AUTOMATED PLANT IDENTIFICATION

A PROJECT REPORT

Submitted by

GOWRI G (2019202013)

in partial fulfillment for the award of the degree

Of

INFORMATION AND COMMUNICATION ENGINEERING

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI-600 025

MAY 2022

BONAFIDE CERTIFICATE

Certified that this project report **AUTOMATED PLANT IDENTIFICATION** is the bonafide work of Gowri G (2019202013) who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE: Chennai

DATE: 02-05-2022

Dr L SAI RAMESH
TEACHING FELLOW
PROJECTGUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI - 600025

COUNTERSIGNED

Dr. S. SRIDHAR HEAD OF THE DEPARTMENT

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI-600 025

TABLE OF CONTENTS

ABSTRACT

Since there are many classifications of herbal plants, it becomes time consuming and less accurate when predicted by the non-professionals. Therefore we need proper classification algorithm to identify the herbals when the input is in abundance and the output should be more precise, In that case, we use Machine Learning for Identification and Classification of Herbal plants, In this project Artificial Neural Network Algorithms are used for classification of Herbal plants. Three main steps that are image preprocessing, feature extraction and recognition were carried out to develop the proposed system. This system can be implemented by using the jupyter notebook software tool. Machine learning concepts drastically decrease the time needed to arrange an exact map. In this project we will be using Artificial Neural Network (ANN) Algorithm for classification.

CHAPTER 1

INTRODUCTION

1.1 General

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

Now, learning more about plants is just a photo click away. The Plant Identification app is a journal guide for all types of plants, flowers, trees. A user just has to click a photo and an instant report about that particular plant image is shown through the mobile app.

In these plant identifiers application, a user can click a photo and get desired information. He can also organically search for the plant species name by “search bar”.
You can create an app for plants with a “photo drag” feature. A user who is in a hurry to reach somewhere pauses for a bit to take a photo of an unknown plant. He can put the photo afterward in the plant identification app. Instead of clicking a photo opening the app at that very moment, your app development will create this exception.

1.2 Problem Statement

There are various Plant identification software available in this world developed and researched by various other developers. The pros and cons of the various Plant identification software are given in details here. Many images are not clear which creates problem to identify plant. Plants may not match to any species in the database or matches to more than 2 or more plants. Same types of leaves are also problem.

1.3 Objective

The Objective of project is to introduce plant expert determination, identifying plant Recognition and comparison using Artificial Neural Network (ANN) and to frame RGB to GRAY scale images.

CHAPTER 2

LITERATURE REVIEW

2.1 Image acquisition

Paper [1]: Sample images in this study consist of many plant structures, e.g .branches, stems, leaves, syconium's and others. Where possible, only intact leaves were selected that had no apparent tearing and also free of damage from pest or disease. Young leaves that are evidently small sized were ignored. Images used in this study were taken at 4leaf stage of plant growth which was the beginning of critical weed control period.

Paper [2]: Since images were acquired from normal growth condition in the field, several plants appeared in each image. Therefore, in the next step, from these collections of images, some parts were cropped to comprise just a single plant.

Paper [3]: Training of the neural network was carried out using 129 examples of each species, each derived from a different wild-collected herbarium specimen. This resulted in 516 training records, each containing data from a single leaf found on a botanical herbarium specimen, these data were converted to a standard ASCII tabulated numeric format suitable for input to the neural network

2.2 Image pre-processing

Paper [1]: The leaf images contain only one object, the leaf. Since all leaves are not perfectly flat, image capturing would always cast a shadow underneath the leaf. As HSV value conversion alters the original colour, this step serves as a guidance for the subsequent edge detection of RGB value leaf images, rather than producing a final image for feature extraction. boundaries were extracted by removing any deformity within the leaf outline and displayed the complete leaf in white patch. shows some leaf images and their respective processed images.

Paper [2]: The captured RGB images were transferred to the computer to be processed using image processing toolbox of jupyter notebook software, inside the images were detected and primary binary images resulted. These binary images were used in the main weed detection algorithm.

Paper [3]: A representation of the network architecture is shown in, though the number of nodes in each layer is different from that shown. The input vectors were normalized in the range ±0.9 to reduce the training time required for the inputs to the hidden nodes to reach the domain of the sigmoid activation function.

2.3 Feature extraction

Paper [1]: Processed images from previous steps were transformed into a set of parameters that describe the leaf features. There are four classes of features extracted in this study: morphological features (shape), Hu moment invariants feature, texture features and histogram of oriented gradients.

Paper [2]: Shape features are a set of features commonly used in many studies. Shape values included in this study are: area, perimeter, eccentricity, and minor and major axes as shown in. Shape factors and moment invariants as region-based feature and Fourier descriptors as boundary-based features were extracted from images and were evaluated.

Paper [3]: The samples were binarized and three sets of shape-based features including shape factors, moment invariant features and boundary Fourier descriptors were extracted.

2.4 Proposed Methodology

A typical image based plant identification system is shown in fig. 1 and the major steps are explained in consecutive sub-sections.

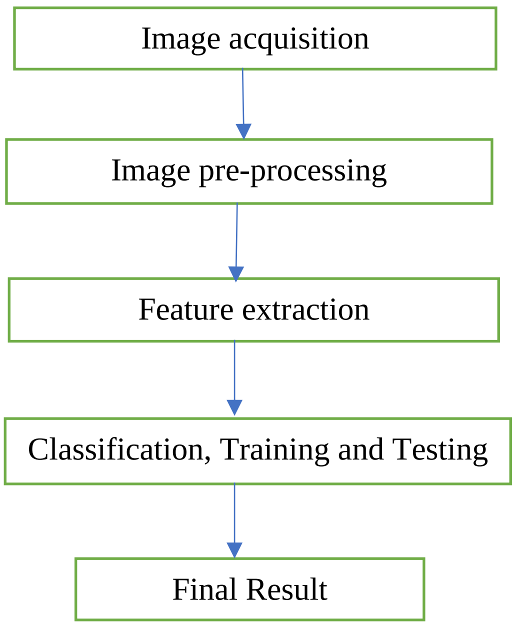


Fig. 1. Flow diagram of proposed system

CHAPTER 3

SYSTEM DESIGN AND ARCHITECTURE

3.1 Architecture Diagram

The proposed work of the system architecture is shown in Fig 3.1. The proposed system works on Automated Plant Identification.

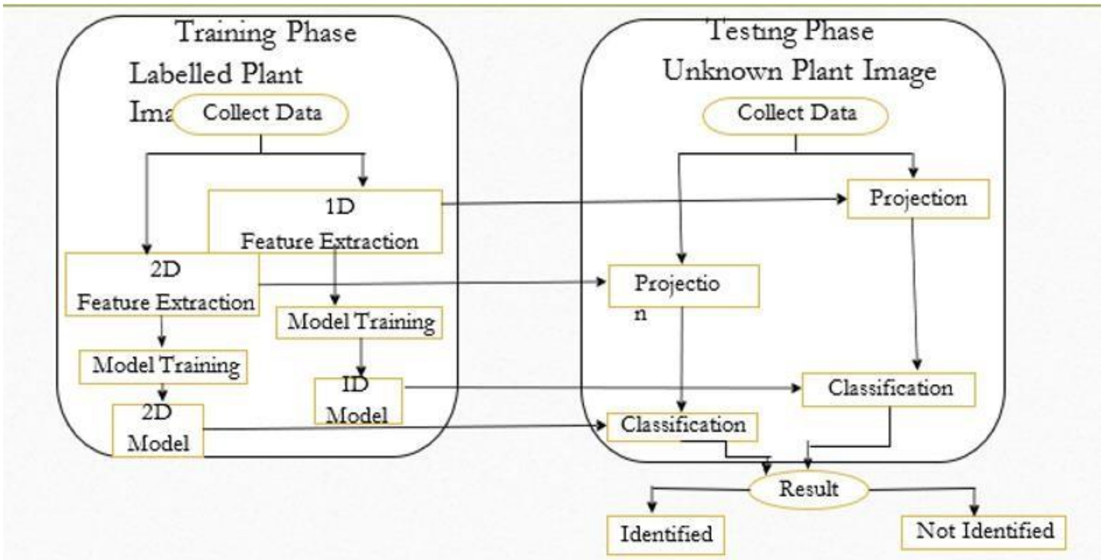


Fig.3.1 System Architecture

3.2.1 Architecture Explanation:

For leaves selection data from leave image dataset were cropped out and saved as new images with a standard resolution. For image pre-processing the uploaded plant images will be splitted into frame Processed images from previous steps were transformed into a set of parameters that describe the leaf features. Thus, the processed data is split into test and train datasets. The data loader loads these train and test datasets to the ANN classifier. By using training dataset for classification. In classification process plant identifying(shape) done in the image to identify the plant. The Model evaluation is done by confusion matrix and accuracy is determined.

3.1.2 Training Dataset (Image)

For Plant Identification datasets are collected from the GitHub. This Contains 100% real image.

3.1.3 Data Pre-processing

Dataset preprocessing includes the splitting the image. Followed by the leaf identification and identify the image. The modules work of the flow diagram is shown in Fig 3.2.

3.3.4 List of Modules:

- ➢ Image acquisition
- ➢ Image Pre-processing
- ➢ Segmentation
- ➢ Feature Extraction
- ➢ Classification

3.4.5 Modules Explanation:

Image acquisition:

- ✦ Sample images in the study consist of many plant image structure.
- ✦ Young leaves that are evidently small sized were ignored. Selected leaves were cropped out and saved as new images with a standard resolution

Image Pre-processing:

- ✦ Since all leaves are not perfectly flat, image capturing would always cast a shadow underneath the leaf.
- ✦ The shadow would disrupt the edge detection as it has a huge contrast with the background, confusing the algorithms to draw the boundary based on shadow instead of on the leaf.

Segmentation:

- The image to obtain a binary image separating the leaf from the background.
- We do this by estimating foreground and background colour distributions in the saturation-value space of the HSV colour space.

Feature Extraction:

- Processed images from previous steps were transformed into a set of parameters that describe the leaf features.
- There are four classes of features extracted in this study: morphological features (shape), texture features and histogram of oriented gradients

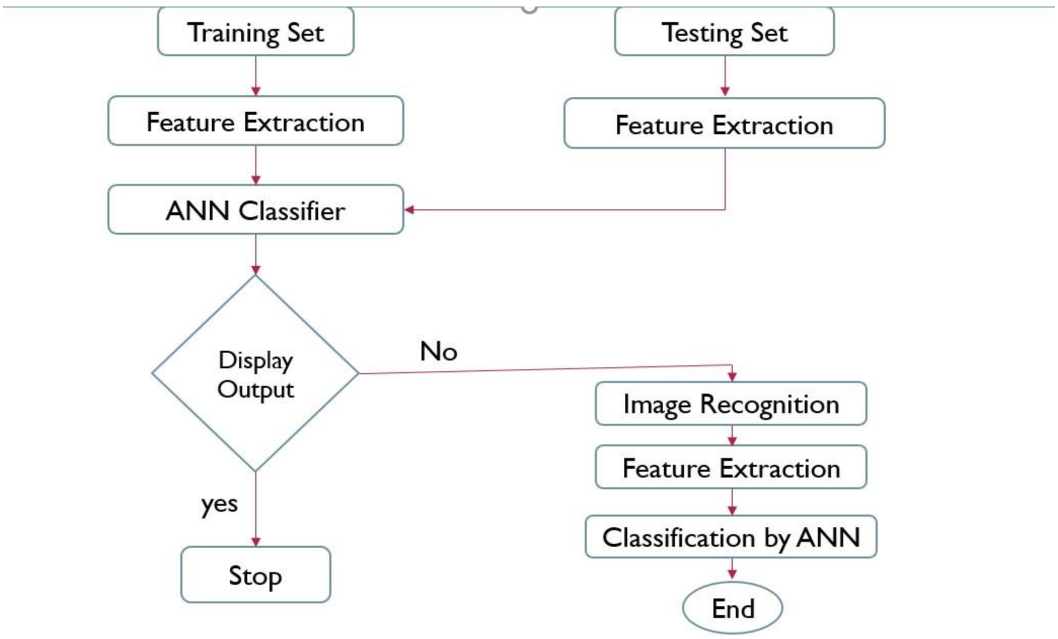


Fig.3.2 Flow Diagram

Classification:

- ANN was developed using a multilayer perceptron network to train and classify the extracted features values into three classes representing the three species used in this study, data samples were divided into training and testing sets.
- Steps in ANN implementation:

Step1: start.

Step2: Read the dataset.

Step3: Label the data if necessary.

Step4: Split the data for train (75%) and test (25%).

Step5: Train the data.

Step6: After trained the data successfully, Test the data.

Step7: Train and evaluate the model

Step8: End

3.2 Flow Chart Diagram

This is the Flowchart design of the proposed system. The Flowchart design of the prediction workflow is shown in Fig.3.2.

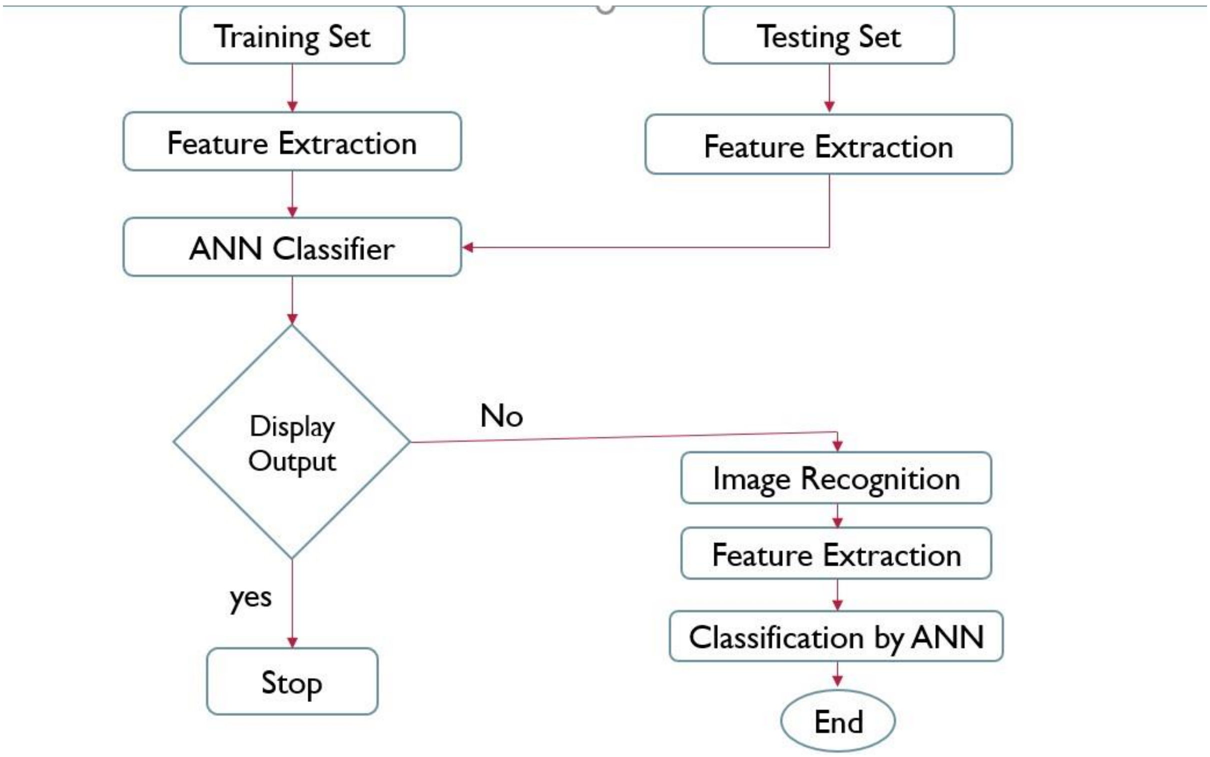


Fig.3.2. Flow Chart Diagram

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Dataset Gathering

For making the model efficient for real time prediction. We have gathered the data from plant identification. To avoid the training of the model we have considered 100% leaf identification.

4.2 Model Training

4.2.1 Train and Test Split

The dataset is split into train and test dataset with a ratio of 75% train image (1906) and 25% (476) test image. The train and test split image.

4.2.2 Data Loader

It is used to load the image and their labels with a batch size of 32.

4.2.3 Training

The training is done for 10 epochs with a learning rate of loss(0.0185),accuracy(0.9974)using the Adam optimizer.

4.3 Model Predication

The model is loaded in the application. The new leaf image for prediction is preprocessed and passed to the loaded model for prediction. The trained model performs the prediction and return if the image is a identify image along with the confidence of the prediction.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Training Analysis

The Dataset is divided as 80% for training and the remaining consider as the testing dataset. Training and Testing accuracy with 20 frames is 98% and 94%. Figure 5.1 shows the Accuracy and Loss of training and testing analysis.

```
Epoch 1/10
60/60 [==============================] - 32s 530ms/step - loss: 1.8790 - accuracy: 0.3447 - val_loss: 1.2185 - val_accuracy: 0.
5840
Epoch 2/10
60/60 [==============================] - 36s 599ms/step - loss: 1.0281 - accuracy: 0.6527 - val_loss: 0.9088 - val_accuracy: 0.
6870
Epoch 3/10
60/60 [==============================] - 35s 583ms/step - loss: 0.6816 - accuracy: 0.7723 - val_loss: 0.8446 - val_accuracy: 0.
7374
Epoch 4/10
60/60 [==============================] - 44s 742ms/step - loss: 0.4536 - accuracy: 0.8452 - val_loss: 0.8137 - val_accuracy: 0.
7458
Epoch 5/10
60/60 [==============================] - 43s 694ms/step - loss: 0.2881 - accuracy: 0.8956 - val_loss: 0.7972 - val_accuracy: 0.
7626
Epoch 6/10
60/60 [==============================] - 35s 585ms/step - loss: 0.1931 - accuracy: 0.9386 - val_loss: 0.7569 - val_accuracy: 0.
7920
Epoch 7/10
60/60 [==============================] - 43s 717ms/step - loss: 0.1217 - accuracy: 0.9654 - val_loss: 0.8216 - val_accuracy: 0.
7983
Epoch 8/10
60/60 [==============================] - 34s 560ms/step - loss: 0.0527 - accuracy: 0.9832 - val_loss: 0.9763 - val_accuracy: 0.
8004
Epoch 9/10
60/60 [==============================] - 37s 623ms/step - loss: 0.0426 - accuracy: 0.9858 - val_loss: 0.9492 - val_accuracy: 0.
```

Figure 5.1: Training and Testing Loss

5.2 Test Cases

To check the accuracy of the system we are using sample test cases.

Following are the test cases:

```
In [43]: test_dir=pathlib.Path('./train/Seledri/Seledri9.jpg')
         img = tf.keras.utils.load_img(
             test_dir, target_size=(img_height, img_width)
         )
         PIL.Image.open(str(test_dir))

Out[43]:
```

```
In [44]: img_array = tf.keras.utils.img_to_array(img)
         img_array = tf.expand_dims(img_array, 0) # Create a batch

         predictions = model.predict(img_array)
         score = tf.nn.softmax(predictions[0])

         print(
             "This image most likely belongs to {} with a {:.2f} percent confidence."
             .format(class_names[np.argmax(score)], 100 * np.max(score))
         )

This image most likely belongs to Seledri with a 15.62 percent confidence.
```

Gray Scale Image:

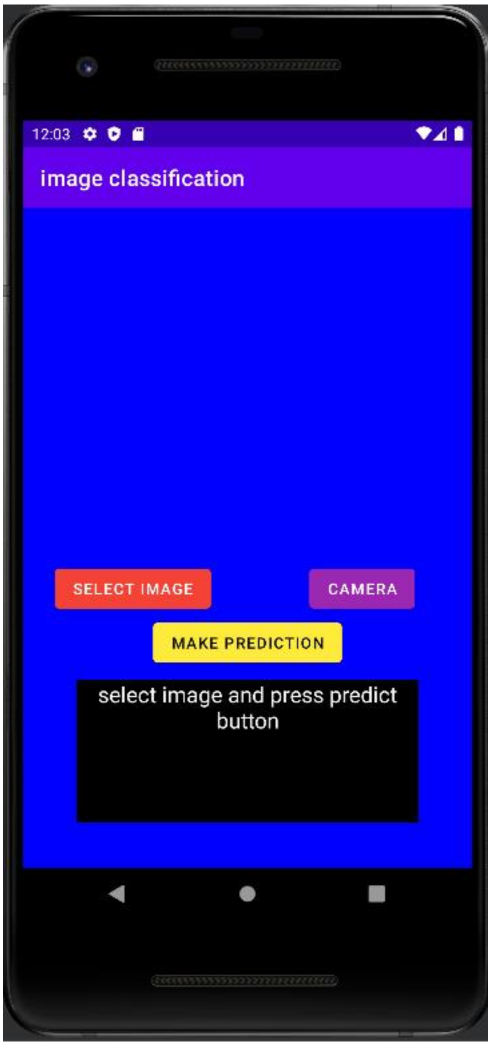
Text Feature

```
import cv2
img = cv2.imread('./train/Pepaya/Pepaya6.jpg', 0)
cv2.imshow('Grayscale Image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

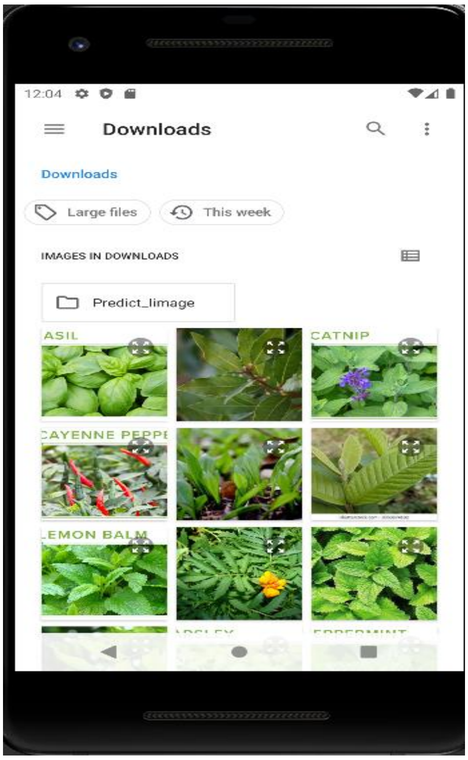
Colour Feature

```
import cv2

image = cv2.imread('./train/Pepaya/Pepaya9.jpg')

# converting BGR to RGB
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

cv2.imshow('image', image_rgb)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

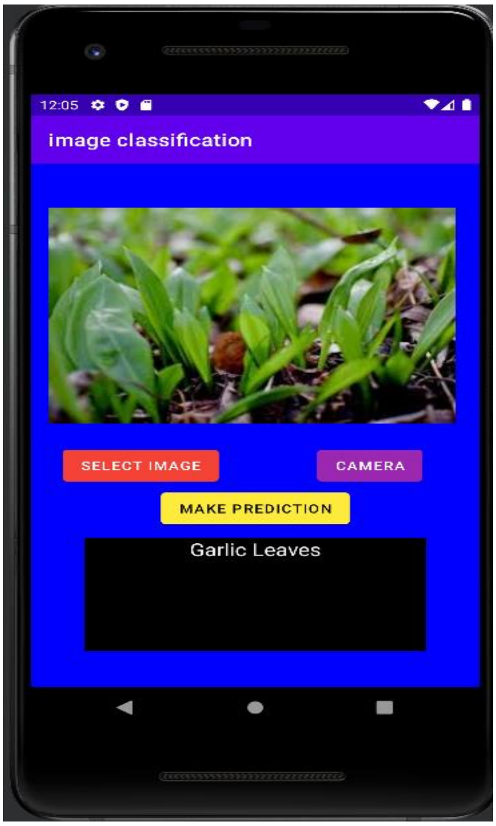
Leaf Identification App:



Select the Image:



Predict the image



CHAPTER 6
REFERENCES

- Soon Jye Kho, Sukumaran Manickam, Soraya Malek, Mogae Musleh & Surinder Kaur Dhillon (2017) Automated plant identification using artificial neural network and support vector machine, Frontiers in Life Science, 10:1, 98-107, DOI: https://doi.org/10.1080/21553769.2017.1412361

- Adel Bakhshipoua , Abdolabbas Jafari, Evaluation of support vector machine and artificial neural networks in weed detection using shape features, Received 25 January 2017; Received in revised form 20 December 2017; Accepted 22 December 2017, https://doi.org/10.1016/j.compag.2017.12.032

- Liwen Gao, Xiaohua Lin, Fully segmentation method for medicinal plant leaf images in complex background , Received 28 March 2019; Received in revised form 18 July 2019; Accepted 23 July 2019, Available online 30 July 2019, https://doi.org/10.1016/j.compag.2019.104924