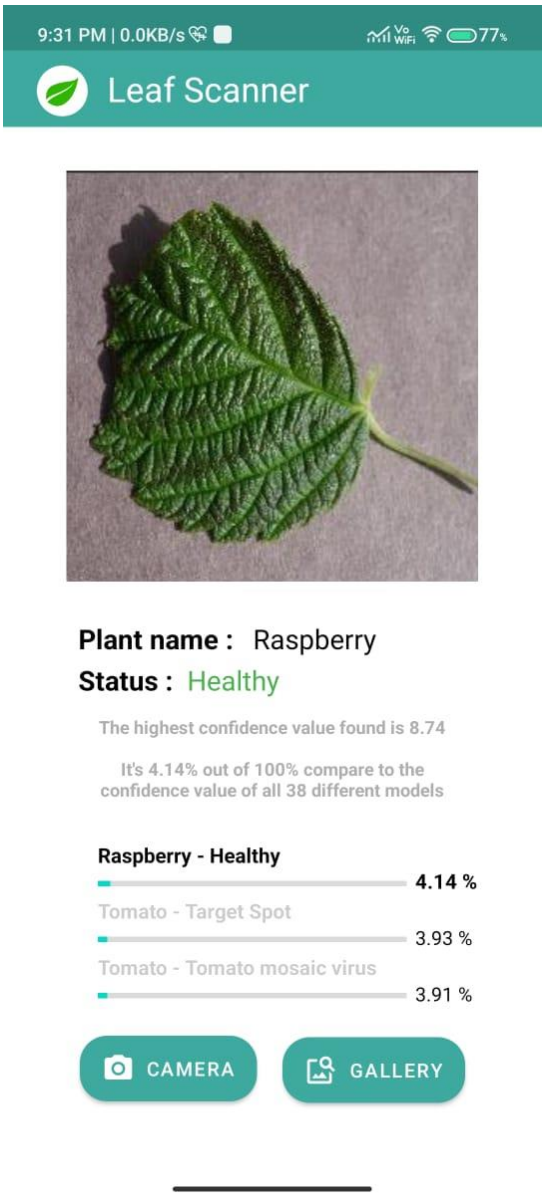


Figure 45: Test result of raspberry - healthy

The image is identified correctly as raspberry – healthy. Which the class has highest confidence value among all 38 classes.

Soybean – Healthy

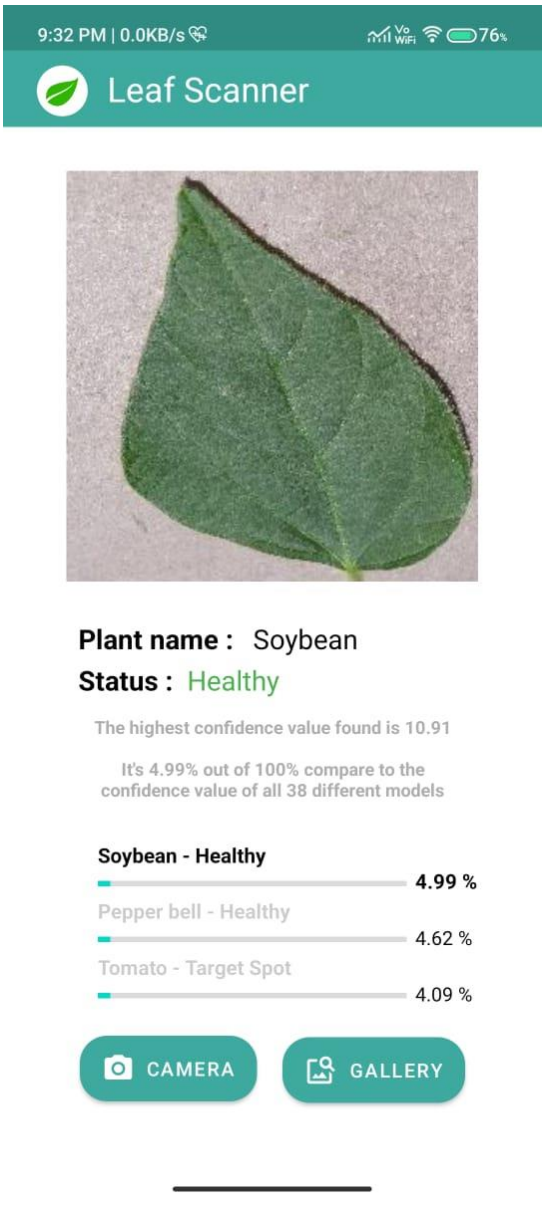


Figure 46: Test result of soybean - healthy

The image is identified correctly as soybean – healthy. Which the class has highest confidence value among all 38 classes.

Squash - Powdery mildew

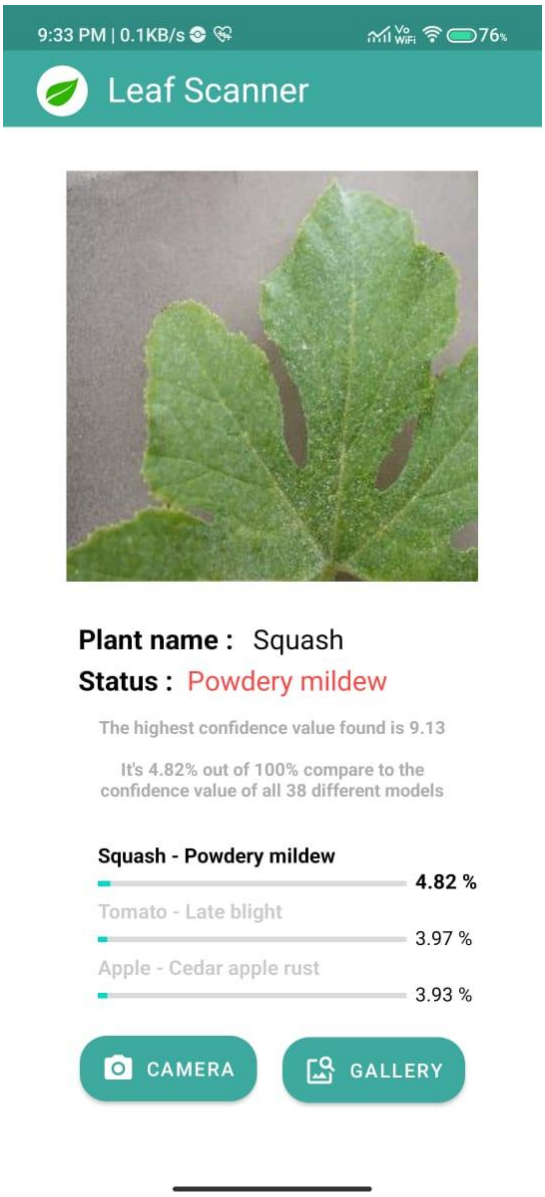


Figure 47: Test result of squash - powdery midlew

The image is identified correctly as squash - powdery mildew. Which the class has highest confidence value among all 38 classes.

Strawberry – Healthy

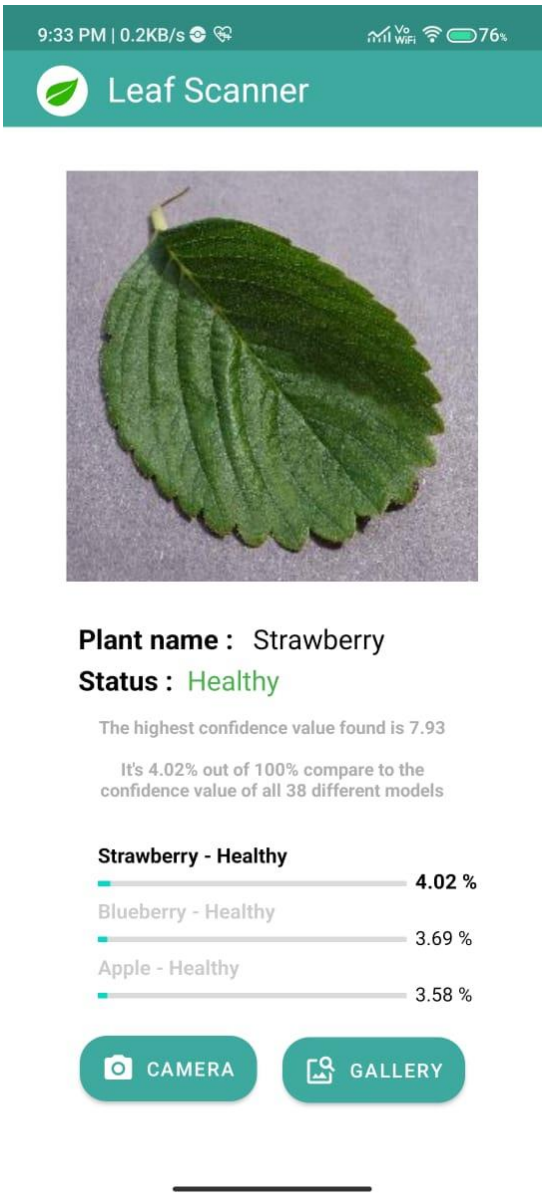


Figure 48: Test result of strawberry - healthy

The image is identified correctly as strawberry – healthy. Which the class has highest confidence value among all 38 classes.

Strawberry - Leaf scorch

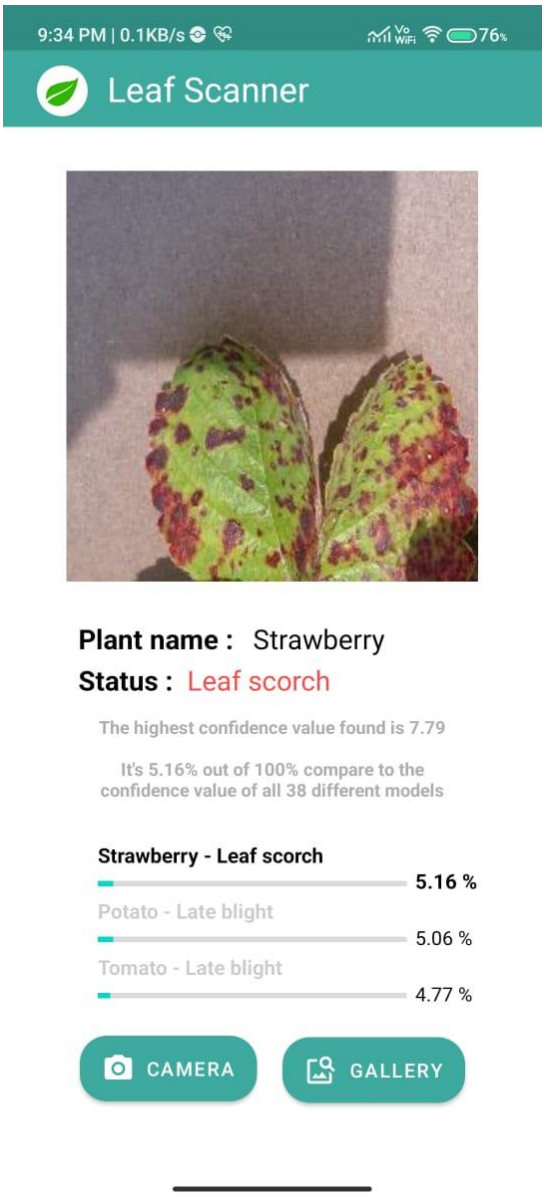


Figure 49: Test result of strawberry - leaf scorch

The image is identified correctly as strawberry - leaf scorch. Which the class has highest confidence value among all 38 classes.

Tomato - Bacterial spot

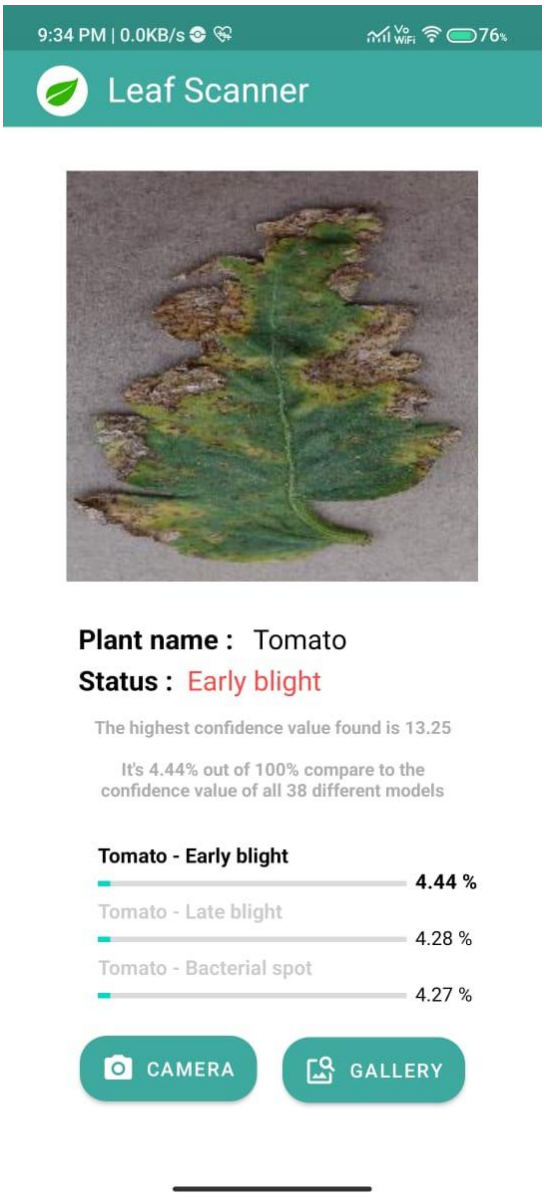


Figure 50: Test result of tomato - bacterial spot

The image identified wrongly as tomato – early blight, which the image is showing an image of tomato – bacteria spot. This is because it is high similarity of the image pattern and cause the confidence value of tomato – early blight is higher than tomato – bacteria spot.

Tomato - Early blight

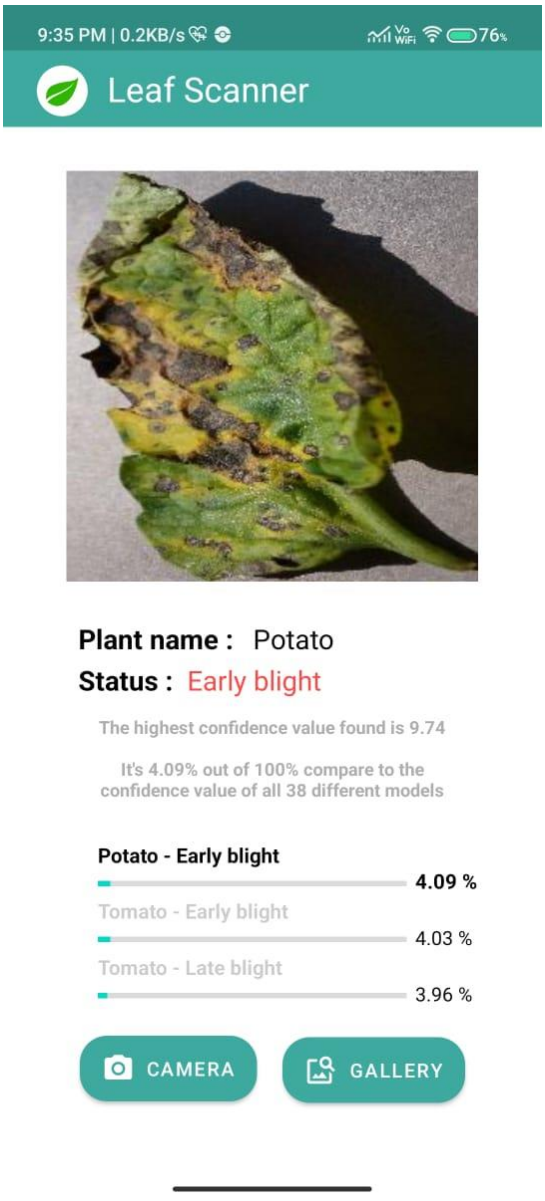


Figure 51: Test result of tomato - early blight

The image identified wrongly as potato – early blight, which the image is showing an image of tomato – early blight. This is because it is high similarity of the image pattern and cause the confidence value of potato – early blight is higher than tomato – early blight.

Tomato – Healthy

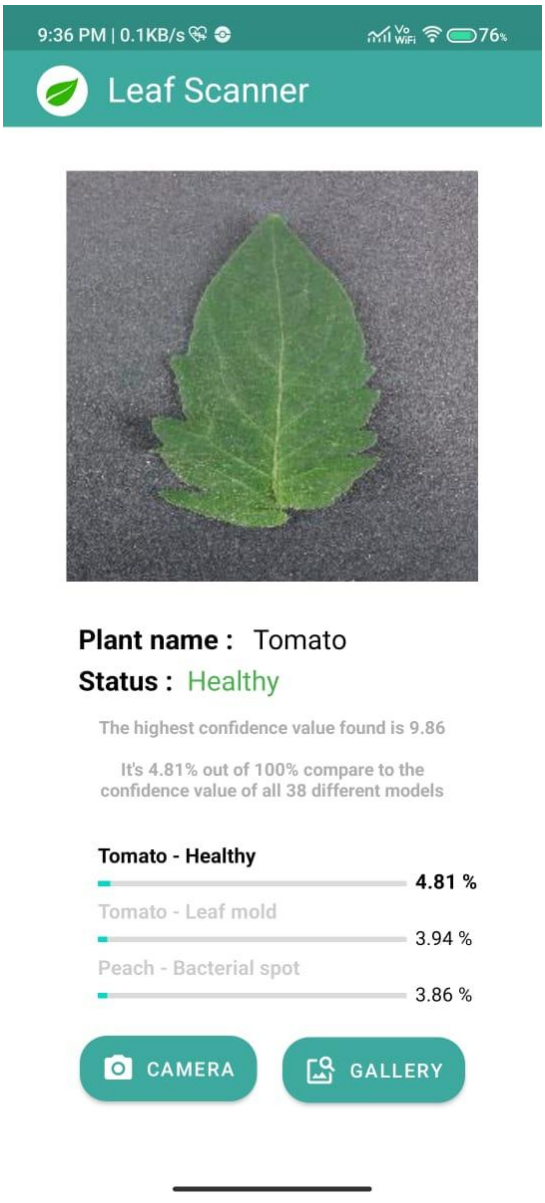


Figure 52: Test result of tomato - healthy

The image is identified correctly as tomato – healthy. Which the class has highest confidence value among all 38 classes.

Tomato – Late blight

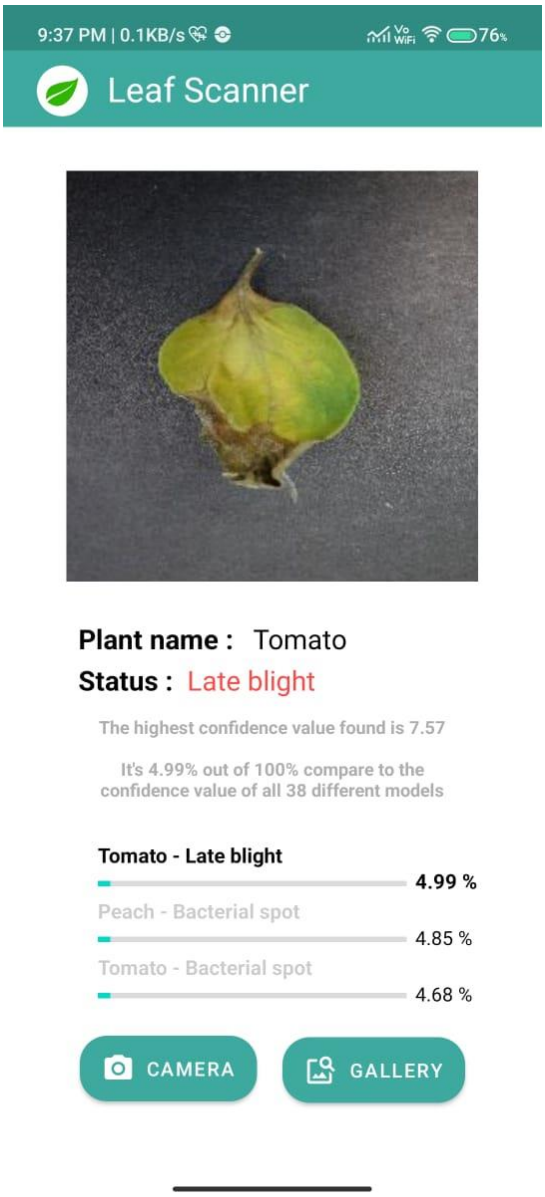


Figure 53: Test result of tomato - late blight

The image is identified correctly as tomato – late blight. Which the class has highest confidence value among all 38 classes.

Tomato – Leaf mold

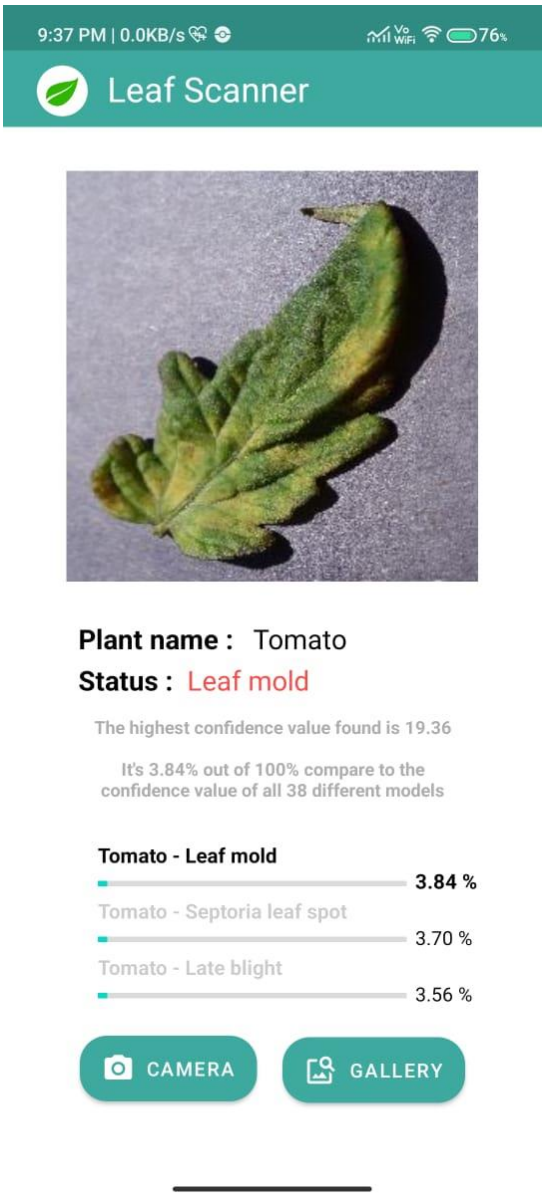


Figure 54: Test result of tomato - leaf mold

The image is identified correctly as tomato – leaf mold. Which the class has highest confidence value among all 38 classes.

Tomato – Septoria leaf spot

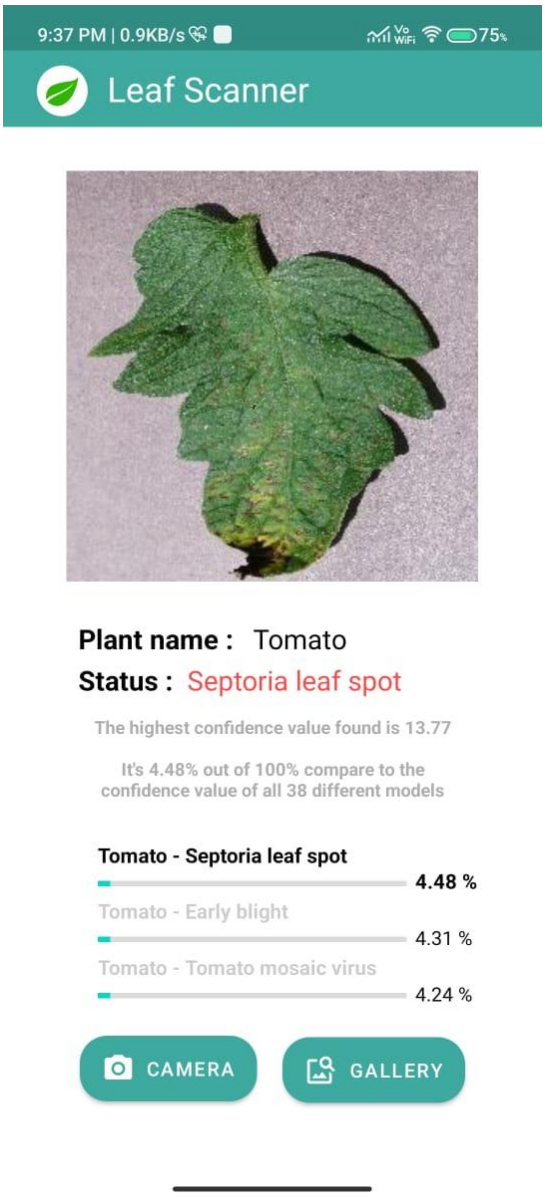


Figure 55: Test result of tomato - septoria leaf spot

The image is identified correctly as tomato – septoria leaf spot. Which the class has highest confidence value among all 38 classes.

Tomato – Spider mites two spotted spider mite

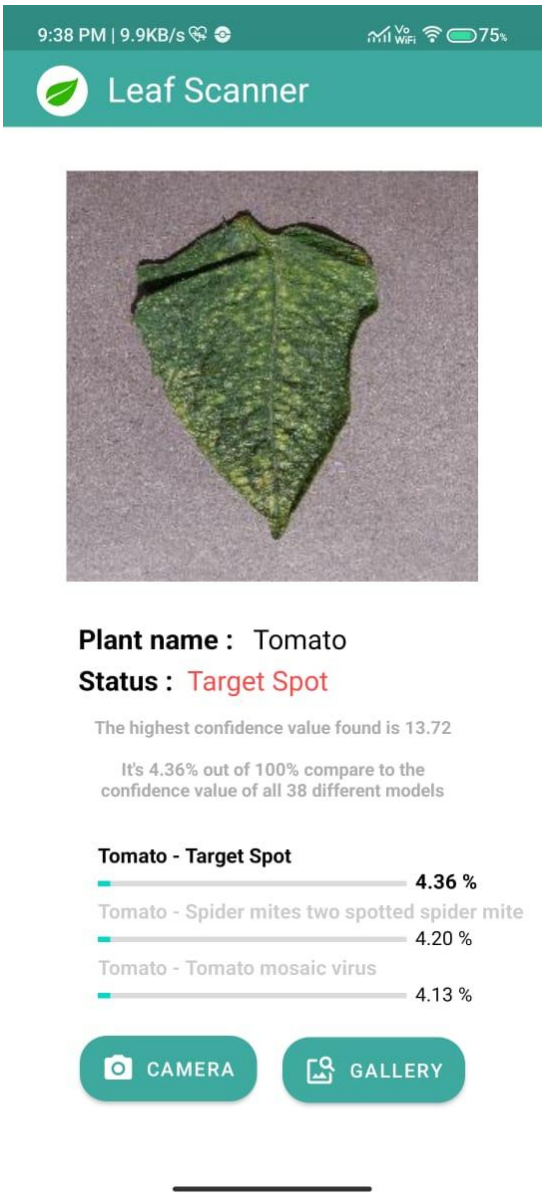


Figure 56: Test result of tomato - spider mites two spotted spider mite

The image identified wrongly as tomato – target spot, which the image is showing an image of tomato – spider mites two spotted spider mite. This is because it is high similarity of the image pattern and cause the confidence value of tomato – target spot is higher than tomato – spider mites two spotted spider mite.

Tomato - Target Spot

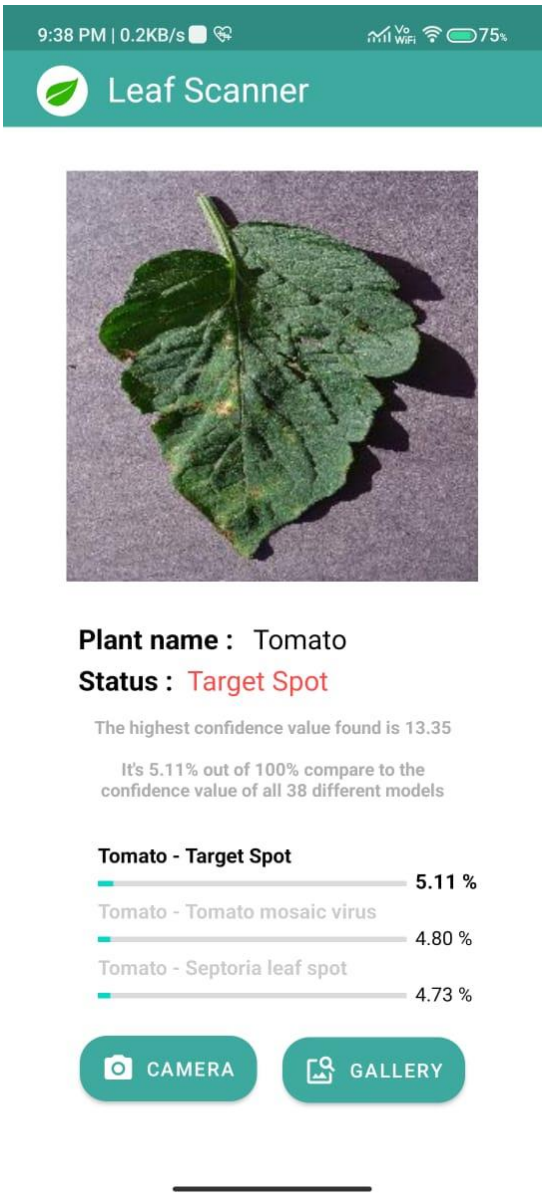


Figure 57: Test result of tomato - target spot

The image is identified correctly as tomato - target spot. Which the class has highest confidence value among all 38 classes.

Tomato - Tomato mosaic virus

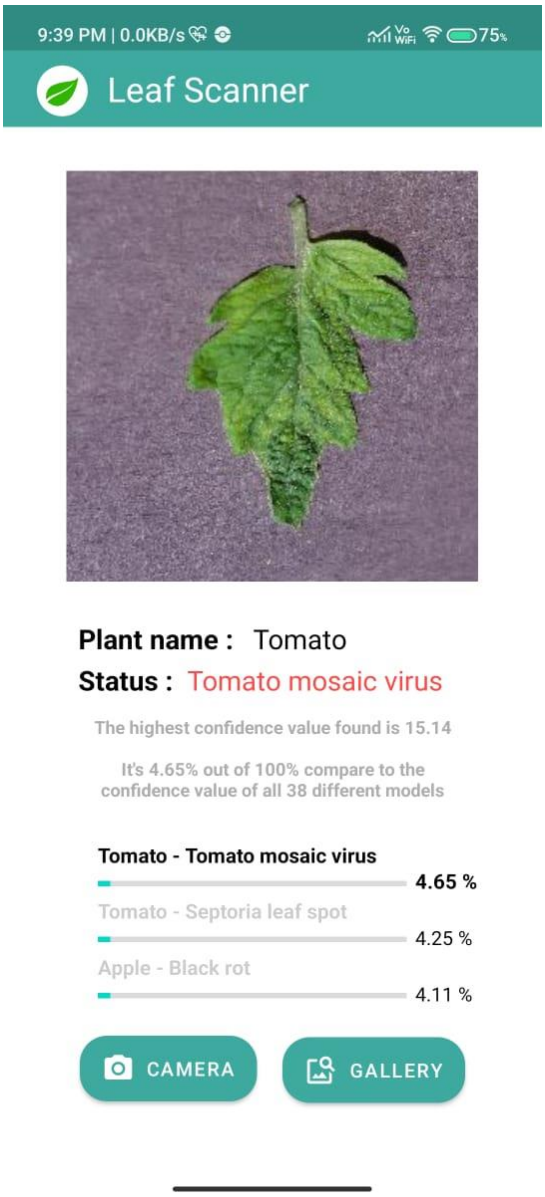


Figure 58: Test result of tomato - tomato mosaic virus

The image is identified correctly as tomato - tomato mosaic virus. Which the class has highest confidence value among all 38 classes.

Tomato - Tomato yellow leaf curl virus

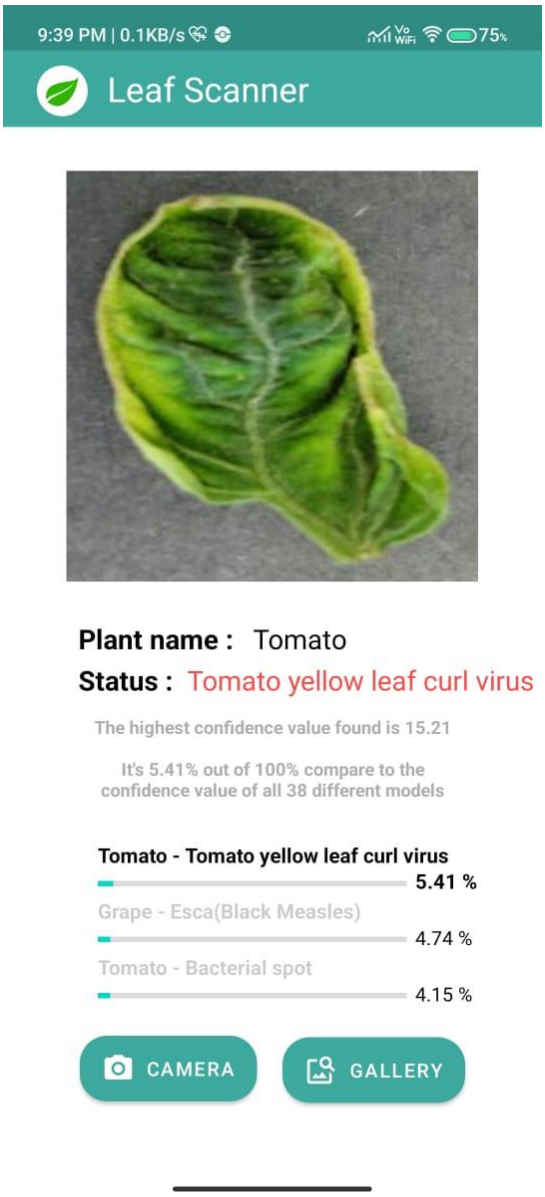


Figure 59: Test result of tomato - tomato tellow leaf curl virus

The image is identified correctly as tomato - tomato yellow leaf curl virus. Which the class has highest confidence value among all 38 classes.

Testing summary of round 1

Table 2: Testing summary of round 1

	Classes	Relevant images
1	Apple - Apple scab	Successfully recognised
2	Apple - Black rot	Successfully recognised
3	Apple - Cedar apple rust	Successfully recognised
4	Apple - Healthy	Recognised as **Apple - black rot**
5	Blueberry – Healthy	Recognised as **Tomato - target spot**
6	Cherry – Healthy	Recognised as **Potato - healthy**
7	Cherry – Powdery mildew	Successfully recognised
8	Corn - Cercospora leaf spot / Grayleaf spot	Successfully recognised
9	Corn - Common rust	Successfully recognised
10	Corn – Healthy	Successfully recognised
11	Corn - Northern leaf blight	Successfully recognised
12	Grape - Black rot	Recognised as **Grape - esca**
13	Grape - Esca(Black Measles)	Successfully recognised
14	Grape – Healthy	Successfully recognised
15	Grape - Leaf blight (Isariopsis leaf spot)	Successfully recognised
16	Orange - Haunglongbing(Citrus greening)	Successfully recognised
17	Peach - Bacterial spot	Successfully recognised
18	Peach – Healthy	Recognised as **Cherry – powdery mildew**
19	Pepper bell - Bacterial spot	Successfully recognised
20	Pepper bell - Healthy	Successfully recognised
21	Potato - Early blight	Successfully recognised
22	Potato – Healthy	Successfully recognised
23	Potato - Late blight	Successfully recognised
24	Raspberry - Healthy	Successfully recognised
25	Soybean – Healthy	Successfully recognised
26	Squash - Powdery mildew	Successfully recognised
27	Strawberry – Healthy	Successfully recognised
28	Strawberry - Leaf scorch	Successfully recognised
29	Tomato - Bacterial spot	Recognised as **Tomato - early blight**
30	Tomato - Early blight	Recognised as **Potato - early blight**
31	Tomato – Healthy	Successfully recognised
32	Tomato – Late blight	Successfully recognised
33	Tomato – Leaf mold	Successfully recognised
34	Tomato – Septoria leaf spot	Successfully recognised
35	Tomato – Spider mites two spotted spider mite	Recognised as **Tomato - target spot**
36	Tomato - Target Spot	Successfully recognised
37	Tomato - Tomato mosaic virus	Successfully recognised
38	Tomato - Tomato yellow leaf curl virus	Successfully recognised

Testing result of round 1 shows that the images can be successfully recognised, but it might be recognised wrongly because of high similarity of the image pattern.

4.1.2 Testing – Round 2

Round 2 re-test all the negative result from round 1 by input another related image.

Apple - Healthy

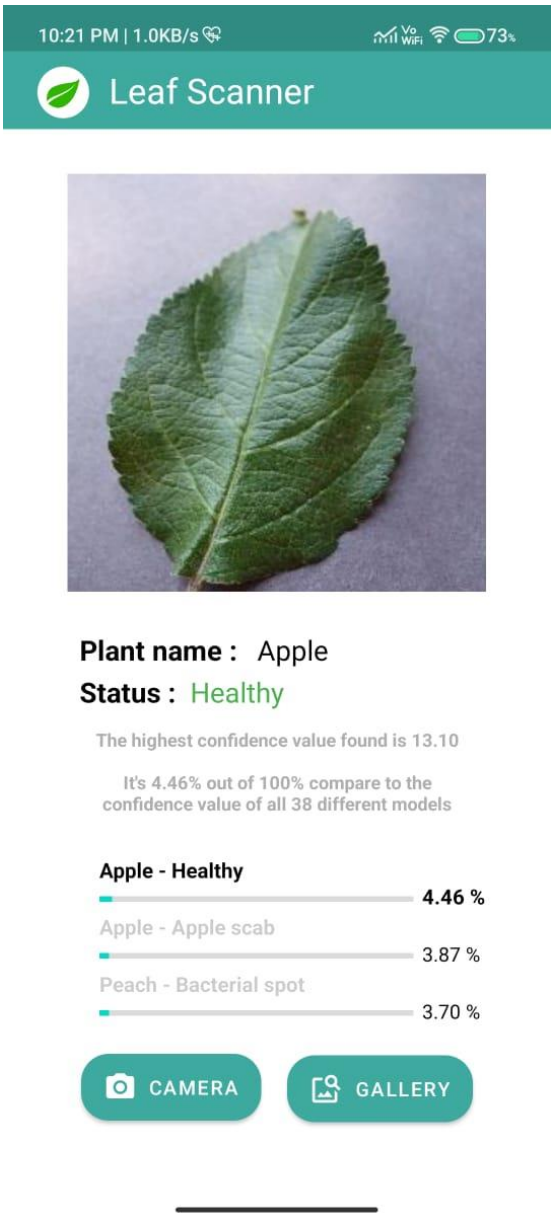


Figure 60: Test result of Apple - healthy 2

The image is identified correctly as apple - healthy. Which the class has highest confidence value among all 38 classes.

Blueberry – Healthy

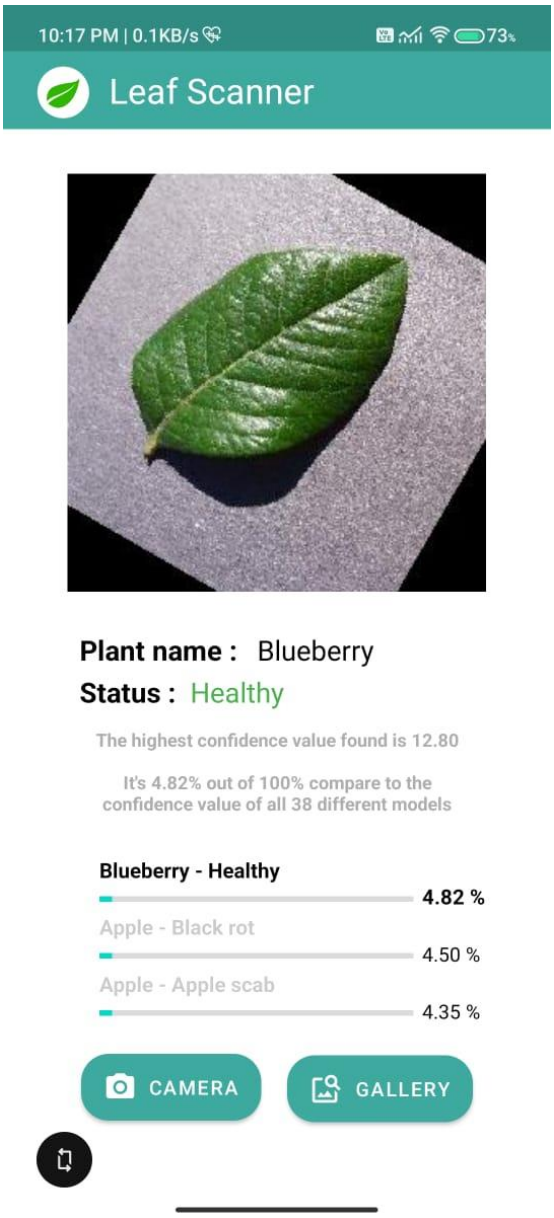


Figure 61: Test result of bluebrry - healthy 2

The image is identified correctly as blueberry – healthy. Which the class has highest confidence value among all 38 classes.

Cherry – Healthy

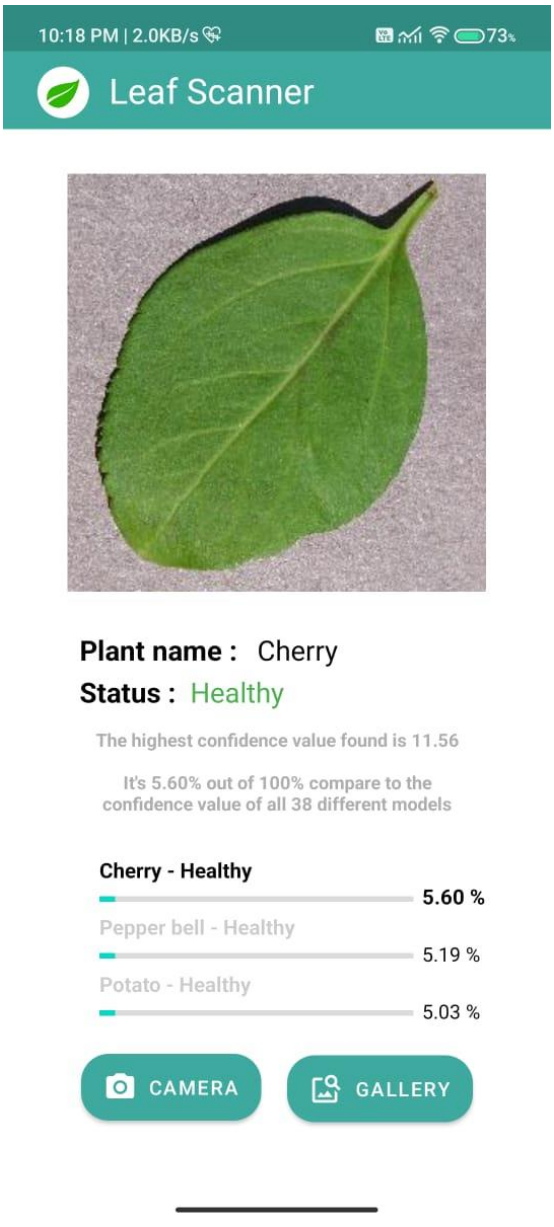


Figure 62: Test result of cherry - healthy 2

The image is identified correctly as cherry – healthy. Which the class has highest confidence value among all 38 classes.

Grape - Black rot

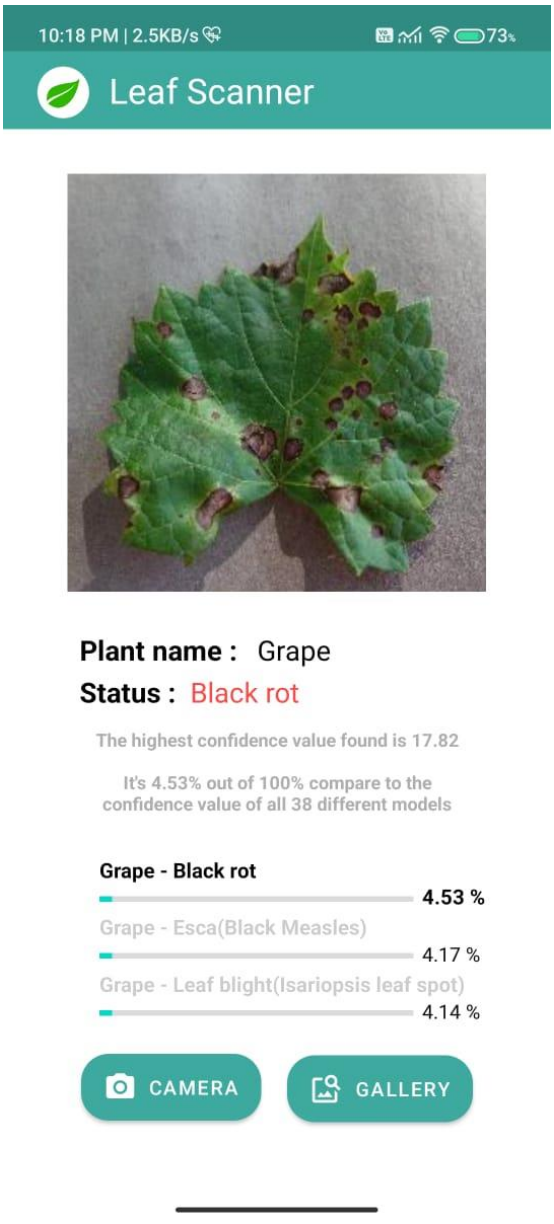


Figure 63: Test result of grape - black rot

The image is identified correctly as grape - black rot. Which the class has highest confidence value among all 38 classes.

Peach – Healthy

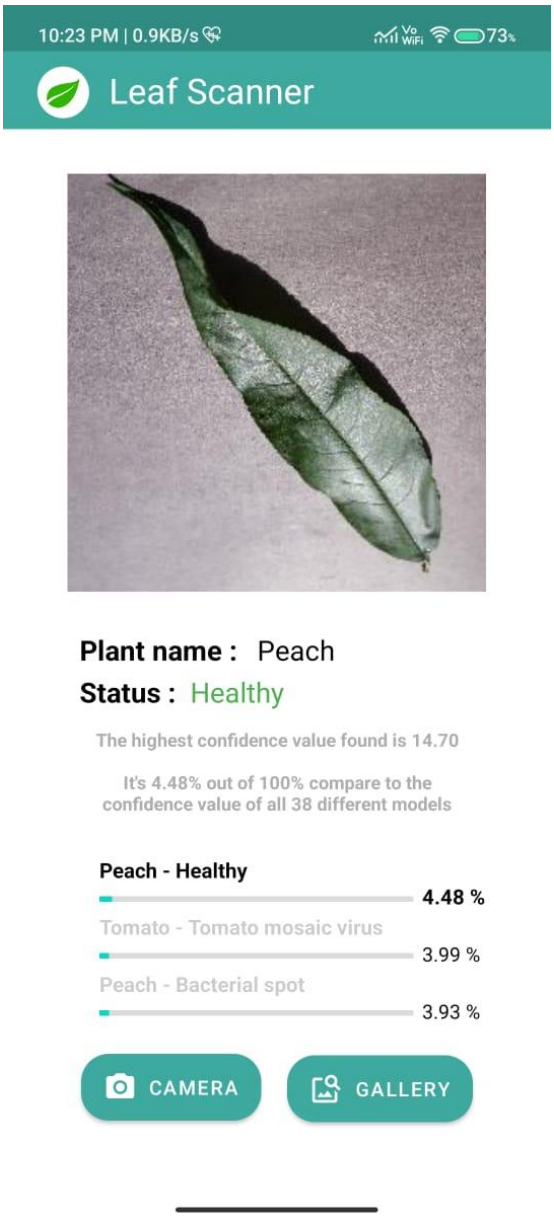


Figure 64: Test result of peach – healthy 2

The image is identified correctly as peach – healthy. Which the class has highest confidence value among all 38 classes.

Tomato - Bacterial spot

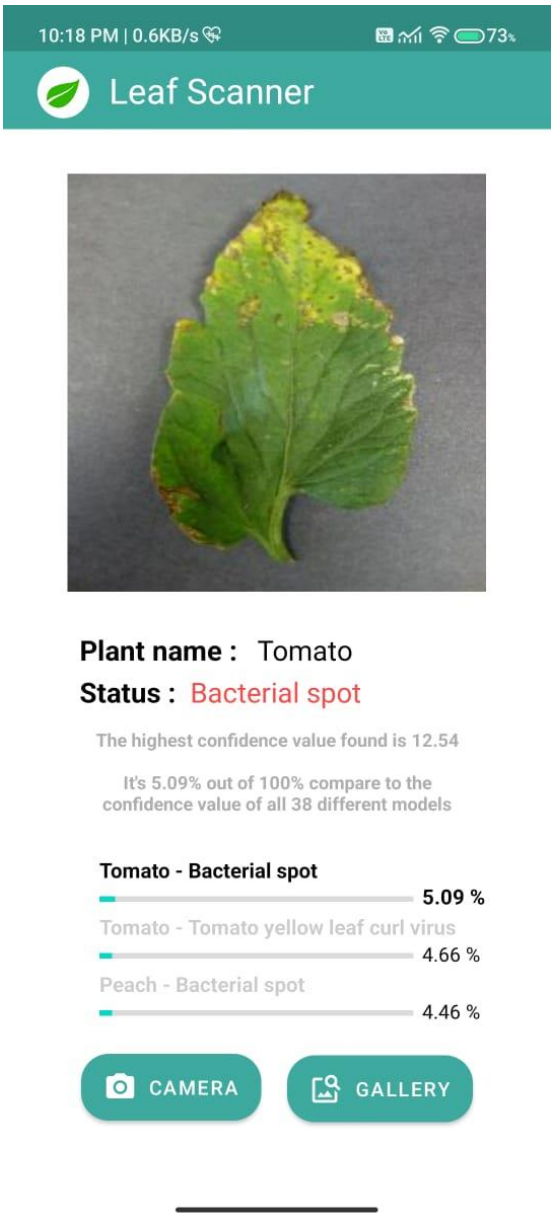


Figure 65: Test result of tomato - bacteria spot 2

The image is identified correctly as tomato - bacterial spot. Which the class has highest confidence value among all 38 classes.

Tomato - Early blight

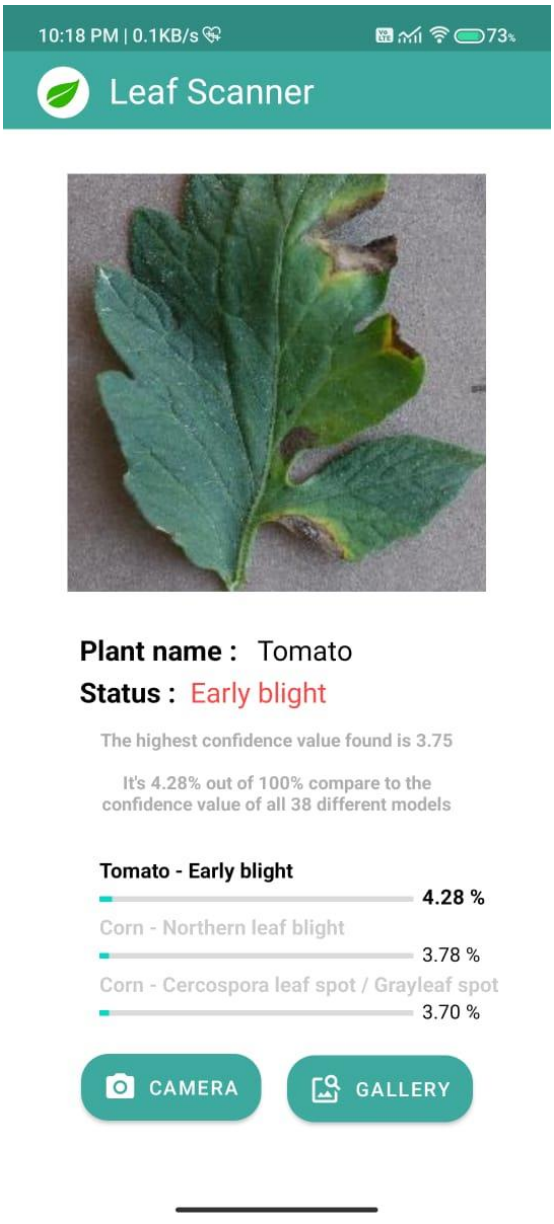


Figure 66: Test result of tomato - early blight 2

The image is identified correctly as tomato - early blight. Which the class has highest confidence value among all 38 classes.

Tomato – Spider mites two spotted spider mite

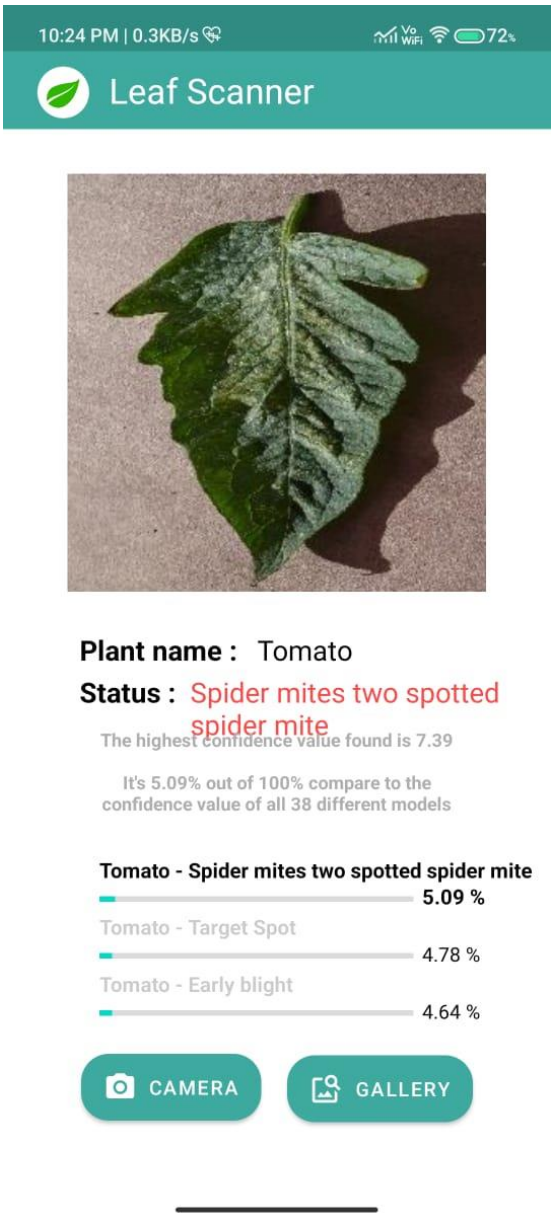


Figure 67: Test result of tomato - spider mites two spotted spider mite 2

The image is identified correctly as tomato – spider mites two spotted spider mite. Which the class has highest confidence value among all 38 classes.

Testing summary of round 2

Table 3: Testing summary of round 2

	Classes	Relevant images
1	Apple - Healthy	Successfully recognised
2	Blueberry – Healthy	Successfully recognised
3	Cherry – Healthy	Successfully recognised
4	Grape - Black rot	Successfully recognised
5	Peach – Healthy	Successfully recognised
6	Tomato - Bacterial spot	Successfully recognised
7	Tomato - Early blight	Successfully recognised
8	Tomato – Spider mites two spotted spider mite	Successfully recognised

Testing result of round 2 shows that the images still can be successfully recognised even though it failed at round 1 by replace the image to another.

4.1.3 Testing – Round 3

Round 3 perform testing by input irrelevant images.

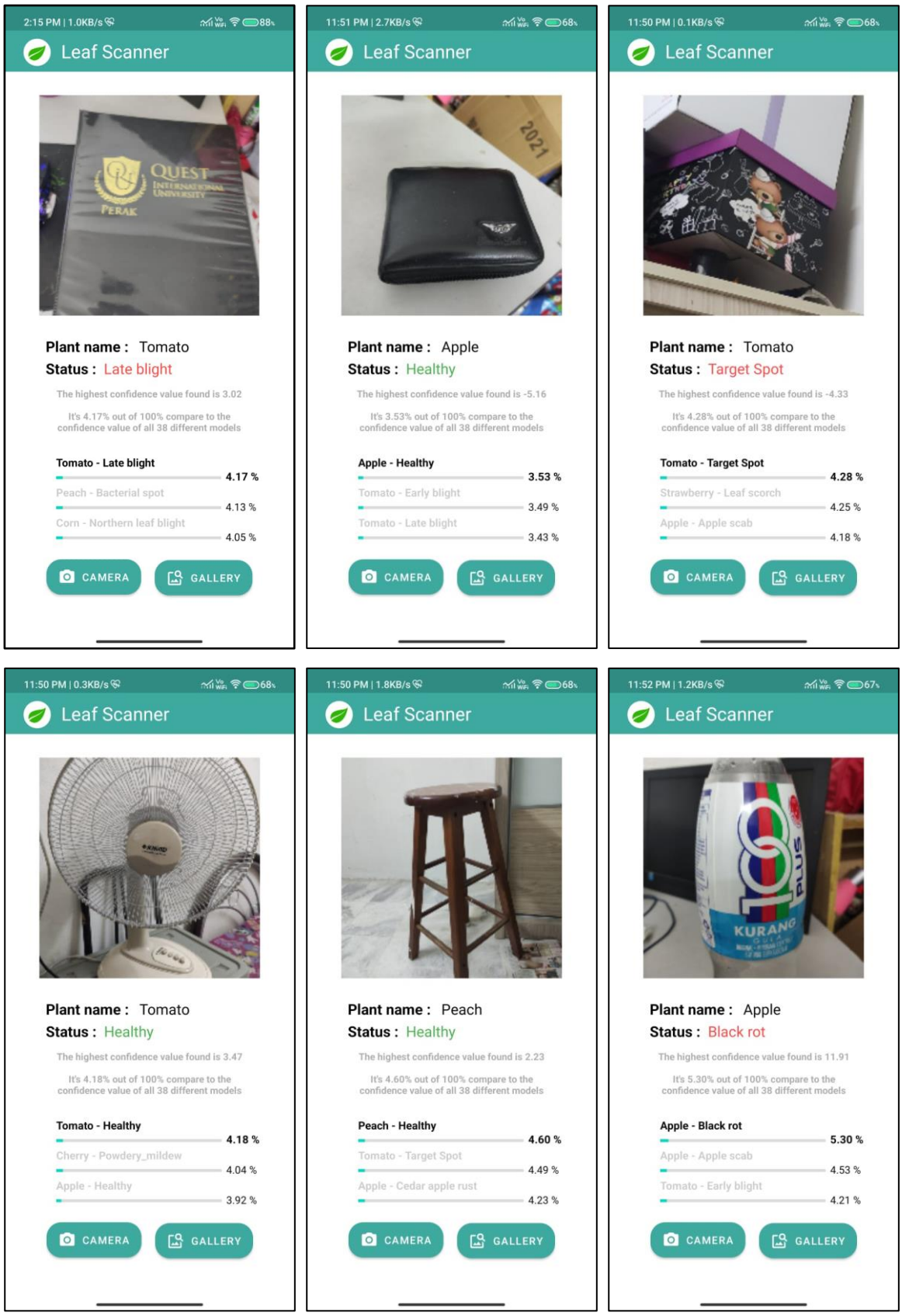


Figure 68: Six negative test case by input irrelevant images

Testing summary of round 3

Table 4: Testing summary of round 3

	Irrelevant images	Highest confidence value class
1	Document folder	Classified as Tomato – Late blight
2	Wallet	Classified as Apple - Healthy
3	Gift box	Classified as Tomato - Target spot
4	Stand fan	Classified as Tomato - Healthy
5	Stool	Classified as Peach - Healthy
6	Bottle	Classified as Apple – Black rot

The test result showing that it will still display one of the classes as output even if irrelevant image is inputted.

In summary, from all 3 rounds testing result shown at above, it shows that if the input images are within the class, it can be successfully recognised, but it might be recognised wrongly because of high similarity of the image pattern. For the image outside the class, which are the irrelevant images, it will be misclassified as one of the classes which have highest confidence value.

Chapter 5: Conclusion

5.1 Conclusion

In conclusion, for model building, it will need a ton of dataset of images that use to train a model. In this assignment, it uses CNN deep learning to generate the neuron network then followed by training process. Eventually, convert the trained model in form of TFlite, then export it to adopt it in the mobile application.

The essential part of the mobile application, Leaf Scanner, is load the trained model and read the images. Therefore, by getting the highest confidence value which the class that the image belongs to, it can show it as output. So, this is background process to perform the image classification.

From the tested result, it's clearly shows that the Leaf Scanner able to achieve the objective successfully which is the application that capable to detect and identify the type of disease of various type of plant is developed.

5.2 Strengths and weaknesses of the developed intelligent agent program.

Leaf Scanner can successfully classify the images which are within the classes. On the other hand, the processing time to perform image classification is fast, it not even need a second to classify the image. Next, it doesn't require internet connection to perform the function because the trained model is loaded into the mobile app.

The dataset that in the apps is limited, which means it is only able to detect up to 38 different plants with its health status. The image would be classified wrongly due to the similarity of image pattern compared to other classes. Apart from this, for the images that is irrelevant to the classes, it would be misclassified it to any one of the classes which have highest confidence value. The image accuracy will be affected by the factor such as brightness and image quality.

References

Choudhuri, A. (12 January, 2022). *Types of intelligent agent*. Retrieved from ProbeGroup: https://www.probegroup.com.au/blog/types-of-intelligent-agent

Deep learning for computer vision. (n.d.). Retrieved from Run Ai: https://www.run.ai/guides/deep-learning-for-computer-vision/tensorflow-cnn

MWITI, D. (n.d.). *CNN tensorflow*. Retrieved from cnvrg: https://cnvrg.io/cnn-tensorflow/

Shetty, S. (23 August, 2018). *Why TensorFlow always tops machine learning and artificial intelligence tool surveys*. Retrieved from Packt: https://hub.packtpub.com/tensorflow-always-tops-machine-learning-artificial-intelligence-tool-surveys/