Faculty of Computers &
Artificial Intelligence

Benha University

Rice Diseases Diagnosis

A senior project submitted in partial fulfillment of the requirements for the degree of Bachelor of Computers and Informatics.

Scientific Computing Departement,

Project Team

1- Aliaa Faisal Abd-Elmotaleb Kashwa

2- Amira Abdelghany AboElmoaty Nassar

3- Fatma Elzahraa Mahmoud Said Mahmoud

4- Sara Alaa Mohamed Ahmed Awad

5- Sanaa Rawy Mansour Mohamed Eissa

Under Supervision of

Dr. Eman Monier

Benha, July 20

J u l y – 2 0 2 1

ACKNOWLEDGMENT

We would like to thank Dr/ Eman Monier for the continuous support.

We Would Like to Thank Benha University for Supporting Us.

DECLARATION

We hereby certify that this material, which we now submit for assessment on the program of study leading to the award of Bachelor of Computers and Artificial Intelligence in Scientific Computing (SC) is entirely our own work, that we have exercised reasonable care to ensure that the work is original, and does not to the best of our knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of our work.

Signed:

____

____

____

____

____

Date: 13/ 07/ 2021.

ABSTRACT

Rice is of great importance among food crops in the world, and rice occupies the most important main grain crop in Egypt, but there are some diseases that may affect the crop and negatively affect the quantity of the crop in relation to the economy of the country, so discovering these diseases early may help to preserve the entire amount of the crop, increasing production and national income from this crop.

Our goal is to provide a solution to discover some diseases that affect the rice plant, so we made a mobile application that works to discover the rice diseases by inserting the image of the rice and making some operations and algorithms using Tensorflow kares model using the MobileNet algorithm to discover if it is infected with any disease or healthy and determine the type of disease for a rice paper, for example, the BrownSpot, Hispa and Leaf Blast.

Also, our application includes some pages of the definition and the treatment for each disease.

In the end, our goal is to provide a disease-free plant and to preserve a large amount of the plant without harm to have the most benefits from it.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

ACRONYM	Definition of Acronym
API	Application Programming Interface

C h a p t e r   O n e

1 INTRODUCTION

1.1 DEFINATION:

The research paper works on developing new techniques in the rice plant to improve the yield and help farmers obtain increased rice production. Rice is one of the main crops that people depend on for food on a daily basis. The lack of the crop leads to an imbalance in the food system, in addition to the rewarding economic return for the farmer, but with the emergence of pests and diseases affecting plant health and crop quantity, but disturbance and plant infections remain a predominant threat to the overall plant health and nutrition. With the current rate of population development.

Rice diseases cause great damage in agriculture, resulting in significant yield losses. With the significant population increase observed in the world, it was necessary to preserve agricultural crops from rice diseases that often lead to the loss of the entire crop. Rice diseases annually destroy more than 20% of the five major crops that provide half of the calories on which the world's population depends.

Among the most important diseases that affect the rice plant, which cause damage to the crop and the general production of rice, are Brown Spot disease, Hispa , Leaf Blast, Smut and nematode white leaf tip disease.

• This pictures shows the effect of plant diseases on Rice Yield:

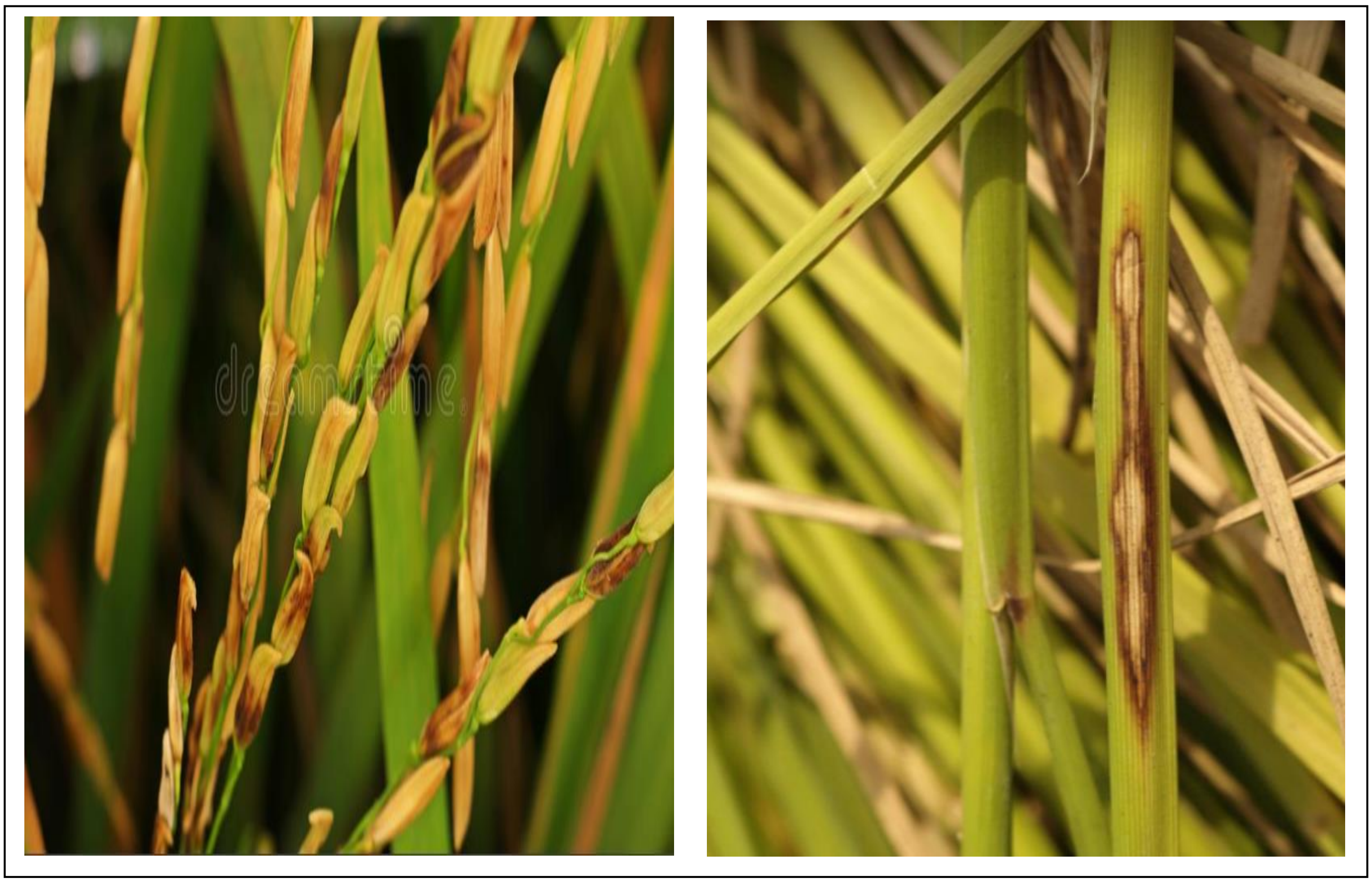


Figure 1 infected Rice Crop

1.2 DISEASES:

1.2.1 Brown spot:

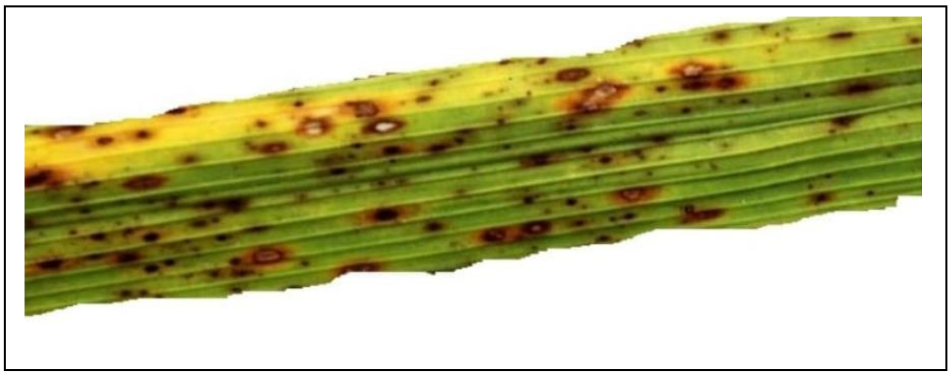


Figure 2 Ex of Brown Spot

Rice Brown spot is a fungal disease caused by the fungus Cochliobolus miyabeanus. It appears in the form of brown spots the size of the head of a match stick on the leaves. These spots appear on the grains, distorting their appearance.

To avoid the occurrence of disease we are working on Use fungicides (e.g., iprodione, propiconazole, azoxystrobin, trifloxystrobin, and carbendazim) as seed treatments. Treat seeds with hot water (53−54°C) for 10−12 minutes before planting, to control primary infection at the seedling stage.

1.2.2 Hispa:

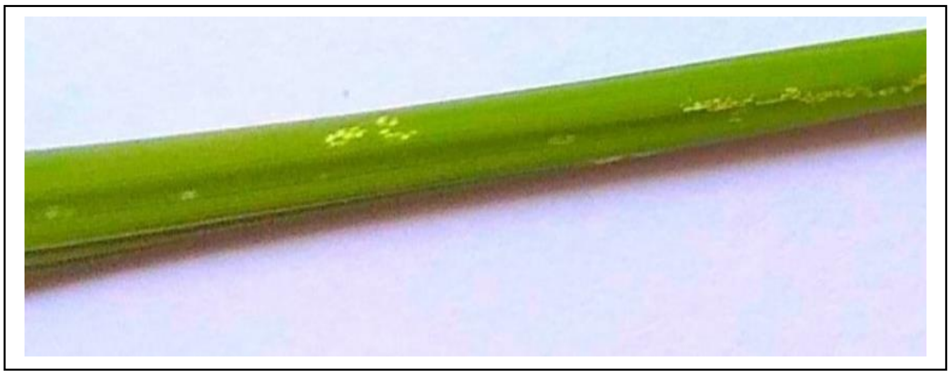


Figure 3 example of Hispa

Rice Hispa is a very serious insect pest of rice, particularly in the Terai region of Nepal. The adult is a small bluish black beetle, measuring 5 mm in length and is recognized by numerous short spines on the body and forewings. Adults scrape the areas between the veins giving a characteristic appearance of white parallel streaks along the leaf.

Avoid over fertilizing the field.

1. Close plant spacing results in greater leaf densities that can tolerate higher Hispa numbers.
2. Leaf tip containing blotch mines should be destroyed.
3. Manual collection and killing of beetles – hand nets.
4. To prevent egg laying of the pests, the shoot tips can be cut.
5. Clipping and burying shoots in the mud can reduce grub populations by 75 - 92%.
6. Reduviid bug eats upon the adults.
7. Spraying of methyl parathion 0.05% or Quinalphos 0.05%

1.2.3 Leaf Blast:

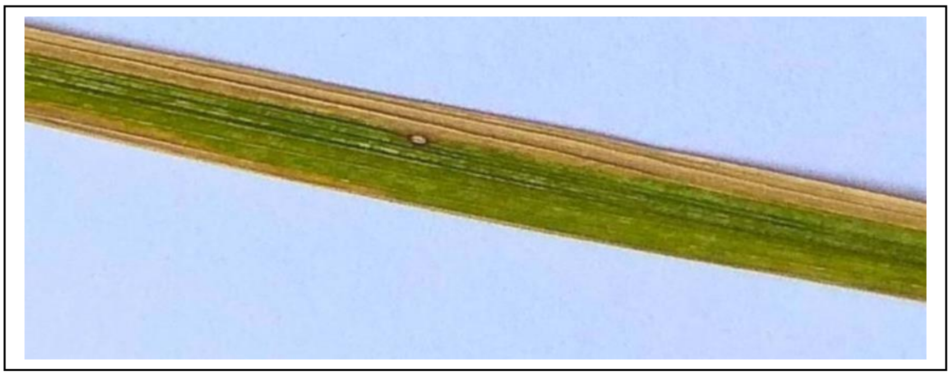


Figure 4: Ex of Leaf Blast

Rice blast, also known as rotten neck, is caused by the fungal pathogen Pyricularia grisea. Like most fungal diseases, rice blast fungus rapidly grows and spreads in warm, humid weather. Because rice is usually grown in flooded fields, humidity is hard to avoid. On a warm, humid day, just one rice blast lesion can release thousands of disease causing spores into the wind. The lesion can keep producing thousands of spores each day for up to twenty days. All these spores fly on even the gentlest breeze, settling on and infecting damp and dewy rice plant tissues. Rice blast fungus can infect rice plants in any stage of maturity.

The best practices for preventing rice blast is to keep rice fields flooded deeply with a continual flow of water. When rice fields are drained for various cultural practices, a higher incident of fungal disease results. Rice blast treatment is done by applying fungicides at precise times of the plant's development. This usually is early in the season, again as plants are in the late boot phase, then again as 80-90% of the rice crop has headed.

1.3 PROBLEM STATEMENT:

We aim to save rice through the use of many important techniques for early and rapid detection of diseases. To make it easier to treat, it is one of the most important problems that affect the field of agriculture because rice consider one of the important crops that countries depend on.

1.4 SCOPE:

Our document Illustrates all details of the process of diagnosing plant diseases of Rice plants, so starting with the group of images of plant leaves, including the infected and the healthy ones, after which we applied a set of deep learning algorithms to classify and analyze the previously collected images, the algorithms outputs were compared. And choose the most accurate one, and then the output of the selected model was linked to the application a phone designed with flutter.

Our application includes details about each of 3 Diseases (Leaf Blast, Hispa, and Brown Spot) to know more about each of them

1.5 DELIVERABLES:

- ❖ Software devices:
    - ▪ Software Model tensorflow keras implemented by python
    - ▪ A mobile Application by using flutter
- ❖ Hardware devices: It does not use any device

1.6 MOTIVATION:

Because the agricultural crops are very important for the future of humanity in all world so rice diseases is very dangerous and has a short and long-term impact for us.

It is important to provide a solution to it so we aimed to help for this problem, as early discovering rice diseases for increasing the quantity and quality.

1.7 IMPACT OF THE PROJECT:

The value of our data is huge and can provide great insight and facts for users.We believe that rice diseases diagnosis using tenser flow kares model would allow Organizations to be much more responsive to their customer, leading to happier users and more successful product. The data comes from users, and it's expected to make it easier to doctor and other users to get the result.

1.7.1 Expected impact in end users:

End users include campaign managers, Product designers, and more. Users will can easily capable of exploring opinions about a certain topic in a time period or a geographical location, with as little effort.

1.7.2 Expected impact in market:

Manufacturers and producers now have easy methods of exploring what their customers want, like or dislike about them. Since the market considered to be more responsive to customers.

1.7.3 Expected impact in Community:

The community may not feel the impact directly. Opinions will make their way to the decision makers much more swiftly, so customers will be happier and product more successful products.

1.8 ORGANIZATION OF THE PROJECT REPORT:

Our document will provide all the aspect of a rice diseases diagnosis system, with all the tools and techniques.

Chapter 1: shows brief introduction of our project, problem statement, scope, deliverables and impact.

Chapter 2: shows Literature Survey.

Chapter 3: shows the Analysis of the project.

Chapter 4: shows the design of the project.

Chapter 5: shows the development steps as we went through them.

Chapter 6: shows Backend implementation of our system.

Chapter 7: shows frontend implementation.

Chapter 8: shows conclusion.

Chapter 9: shows the future work.

Chapter Two

2 SURVEY

Before we begin to describe the design of the project, we had to research and discover what researchers and decision makers have reached in this subject, to start from where the others ended and look at their work so that helps us understand the idea more and get the most possible benefit from their work and direct us to the tools that we have to collect without having to search for them and enables us to develop with it as much as possible.

1. Classification of Rice Leaf Diseases Based on Morphological Changes

Aim: Developed for detecting two different types of rice diseases. In the first stage, uninfected and the diseased leaves are classified based on the number of peaks in the histogram. In the second level Bayes' classifier and SVM are applied to classify the leaf diseases

Achievement: The system has been validated using 1000 test spot images of infected rice leaves collected from the field, gives 79.5% and 68.1% accuracies for Bayes' and SVM Classifier based system respectively.

2. Detection and classification of rice plant diseases

Aim: Detection and classification of rice diseases based on the images of infected rice plants. This prototype system is developed after detailed experimental analysis of various techniques used in image processing operations. That consider three rice plant diseases namely Bacterial leaf blight, Brown spot, and Leaf smut. That capture images of infected rice plants using a digital camera from a rice field.

Achievement: We use Support Vector Machine (SVM) for multi-class classification. We achieve 93.33% accuracy on training dataset and 73.33% accuracy on the test dataset. We also perform 5 and 10-fold cross-validations, for which we achieve 83.80% and 88.57% accuracy, respectively.

3. Journal of critical reviews automatic rice plant disease recognition and identification using convolutional neural network

Aim: Rice crop disease detection & recognition in deep learning by using rice images is a signal in the field of agriculture.

Achievement: Improvement to the proposed method, in future one can implement an autoencoder instead of manually reducing image size. It can compress data without losing the important features, because autoencoders can regenerate up to 90% of the original images.

4. Identification and recognition of rice diseases and pests using convolutional neural networks

Aim: Experimental results show the effectiveness of these models with real datasets. Since large scale architectures are not suitable for mobile devices, a two-stage small CNN architecture has been proposed, and compared with the state-of-the-art memory efficient CNN architectures such as MobileNet, NasNet Mobile and SqueezeNet.

Achievement: Experimental results show that the proposed architecture can achieve the desired accuracy of 93.3% with a significantly reduced model size (e.g., 99% smaller than VGG16).

5. Vision Based Detection and Classification of Disease on Rice Crops Using Convolutional Neural Network

Aim: presents a combination of K-mean s clustering, SVM and CNN based classification techniques for the crop disease / infection with target as Rice crop. Agricultural crops show variation in features of healthy crops in color, texture and growth. Use of image processing algorithms involving K-means clustering, Support vector machine and Convolutional Neural Networks enables classification process.

Achievement: The supervised classification techniques involve prior training process which determines the accuracy. The training dataset involved initially with 10 image set which was increased to over 50 % to obtain accuracy over 95 %.

6. Plants Disease Identification and Classification Through Leaf Images: A Survey

Aim: proposed classification on infection and scientific scenarios in various instances and whether detection of diseases is carried out. The classification includes unsupervised and supervised techniques for rice plants; self-organizing map neural network (SOM-NN) is deployed to find out brown spot diseases and rice blast diseases.

Achievement: The classification achieved the highest, around 97.20%; the classification accuracy of Bayes being 79.5%, SVM (88.1%), PNN (97.76%), and KNN (93.33%). Cons: Out of 60 images, only 50 images were detected accurately.

7. A survey on detection and classification of rice plant diseases

Aim: proposed a system to detect diseases occurring in rice plant diseases such as brown spot, bacterial leaf blight, and leaf smut. The paper surveys various pre-processed images and ML techniques useful for the identification of diseases in rice plants. 145 images are considered out of which 30 images belong to the healthy class, 25 rice images belong to brown spot class, 46 belong to bacterial leaf blight, and the remaining 44 belong to leaf smut class, respectively.

Achievement: With the help of backpropagation NN, total accuracy of 74.2% is achieved just by considering the image features. The paper provides an insight into rice disease detection using the image processing technique.

8. Using Deep Learning Techniques to Detect Rice Diseases from Images of Rice Fields

Aim: convolutional neural network (CNN) was applied to detect and identify diseases in images. We studied 6 varieties of major rice diseases, including blast, bacterial leaf blight, brown spot, narrow brown spot, bacterial leaf streak and rice ragged stunt virus disease. Our studied used well known pre-trained models namely Faster R-CNN, RetinaNet, YOLOv3 and Mask RCNN, and compared their detection performance. The database of rice diseases used in our study contained photographs of rice leaves taken from fields of planting areas.

Achievement: We conducted experiments to train and test each model using a total of 6,330 images. The experimental results showed that YOLOv3 provided the best performance in term of mean average precision (mAP) at 79.19% in the detection and classification of rice leaf diseases. The precision obtained from Mask R-CNN, Faster R-CNN, and RetinaNet was at 75.92%, 70.96%, and 36.11%, respectively.

9. A Lightweight CNN Architecture to Identify Various Rice Plant Diseases in Bangladesh

Aim: For this research, we have collected a total 12 different types of rice disease images. The images has been pre-processed and augmented using different algorithms. Along with different state of the art CNN architectures, a lightweight CNN architecture has been proposed for identifying various rice plant diseases.

Achievement: The experimental result shows that our proposed model can identify the rice plant diseases with a mean validation accuracy of 95.4%. Considering small parameter size, it is evident that our proposed CNN model performs significantly well in detecting various rice plant diseases accurately.

Chapter Three

3 ANALYSIS

In order to make the system useful, we make it perform some tasks, such as showing results when necessary, performing analyses on plants, and showing a specific file for ease of understanding, and this needs great speed and presenting the results in a more accurate manner. Provide programming use by providing API.

Many solutions have been achieved with better performance, and we strive to provide the best and most comprehensive solutions for all users to build the system.

3.1 USERS TYPES

Table 1: System Users:

SYSTEM USERS:	**Description**
1- simple user	1- Personal interest 2- Don't understand advanced analysis because they not professional.
2- Developers	1- Depend on API usage 2- Interested in functions that provided by system.
3- professional analysts	1- Professional interest 2- Upload their own analysis data set

	3- Depend on advanced analysis
4- Data scientists/Data analysts	1- Professional interest 2- Direct access to metadata 3- More specific analysis function for programmatic methods. 4- The system consider as a database system.

3.2 SYSTEM REQUIREMENTS:

The system provides a useful analysis to understand the mechanism of action for users. In order for this analysis to be useful, some rules must be applied:

- After using a large set of data and training it using the model, it led to:
- The system must distinguish between healthy rice images and diseased images.
- Shows the user the image to be downloaded to discover whether the image has a disease or not.
- After previewing the image, if the image is healthy, it appears to the user that it is healthy, does not have disease and does not join any type of disease.
- Show the name of the disease if the image inside the system contains a disease of one of the diseases.
- Analysis of the system after the picture of the disease and determine the type of disease, there must be some suggestions for the prevention of disease (treatment of the disease).

- After the appearance of the result of the disease and its type and the discovery of the treatment for each disease, these results must be saved while the user uses them again and see the results or share them with another person.
- Therefore, the system must be easy to use to provide the appropriate methods for solving in the simplest ways and be compatible with the appropriate standards and work on the speed of extracting the result with the validity of the result.

3.3 FUNCTIONAL REQUIREMENTS:

The services provided by the mobile program that diagnoses the rice plant, as it allows the following:

Table 3-2: Functional Requirements:

users	1. User 2. Developers 3. Analysis
Processing	1. App provide users to upload images (rice image) from gallery. 2. It checks if the image is healthy or has a disease. 3. App available to anyone that can use to detect rice diseases.

3.4 NON-FUNCTIONAL REQUIREMENTS:

Table 3-3: Non-Functional Requirements:

Availability	The application will be available of time.
Reliability	Most people own mobile phones and know how the idea of this application seeks to help them get clear results for what they need and know the health of their plants easily.
Usability	The interface of application need to be friendly user interface to be easily with users.
Security	Database can be edited only by developers.

C h a p t e r   F o u r

4 SYSTEM DESIGN

Although the mobile application that we have done is very limited that there are many ways to design the application and increase it, but with that there are many questions that must be answered, for example, answering whether the application can be implemented on web platforms or not? And whether the application can be used for more than one programming language or not?

We chose to build this application using the mobile application for its ease of use and to be available to all users and because it is the best for our system to obtain a background service (insert the image to discover whether the image is a healthy rice plant or not) and a front-facing through which it is possible to identify the type of disease easily and show the name of the disease under the image Easily, but we can use a mobile website can be implemented by linking it to the mobile, and the web can be used later. Also, the mobile can be applied in more than one language and link it to the mobile.

Following is a context Diagram of the system. While not the most comprehensive diagram. We will then expand it into a full data flow diagram and explain its functions. The data flow diagram is then used as a reference in the implementation chapters to guide us to what functions need to be implemented.

4.1 USE CASE DIAGRAM:

A use case is a software and system engineering term that describes how a user uses a system to accomplish a particular goal. A use case acts as a software modelling technique that defines the features to be implemented and the resolution of any errors that may be encountered.

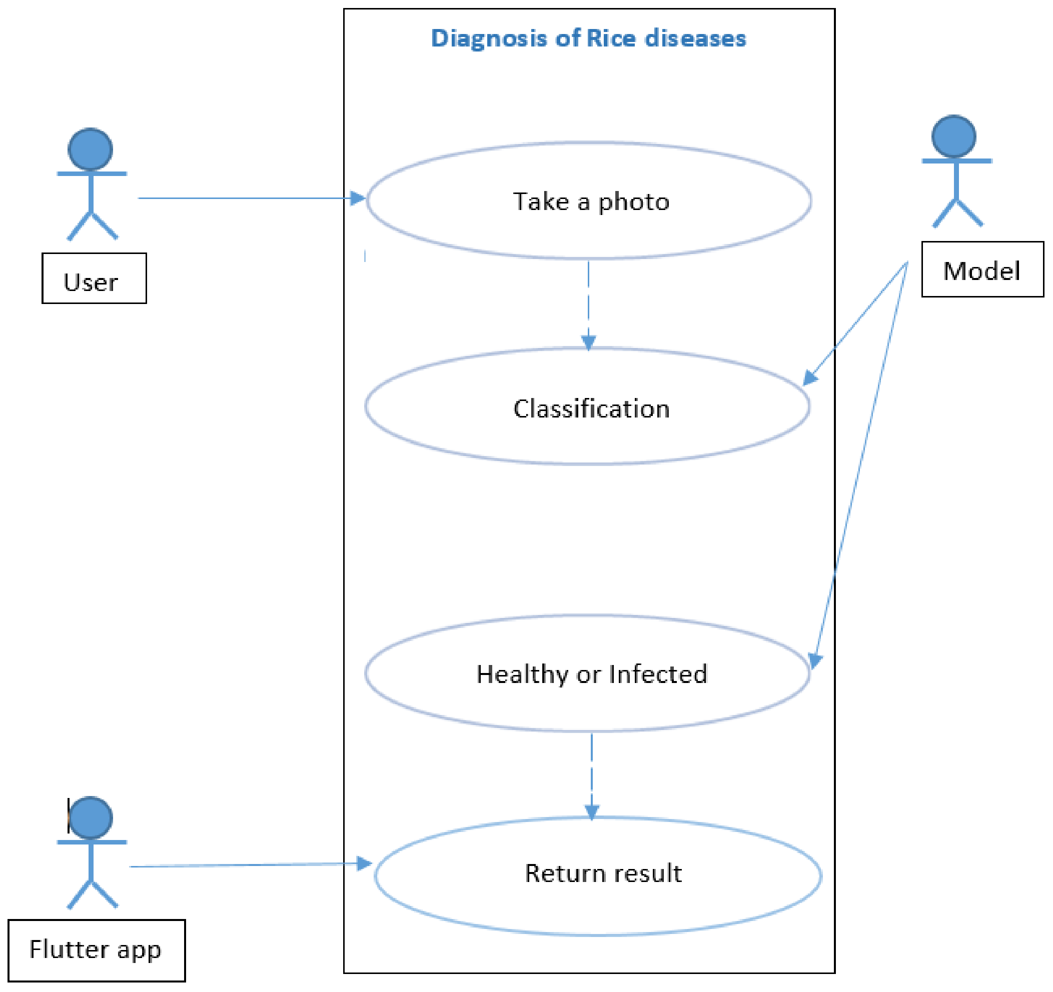


Figure 5 use case diagram

4.2 DATAFLOW DIAGRAM:

Data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system).

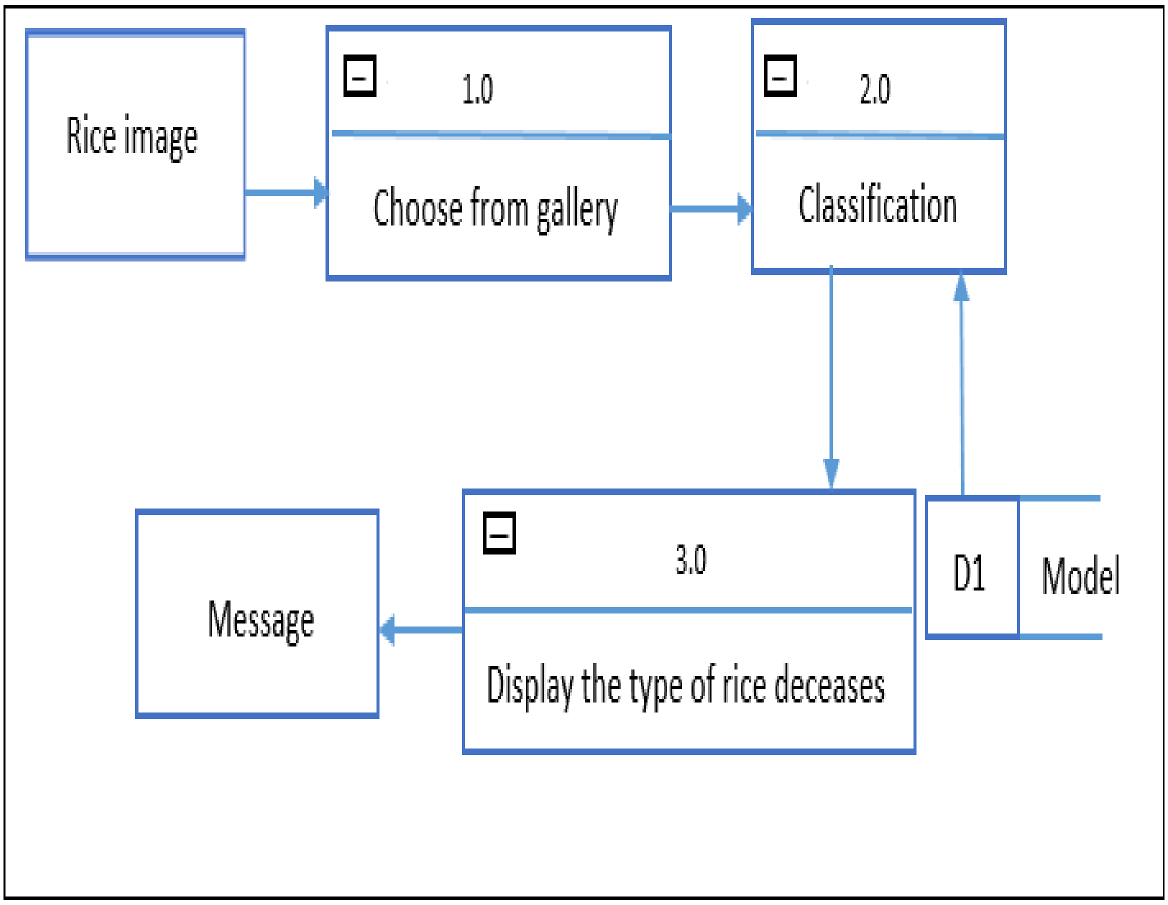


Figure 6 dataflow diagram

Table 4-2 dataflow diagram:

	Process	Description
1	Choose image from gallery	Enter image (rice diseases) in flutter app
2	Classification	Our model classifies the choosing image
3	Display the type of rice diseases	Display massage that show the rice type of diseases.

4.3 CLASS DIAGRAM:

Shows static structure of classifiers in a system Diagram provides basic notation for other structure diagrams prescribed by UML. Helpful for developers and other team members too Business Analysts can use class diagrams to model systems from business perspective.

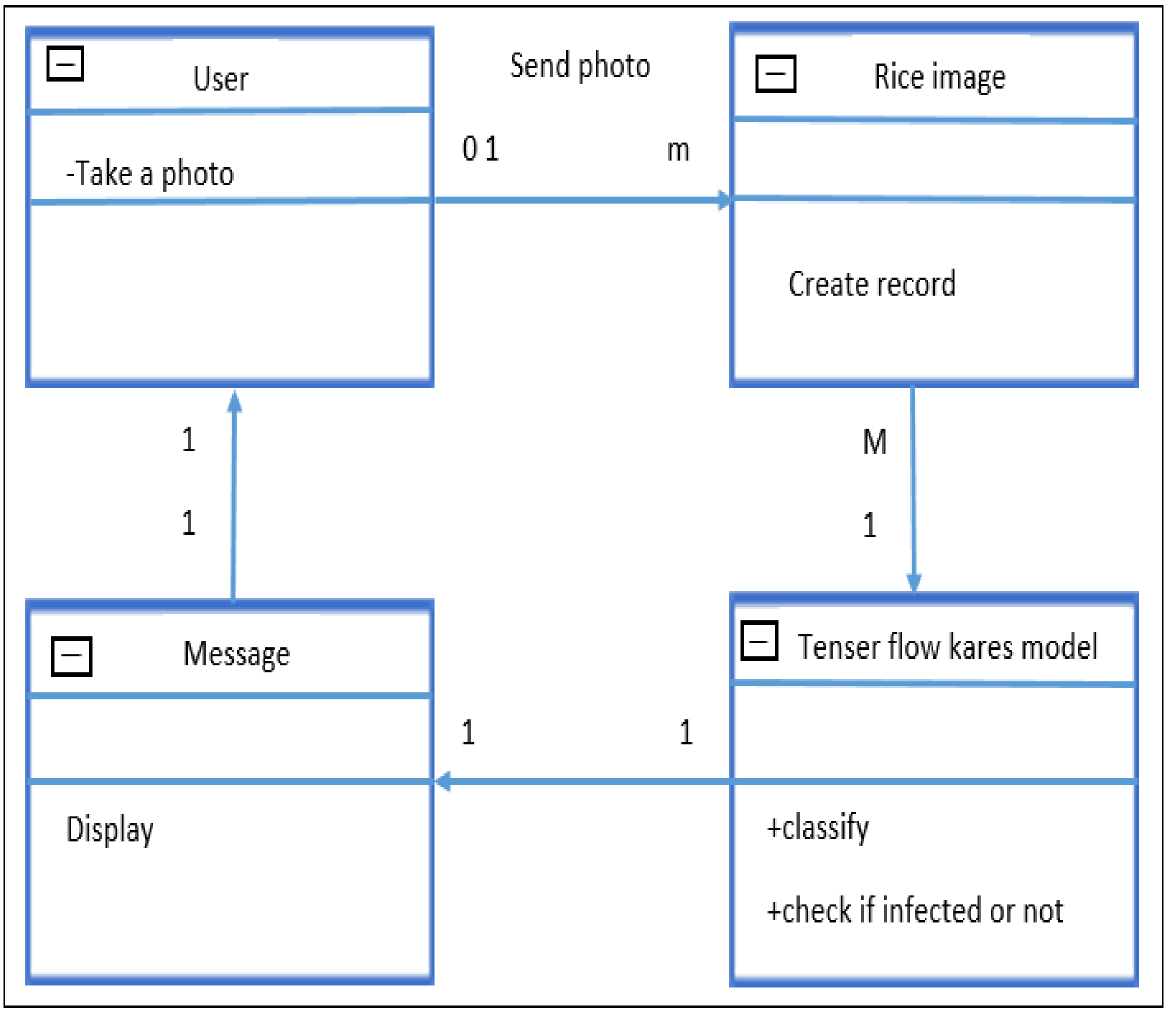


Figure 7 class diagram

Table 3 class diagram

	Class	Description
1	User	choose a photo from gallery
2	Rice Image	Create recod
3	Tensorflow keras model	Classification image if healthy or not
4	Message	Display massage that show the rice type of diseases.

4.4 SEQUENCE DIAGRAM:

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

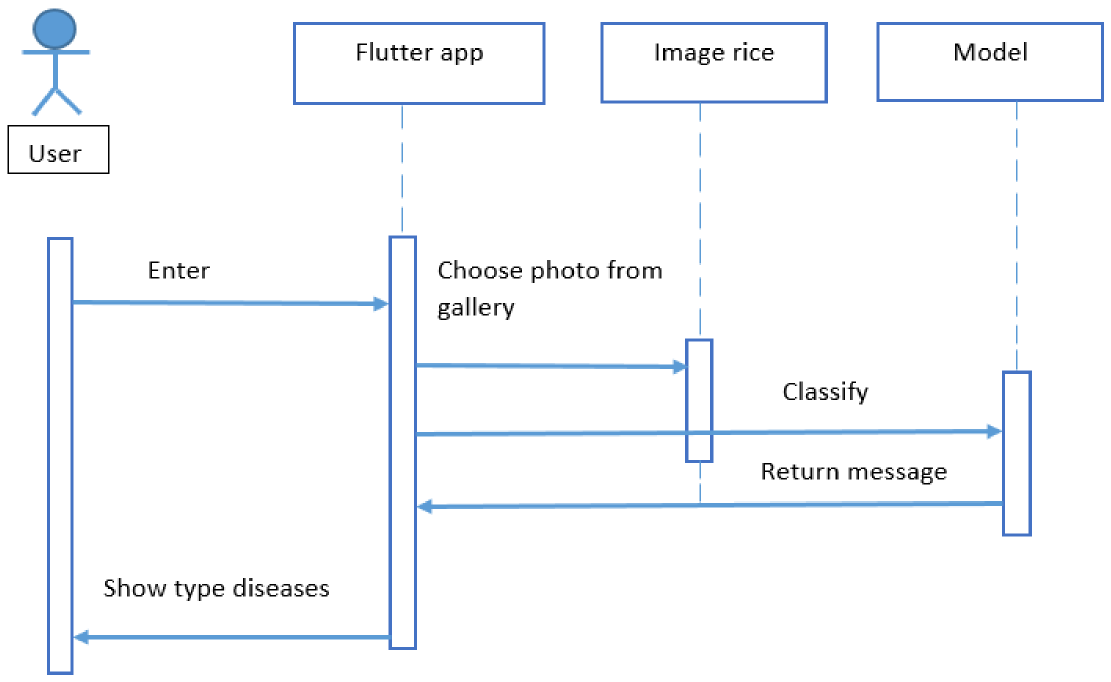


Figure 8 sequence diagram

Table 4 sequence diagram

	Object	Description
1	Flutter app	• User enter to app. • User choose image from gallery • Classification image by using model • Return massage
2	Rice image	Take image from user
3	Model	• Classification image • Return result to app.

4.5 TOOLS THAT USED:

5 Tools

Tools	Description
Programming language (python)	• Create model that use to classification. • Open source • Have many libraries that use to build easy model
Dataset	• Choose dataset that use to train model to use to predict any image we will use to classification
Flutter	To build mobile app
development framework: (TensorFlow Lite API)	Cross-platform that makes it even easier for machine learning models.

C h a p t e r f i v e

5 DEVELOPMENT

System development provides details of the system and takes care of step-by-step details.

5.1 DATA PROCESSING:

In the beginning, there was an obstacle in collecting data, as different sources and different types of data led to the collection of a lot of inaccurate information, but it was avoided in the data collection stage, as it took a great time to understand the work mechanism and work to collect accurate data for later use.

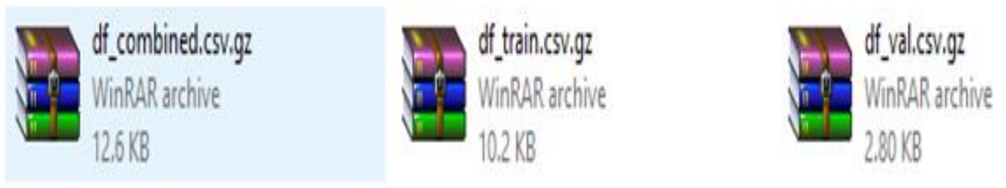
Initially, we searched for a large set of data from the Kaggle website and knew the type of data set to be used.

A 12 GB dataset was used, containing 4 files, 3 files for rice diseases (Leaf Blast diseases, Hispa diseases and Brown spot diseases) and the fourth file for health pictures

The data was divided into two parts, a part for model training and a part for validation.

5.2 DATE (IMAGES) PREPARE PROCESS:

We made some modifications to the data through the code and divided it again and saved it in the form of folders for the possibility if we wanted to modify some files and add pictures to them for use at another time.

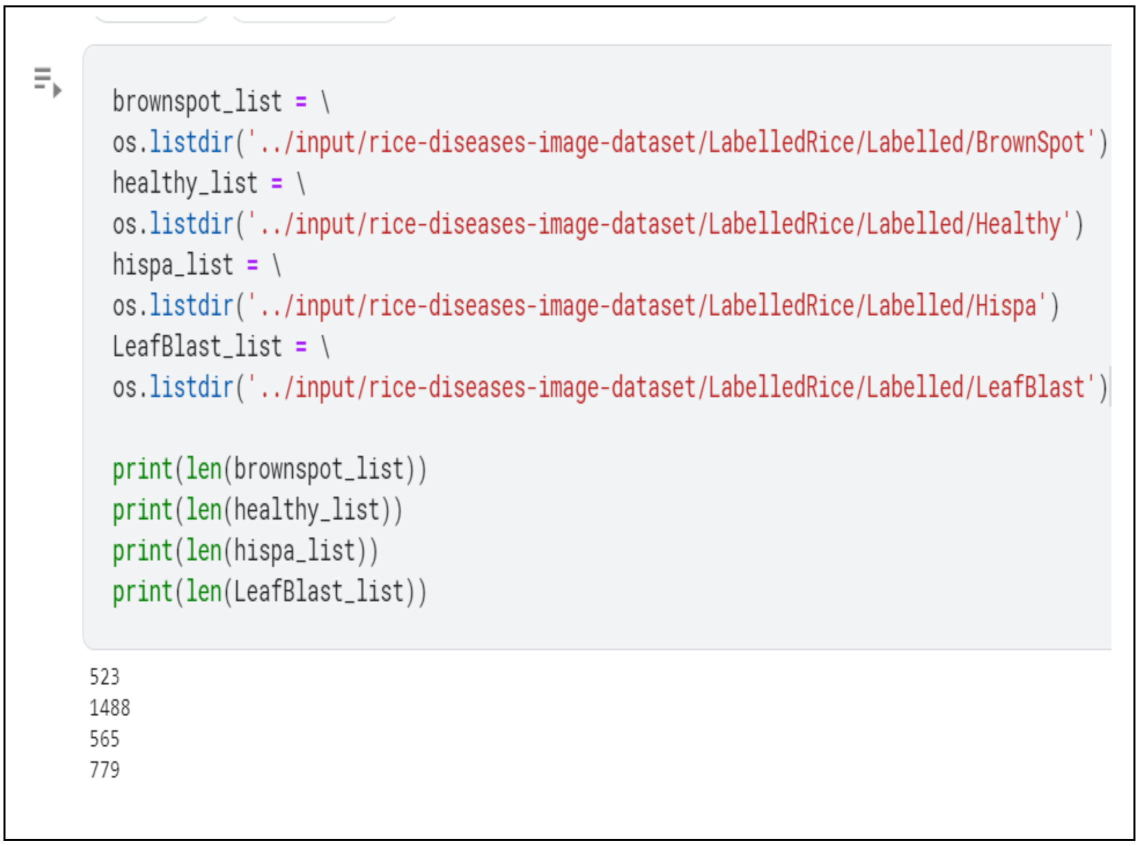

df_combined.csv.gz
WinRAR archive
12.6 KB

df_train.csv.gz
WinRAR archive
10.2 KB

df_val.csv.gz
WinRAR archive
2.80 KB

Figure 9 Folders of Dataset

1- Size of dataset

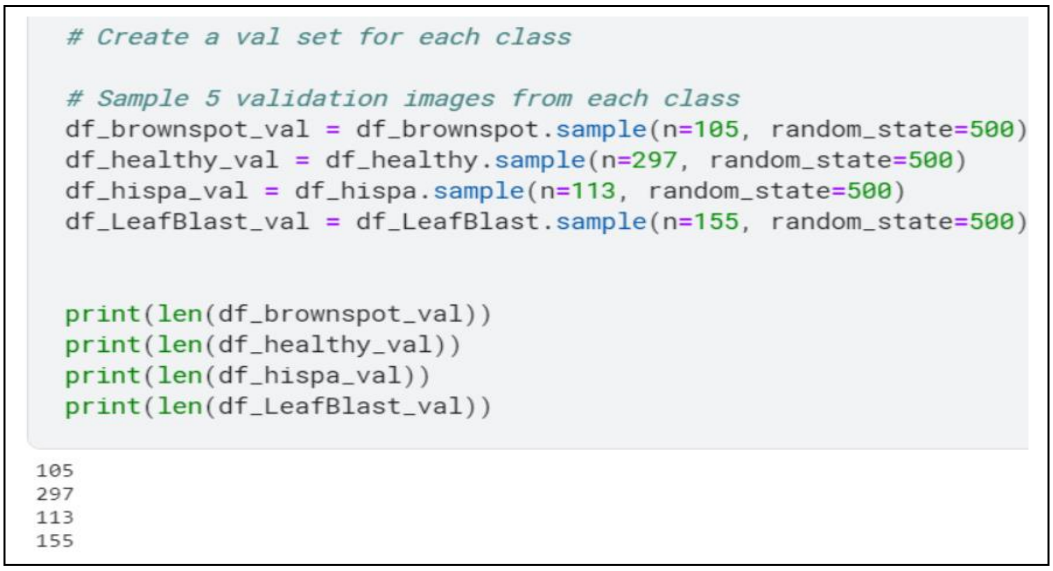


```
brownspot_list = \
os.listdir('../input/rice-diseases-image-dataset/LabelledRice/Labelled/BrownSpot')
healthy_list = \
os.listdir('../input/rice-diseases-image-dataset/LabelledRice/Labelled/Healthy')
hispa_list = \
os.listdir('../input/rice-diseases-image-dataset/LabelledRice/Labelled/Hispa')
LeafBlast_list = \
os.listdir('../input/rice-diseases-image-dataset/LabelledRice/Labelled/LeafBlast')

print(len(brownspot_list))
print(len(healthy_list))
print(len(hispa_list))
print(len(LeafBlast_list))
```

```
523
1488
565
779
```

Figure 10 size of dataset

2- Size of validation data after data splitting


```
# Create a val set for each class

# Sample 5 validation images from each class
df_brownspot_val = df_brownspot.sample(n=105, random_state=500)
df_healthy_val = df_healthy.sample(n=297, random_state=500)
df_hispa_val = df_hispa.sample(n=113, random_state=500)
df_LeafBlast_val = df_LeafBlast.sample(n=155, random_state=500)


print(len(df_brownspot_val))
print(len(df_healthy_val))
print(len(df_hispa_val))
print(len(df_LeafBlast_val))
```

```
105
297
113
155
```

Figure 11 size of validation data after splitting

3- Validation & Training set

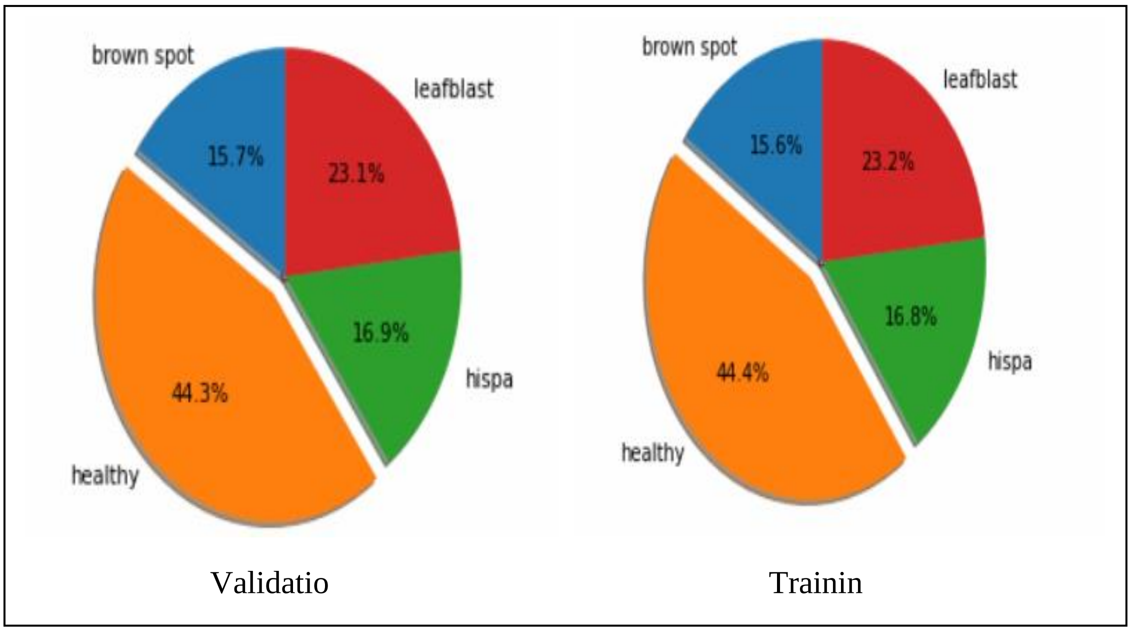
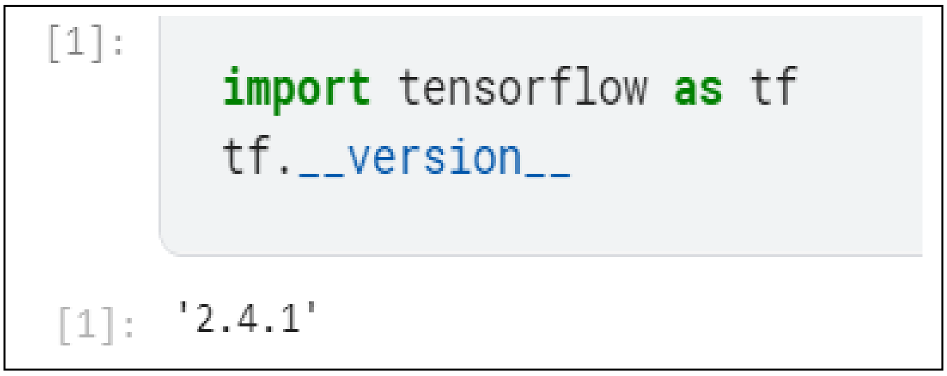


Figure 12 validation & training set

5.3 MODEL:

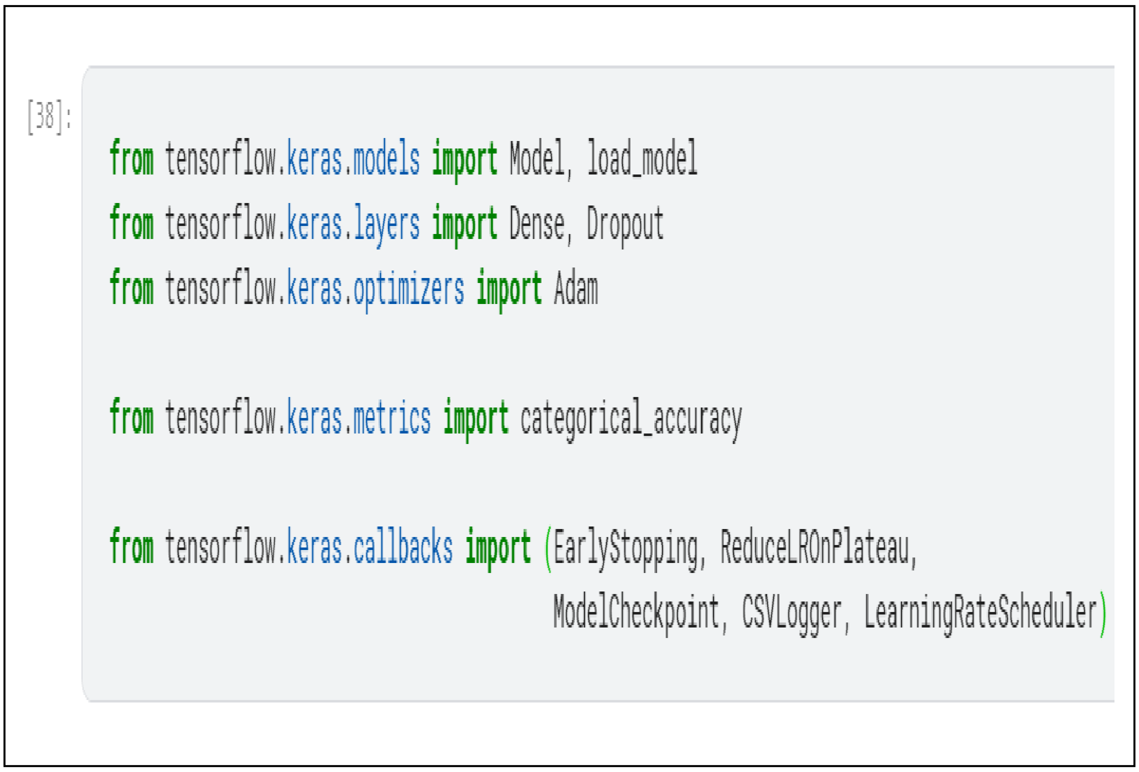
1- Check tensor flow :



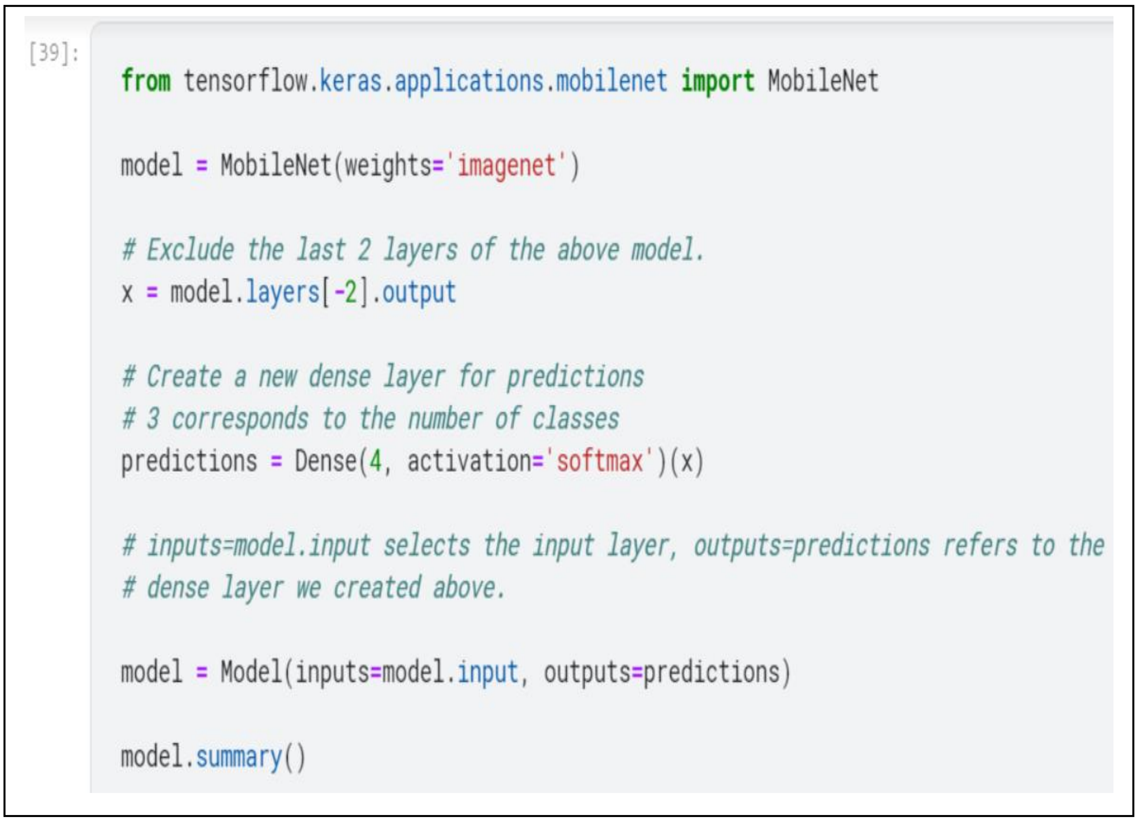
```
[1]:  import tensorflow as tf
      tf.__version__

[1]:  '2.4.1'
```

Figure 13 check tenser flow

2- Model :

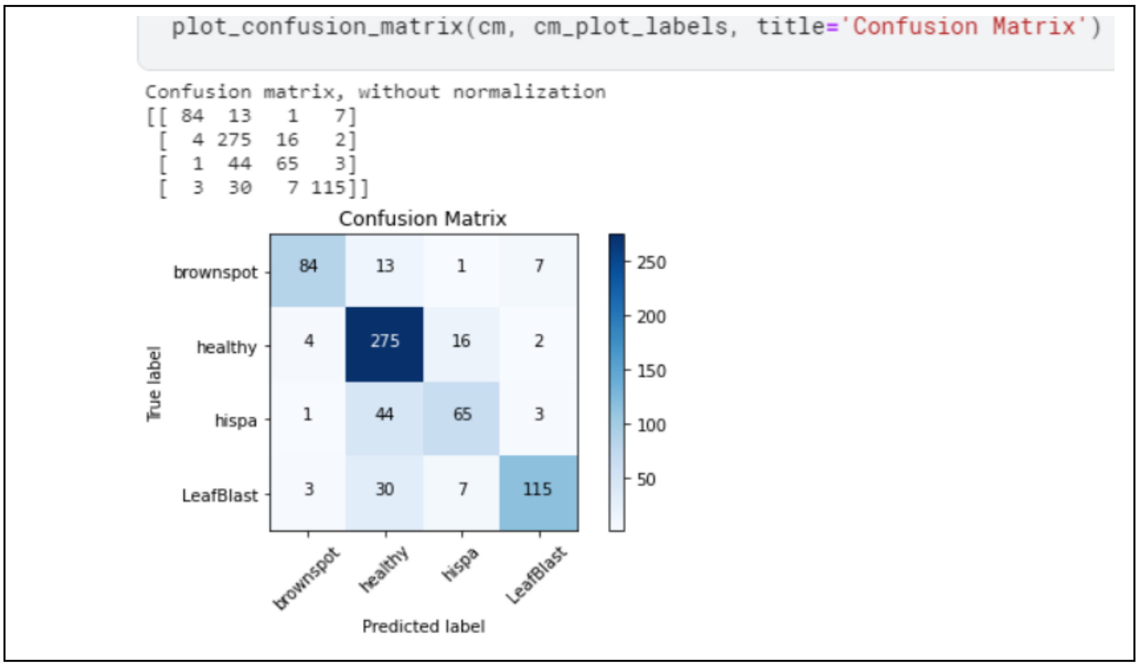


```
[38]:
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.optimizers import Adam

from tensorflow.keras.metrics import categorical_accuracy

from tensorflow.keras.callbacks import (EarlyStopping, ReduceLROnPlateau,
                                        ModelCheckpoint, CSVLogger, LearningRateScheduler)
```



```
[39]:
from tensorflow.keras.applications.mobilenet import MobileNet

model = MobileNet(weights='imagenet')

# Exclude the last 2 layers of the above model.
x = model.layers[-2].output

# Create a new dense layer for predictions
# 3 corresponds to the number of classes
predictions = Dense(4, activation='softmax')(x)

# inputs=model.input selects the input layer, outputs=predictions refers to the
# dense layer we created above.

model = Model(inputs=model.input, outputs=predictions)

model.summary()
```

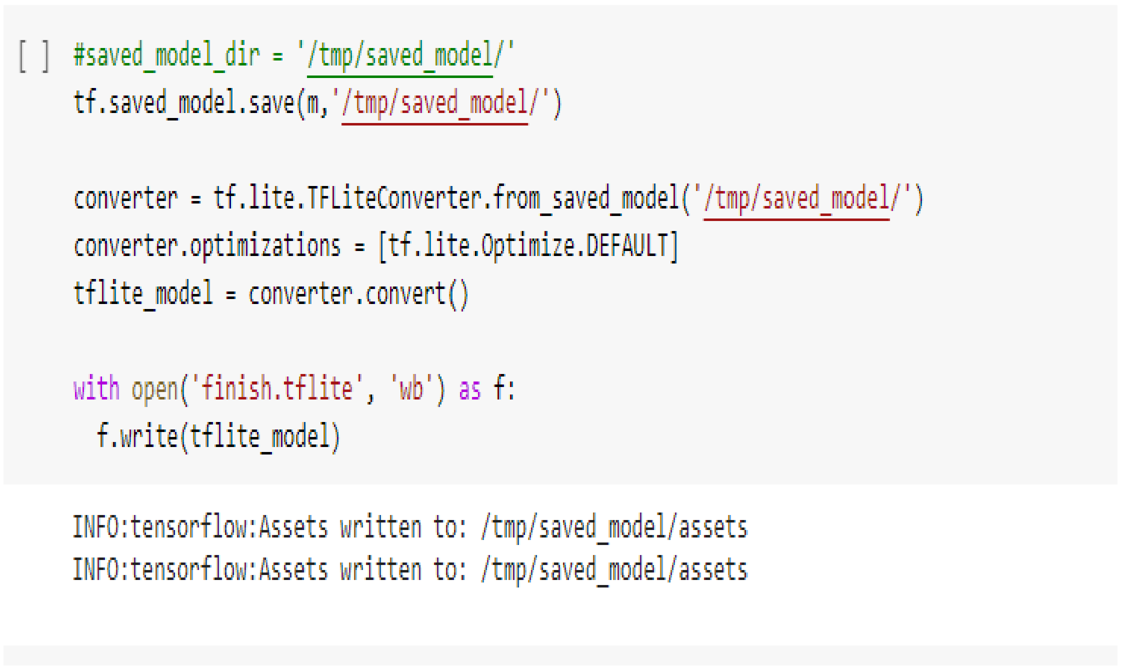
Figure 14 model



```
plot_confusion_matrix(cm, cm_plot_labels, title='Confusion Matrix')

Confusion matrix, without normalization
[[ 84  13   1   7]
 [  4 275  16   2]
 [  1  44  65   3]
 [  3  30   7 115]]
```

	brownspot	healthy	hispa	LeafBlast
brownspot	84	13	1	7
healthy	4	275	16	2
hispa	1	44	65	3
LeafBlast	3	30	7	115

Figure 15 Confusion matrix



```
[ ] #saved_model_dir = '/tmp/saved_model/'
    tf.saved_model.save(m,'/tmp/saved_model/')

    converter = tf.lite.TFLiteConverter.from_saved_model('/tmp/saved_model/')
    converter.optimizations = [tf.lite.Optimize.DEFAULT]
    tflite_model = converter.convert()

    with open('finish.tflite', 'wb') as f:
      f.write(tflite_model)

    INFO:tensorflow:Assets written to: /tmp/saved_model/assets
    INFO:tensorflow:Assets written to: /tmp/saved_model/assets
```

Figure 16 Model.Tflite

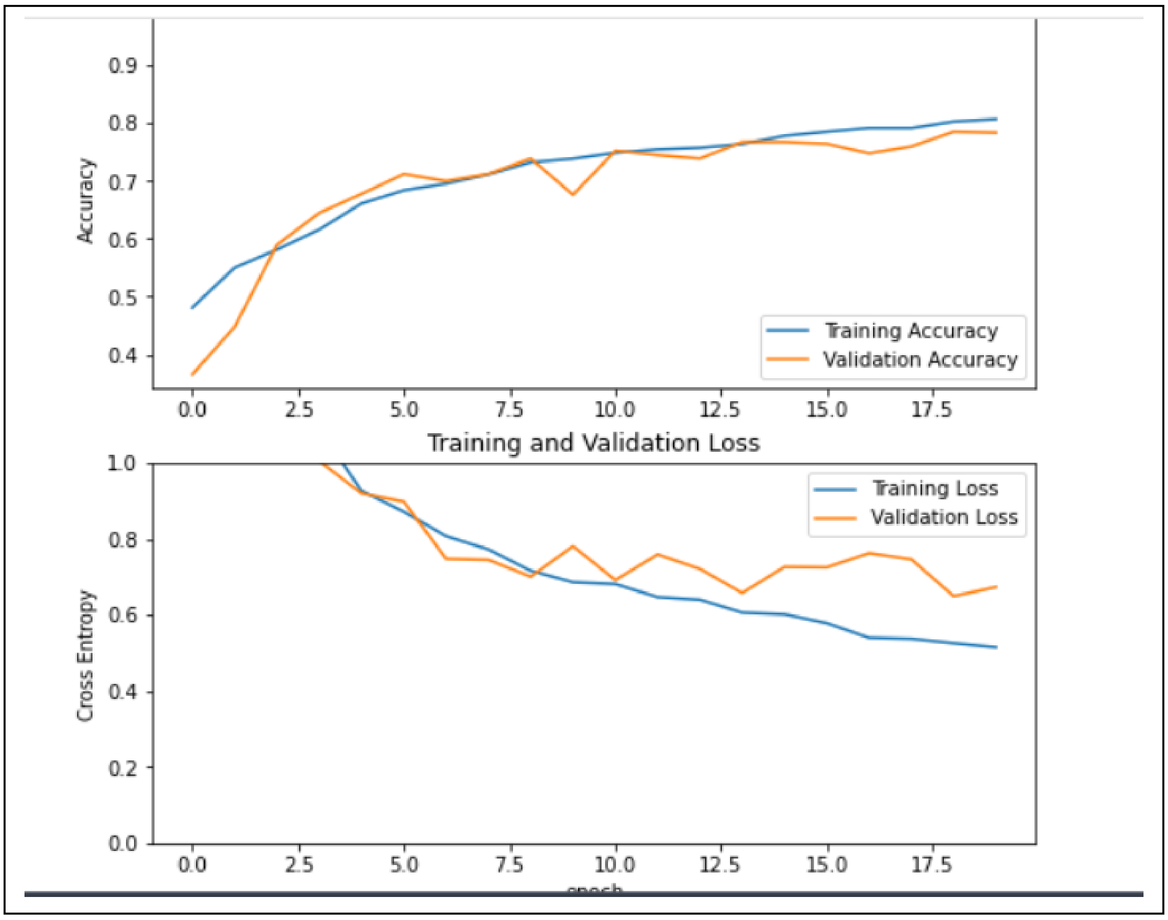
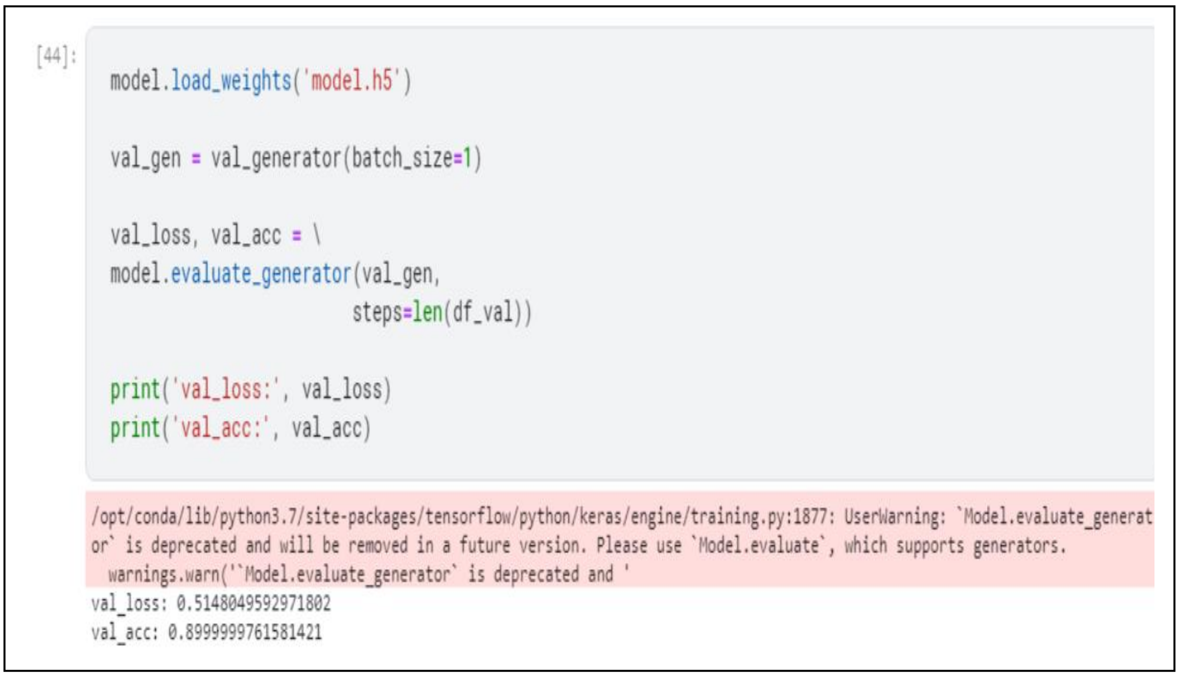


Figure 17 Training & Validation accuracy



```
[44]:
model.load_weights('model.h5')

val_gen = val_generator(batch_size=1)

val_loss, val_acc = \
model.evaluate_generator(val_gen,
                         steps=len(df_val))

print('val_loss:', val_loss)
print('val_acc:', val_acc)
```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/training.py:1877: UserWarning: `Model.evaluate_generat
or` is deprecated and will be removed in a future version. Please use `Model.evaluate`, which supports generators.
  warnings.warn('`Model.evaluate_generator` is deprecated and '
val_loss: 0.5148049592971802
val_acc: 0.8999999761581421
```

Figure 18 accuracy

Chapter six

6 BACKEND IMPLEMENTATION

Now that we have our implementation of the main analysis functions and we've already tested them on some of our data. We need to put a wrapper around them as a way for a user to drive these functions and view their results easily.

Since we chose to build the system as a mobile application, and the user interacts only with the front end or API, that means that all plotting and presentation functions reside on the frontend. As well as driver functions that take input from the user and invoke backend functions.

However, in this chapter we'll only be looking at backend functions on our mobile application, which are basically all the kinds of requests the user can make on our mobile application.

6.1 TENSORFLOW LITE API:

• TensorFlow Lite converter:

The TensorFlow Lite converter takes a TensorFlow model (model .h5 file) and generates a TensorFlow Lite FlatBuffer file (.tflite).

The converter supports SavedModel directories, tf.keras models, and concrete functions.

• Device deployment:

The Tensorflow Lite FlatBuffer file is then integrated and deployed to a client device (e.g. mobile, embedded) and run locally using the Tensorflow Lite interpreter.

This conversion process is shown in the diagram below:

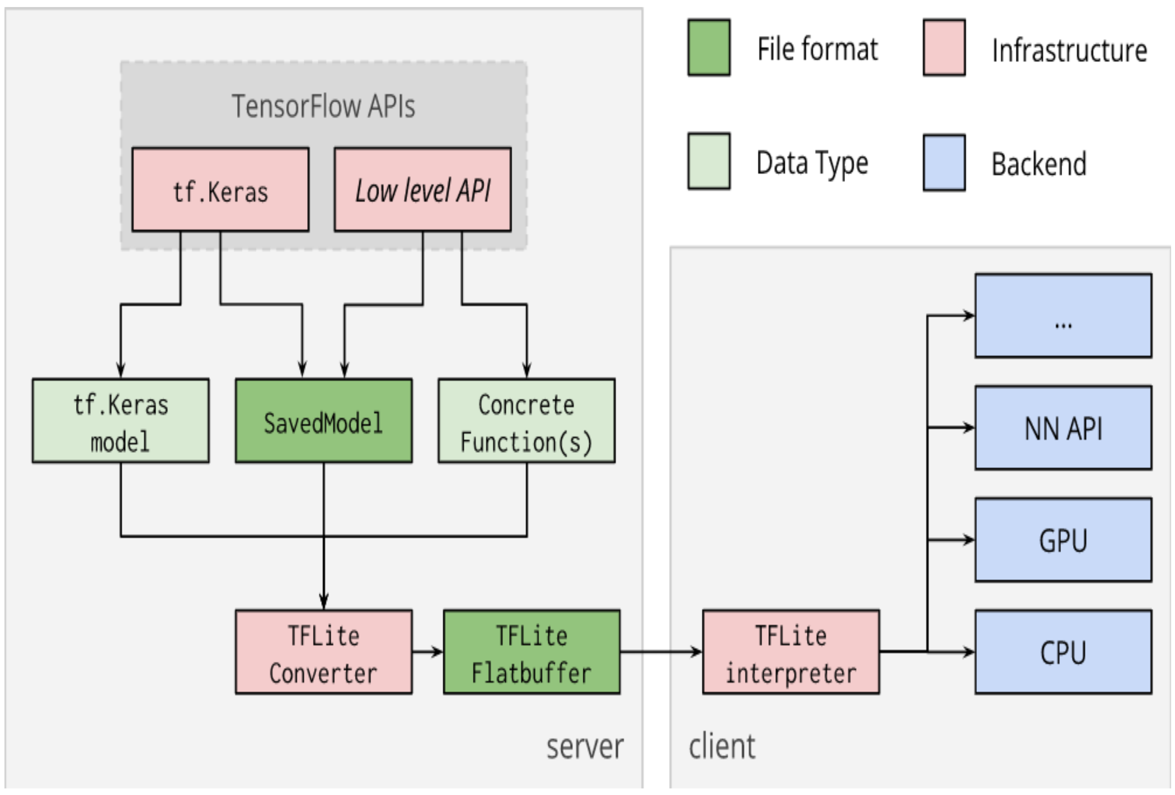


Figure 19 tenser flow API

• Converting models

The TensorFlow Lite converter should be used from the Python API.
Using the Python API makes it easier to convert models as part of a model development pipeline and helps mitigate compatibility issues early on. Alternatively, the command line tool supports basic models.

• TensorFlow Lite API Reference

The API reference documentation provides detailed information for each of the classes and methods in the TensorFlow Lite library. Choose your preferred platform from the list below.

* Python API reference
* Android (Java) API reference
* C++ API reference

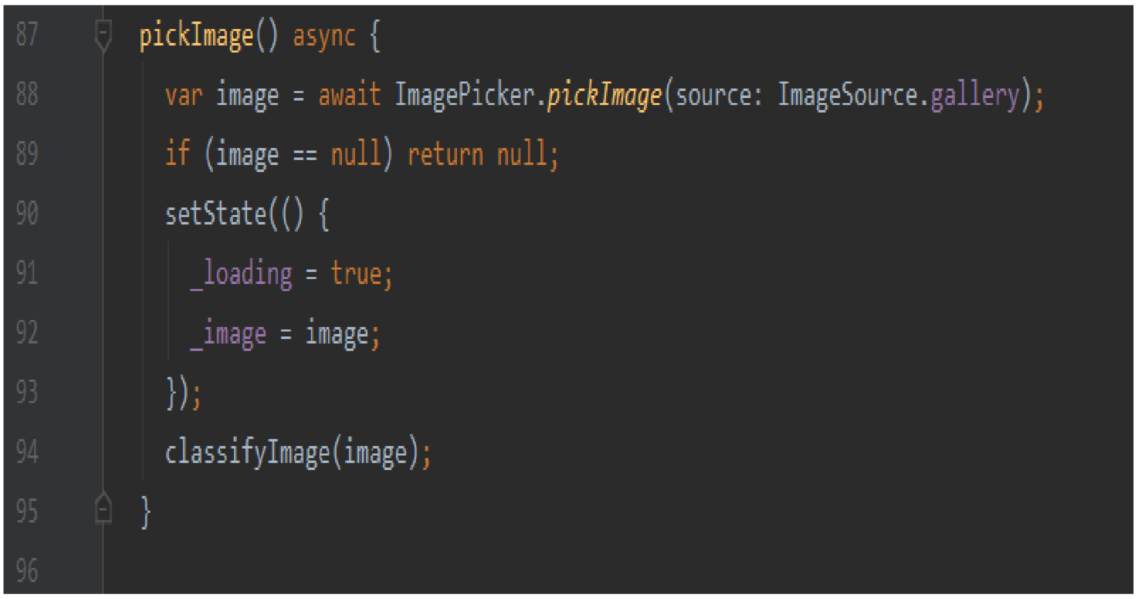
• Python API :- The Python API for converting TensorFlow models to TensorFlow Lite is tf.lite.TFLiteConverter.

TFLiteConverter provides the following classmethods to convert a model based on the original model format:

* TFLiteConverter.from_saved_model(): Converts SavedModel directories.
* TFLiteConverter.from_keras_model(): Converts tf.keras models.
* TFLiteConverter.from_concrete_functions(): Converts concrete functions.

6.2 PACKAGES USED AT BACKEND DEVELOPER:

• **Image Picker:**
A Flutter plugin for iOS and Android for picking images from the image library, and taking new pictures with the camera.

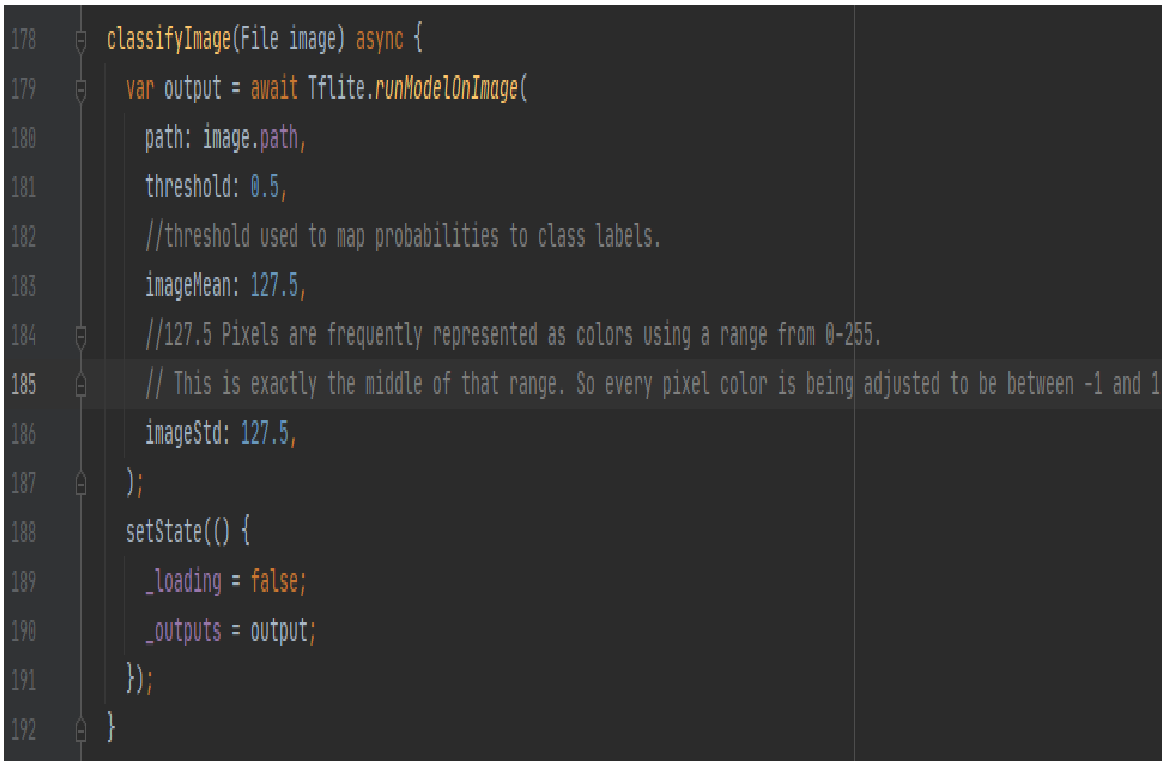


```
pickImage() async {
  var image = await ImagePicker.pickImage(source: ImageSource.gallery);
  if (image == null) return null;
  setState(() {
    _loading = true;
    _image = image;
  });
  classifyImage(image);
}
```

Figure 20 Image Picker

At this function the ImagePicker enables to access gallery and get the image which selected.

- If user don't selected an image the image value is equal to null
- Else Save the selected image at image and update the state by updating the private variables which type is Boolean (_loading) to True and (_img) to image which type is file.

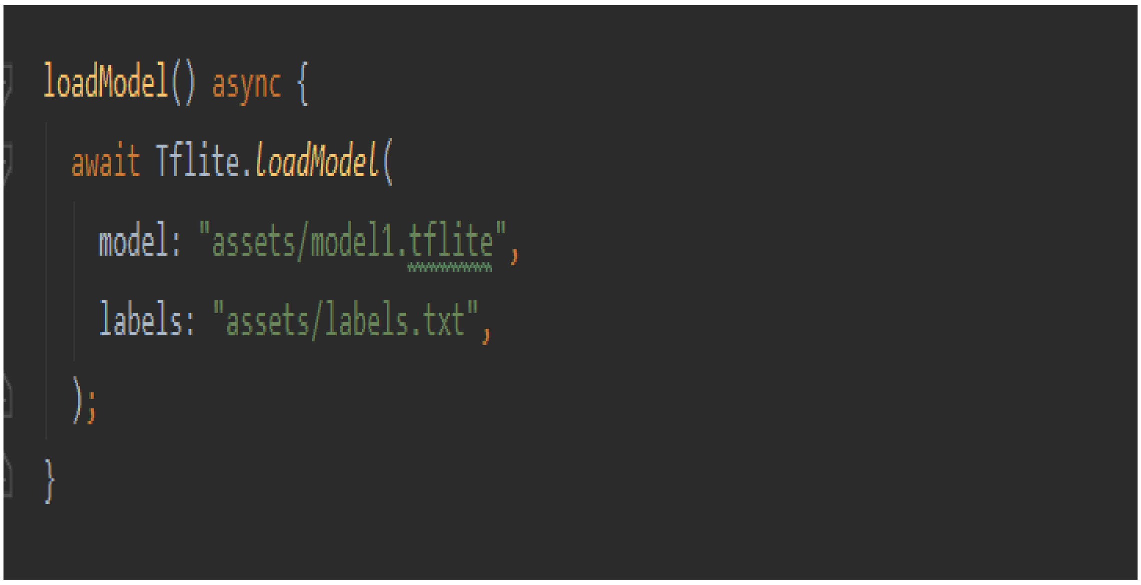


```
classifyImage(File image) async {
  var output = await Tflite.runModelOnImage(
    path: image.path,
    threshold: 0.5,
    //threshold used to map probabilities to class labels.
    imageMean: 127.5,
    //127.5 Pixels are frequently represented as colors using a range from 0-255.
    // This is exactly the middle of that range. So every pixel color is being adjusted to be between -1 and 1
    imageStd: 127.5,
  );
  setState(() {
    _loading = false;
    _outputs = output;
  });
}
```

- ❖ TFlite run model on the selected image and return the results and save it at the variable output which type is var
- ❖ Update the state by update the value of _loading and _output

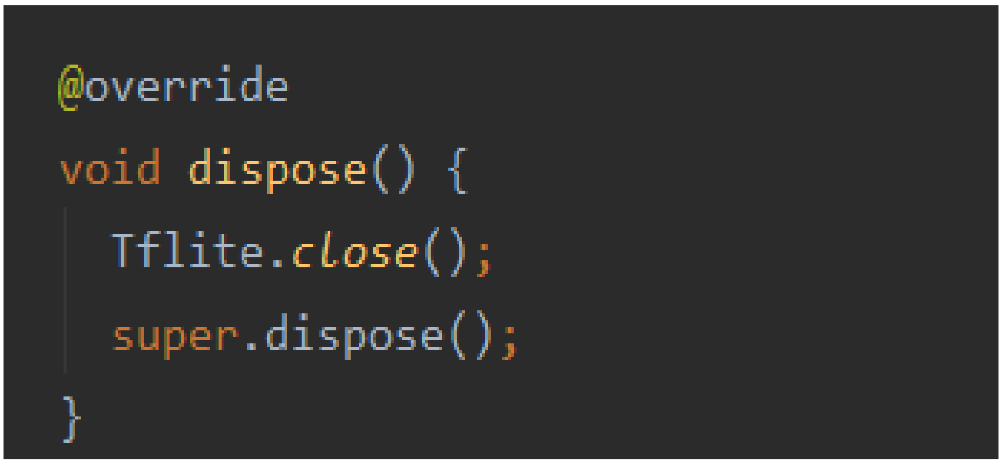
- **Tflite:**

A Flutter plugin for accessing TensorFlow Lite API. Supports image classification, object detection (SSD and YOLO), Pix2Pix and Deeplab and PoseNet on both iOS and Android.



```
loadModel() async {
  await Tflite.loadModel(
    model: "assets/model1.tflite",
    labels: "assets/labels.txt",
  );
}
```

Load the model by using function TFlite.loadModel();
passing model path and labels path.



```
@override
void dispose() {
  Tflite.close();
  super.dispose();
}
```

After closing the application TFlite close the model.

C h a p t e r s e v e n

7 FRONTEND IMPLEMENTATION

Almost everything related to the system was discussed in the previous chapters. However, the most interesting part, and the one that the user often sees, has yet to be detailed.

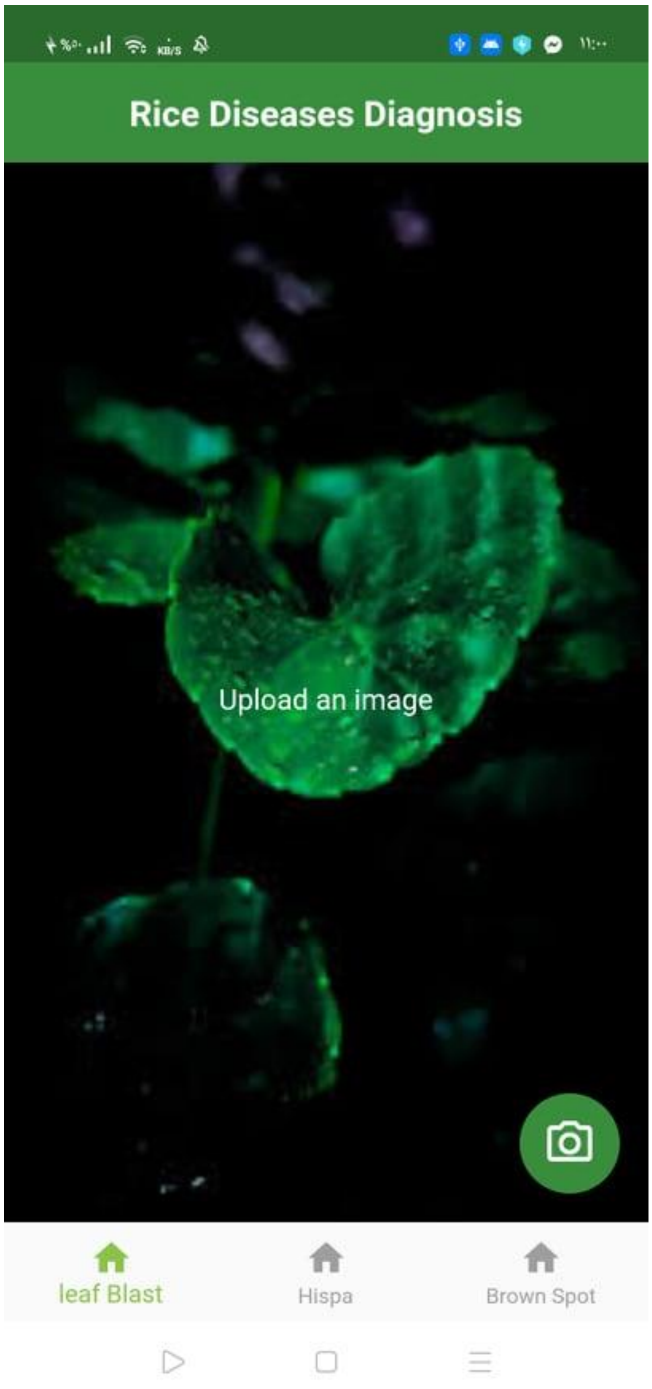
In this chapter, we will discuss the front end.

When the user visits the front-end for the first time, he will simply see an interface, a button that allows you to load the image, some windows that give him some information about the disease resulting from the image, and suggest some practical ways to treat this disease, like most mobile applications. The user enters an image, then request is executed.

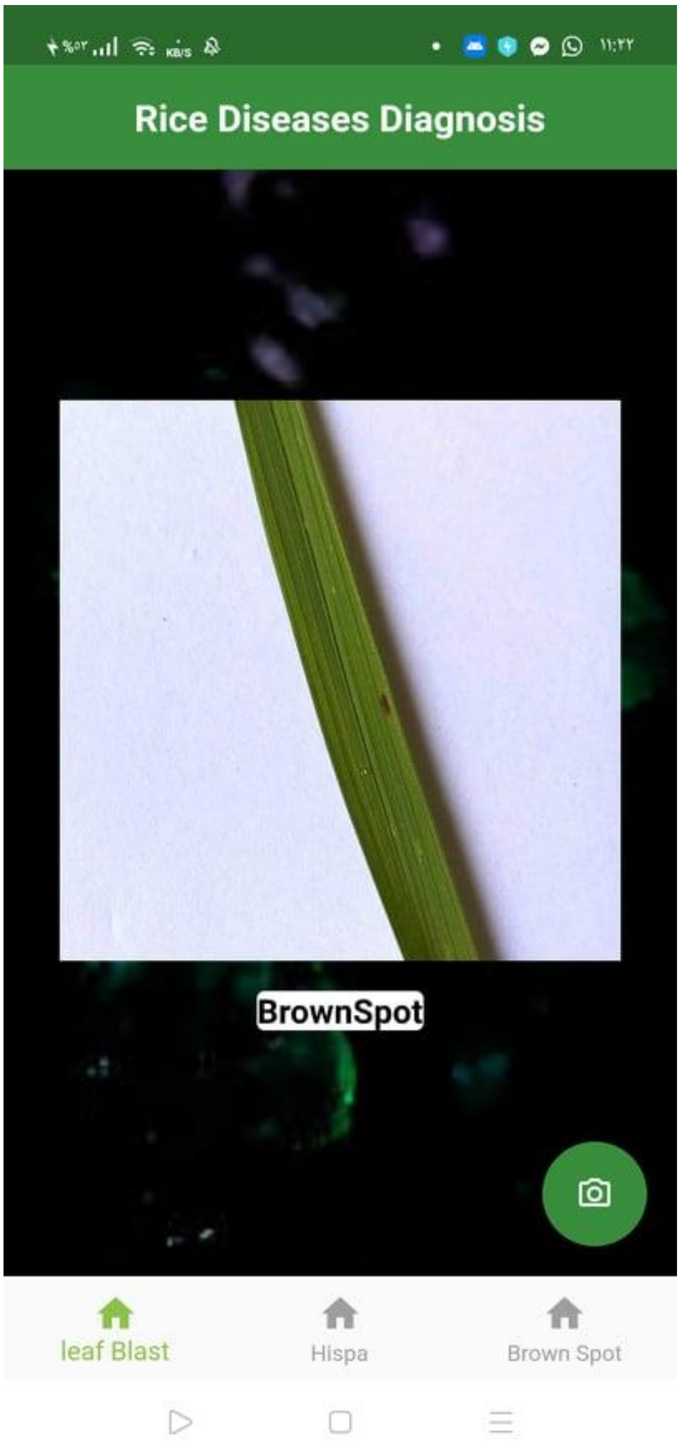
Rice Doctor is an introductory, diagnostic and therapeutic application for farmers, gardeners, and everyone who works in agriculture working in the field of rice cultivation. Whether you're a farmer interested in sustainable practice or a fan of rice in urban gardens, the app tests your crops for diseases, pests, and nutrient deficiencies with the help of a simple smartphone picture.

In this application (Rice Doctor App) an image processing based approach is proposed and used to detect rice plant diseases.

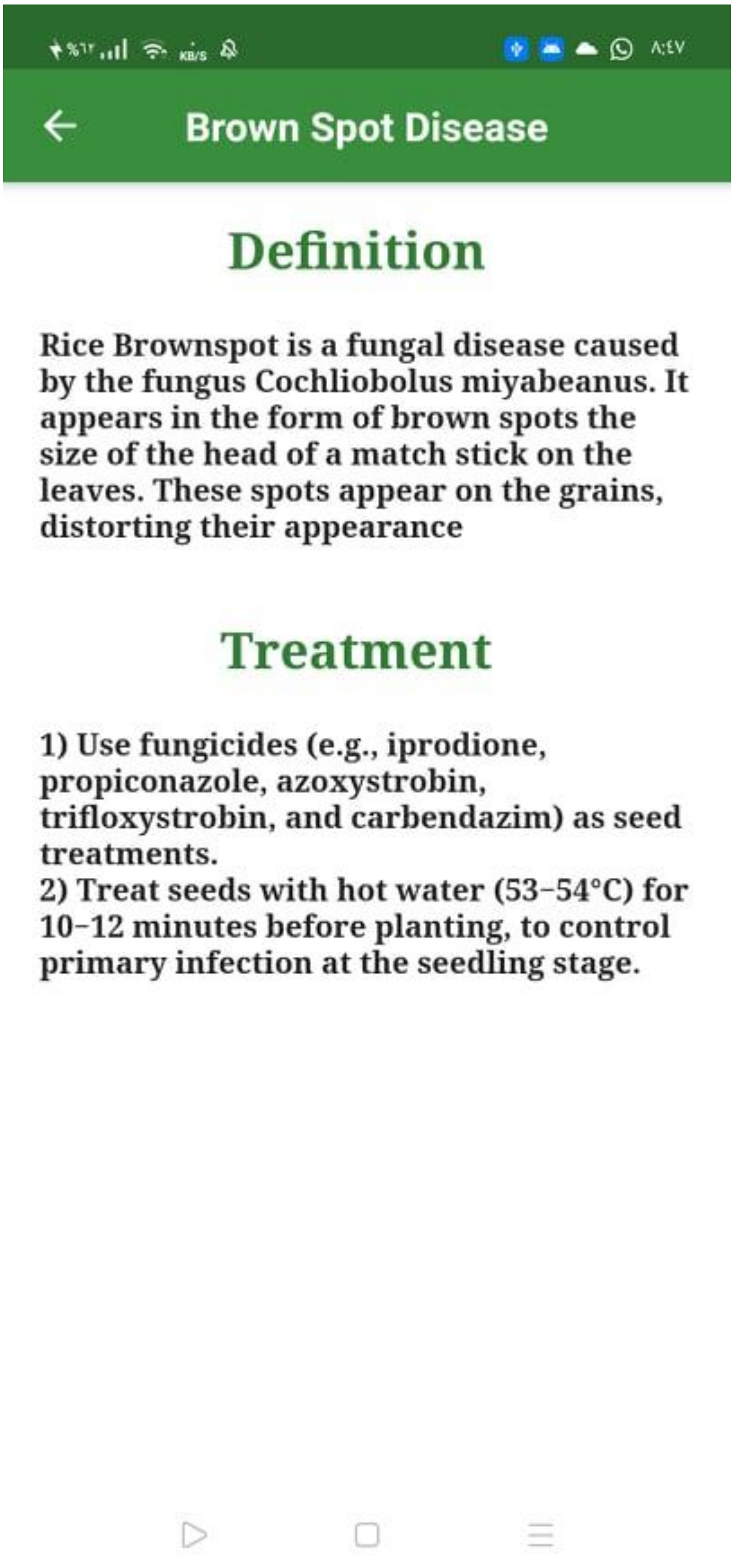
- ✓ This is the first interface of the application that appears directly to the user when opening the application. It consists of the application bar that bears the name of the application, a background and a button that leads you to your phone gallery, and at the end of the interface there is a navigation bar so that the user can navigate between disease identification pages and find treatment methods.



- ✓ Choose the image from the gallery of previously captured images.
- ✓ When uploading the image to the application, the application determines whether it is infected or not, and if it is infected, it will determine the name of this disease.



- ✓ User can enter any page to know more about disease and its treatment.



Definition

Rice Brownspot is a fungal disease caused by the fungus Cochliobolus miyabeanus. It appears in the form of brown spots the size of the head of a match stick on the leaves. These spots appear on the grains, distorting their appearance

Treatment

1) Use fungicides (e.g., iprodione, propiconazole, azoxystrobin, trifloxystrobin, and carbendazim) as seed treatments.
2) Treat seeds with hot water (53–54°C) for 10–12 minutes before planting, to control primary infection at the seedling stage.

- ✓ This application is designed using Flutter.

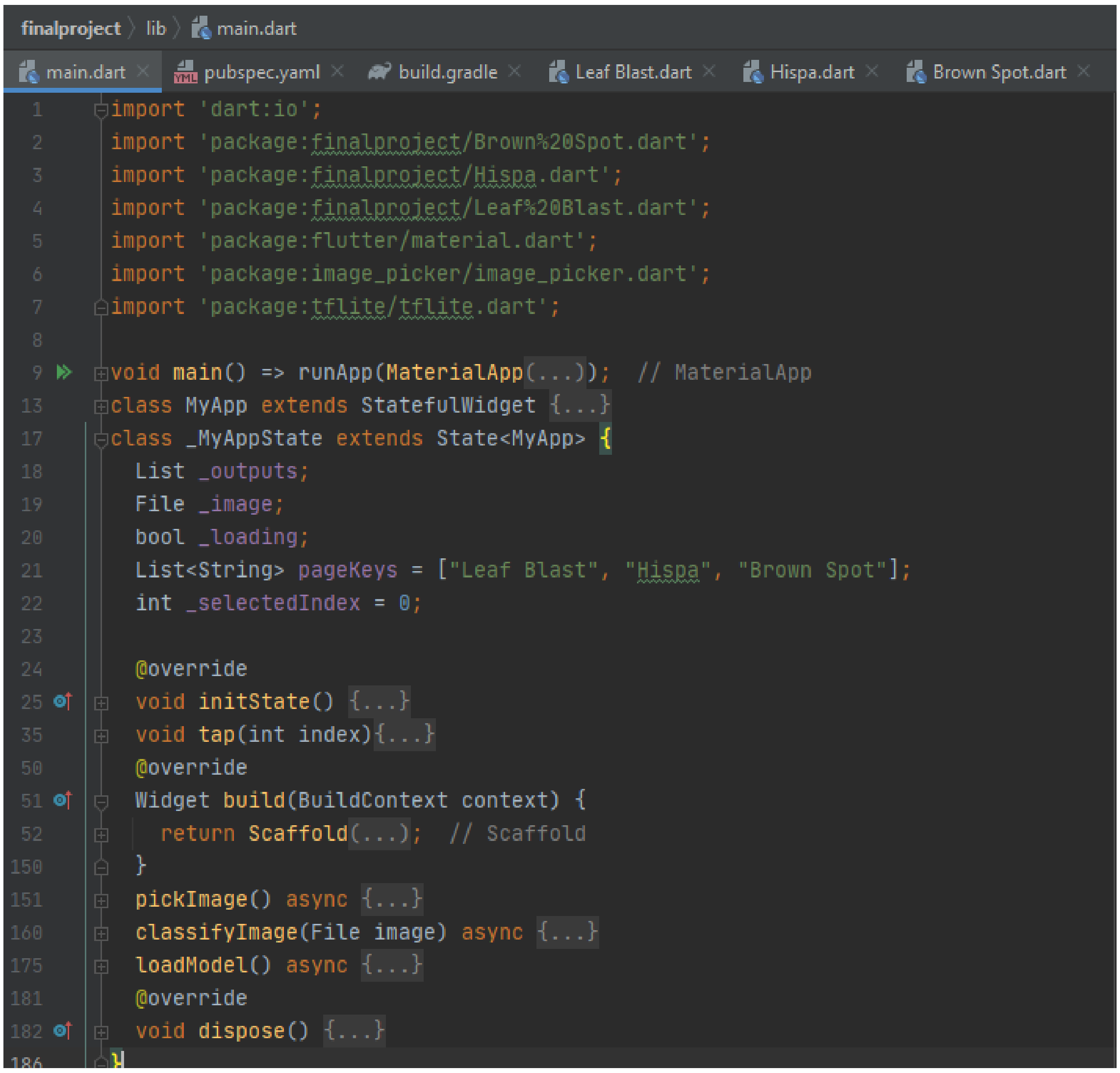
  Flutter is Google’s UI toolkit for building simple, natively compiled applications for mobile, web, and desktop from a single codebase.

- ✓ Software used to Design User Interface :-
  - Adobe Photoshop CC used to Design UI.

- ✓ Software used to implement User Interface :-
  - Android Studio

- ✓ Language used to implement User Interface :-
  - Dart

The implementation code:



```
import 'dart:io';
import 'package:finalproject/Brown%20Spot.dart';
import 'package:finalproject/Hispa.dart';
import 'package:finalproject/Leaf%20Blast.dart';
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:tflite/tflite.dart';

void main() => runApp(MaterialApp(...));  // MaterialApp
class MyApp extends StatefulWidget {...}
class _MyAppState extends State<MyApp> {
  List _outputs;
  File _image;
  bool _loading;
  List<String> pageKeys = ["Leaf Blast", "Hispa", "Brown Spot"];
  int _selectedIndex = 0;

  @override
  void initState() {...}
  void tap(int index){...}
  @override
  Widget build(BuildContext context) {
    return Scaffold(...);  // Scaffold
  }
  pickImage() async {...}
  classifyImage(File image) async {...}
  loadModel() async {...}
  @override
  void dispose() {...}
}
```

8 CONCLUSION

Despite the importance of the rice crop to people, the impact of diseases on it reduces its production, but with the use of modern technologies using artificial intelligence and machine learning to understand the mechanism of the plant and obtain high-accuracy results, it has led to the fact that it is possible to discover diseases that affect the plant from the beginning and thus treat them faster Thus, high results are achieved.

In this paper, we used a dataset for a large group of rice plants and photographed them at different angles in different weather conditions, using pictures of rice diseases, and training the model on them to obtain high accuracy results.

Several augmentation technique was used. Affiliate ready libraries to transfer learning technology (Mobile Net) to prepare a file Pictures and increase their number through some operations for that technology, a new architecture, the mobile network, has been proposed to detect plant diseases. The accuracy is 89.99999.

Due to its architectural design it proved to be successful in situations with complex surroundings. Accuracy could potentially be enhanced by the exploitation of other information sources such as location. This model was connected to a mobile app using Tensor Flow Lite for high quality and accuracy when used with our model (Mobile Net).

Chapter Nine

9 FUTURE WORK

We have added the common diseases that severely affect the productivity of the land plant and affect the national income, so we will seek in the future to add more other types of diseases that are not common to the rice plant, even if they affect slightly for the sake of the integrated application.

We also seek to provide data for most other diseases and treat them through the application to know the most diseases that affect the cultivation of the farmer and help him avoid those diseases and provide treatment methods for him in a simple way through our application.

It is likely to increase the spread of the application we will create a version of it available on the web (web app) and a version on the desktop (desktop app).

We will continuously develop the application design and internal updates to extend its life and continuity.

Also, the continuous development of treatment methods so that the farmer can keep pace with what is modern in the field of treating diseases

Finally, we will strive to increase the efficiency of the model as much as possible

REFERENCES

1. https://www.kaggle.com/minhhuy2810/rice-diseases-image-dataset

2. https://www.tensorflow.org

3. https://www.python.org

4. https://flutter.dev/

5. Prajapati, Harshadkumar B., Jitesh P. Shah, and Vipul K. Dabhi. "Detection and Classification of Rice Plant Diseases." Intelligent Decision Technologies. IOS Press, January 1, 2017. https://content.iospress.com/articles/intelligent-decision-technologies/idt301.

6. Kaur, Sukhvir, Shreelekha Pandey, and Shivani Goel. "Plants Disease Identification and Classification Through Leaf Images: A Survey." Archives of Computational Methods in Engineering. Springer Netherlands, January 19, 2018. https://link.springer.com/article/10.1007%2Fs11831-018-9255-6.