



PLANT IDENTIFICATION SYSTEM USING MACHINE LEARNING

¹Regatte Sahithi Reddy, ²N. Ankitha, ³M. Lokesh, ⁴M. Anji Reddy

¹ Assistant professor in ECE Department, ^{2,3,4} Undergraduate Students of ECE Department

¹Department of Electronics and Communication Engineering

¹TKR College of Engineering & Technology, Hyderabad, INDIA

Abstract. Botanists must recognise and classify the enormous variety of plants that exist in nature. It is necessary to recognise a plant by one or more traits and associate that recognition with a name, either a common or scientific name, in order to identify it. Image processing is used to create a system that autonomously recognises plants based on their leaf structure. Additionally, plants are undergoing evolutionary change, which has an effect on categorization and identification. Botanists travel to remote locations and forests to identify and classify plants. As a result, an effort is made to create an automatic identification system where a person can take a picture of a leaf and submit it to a website to learn more about the plant. Our goal in this effort is to use neural networks to digitally identify different plant species using an image of a single leaf. Convolutional neural networks, Tensorflow, and Keras will be used to approach our assignment. The aforementioned method produces outcomes that are satisfactory and highly accurate.

Keywords: *Tensorflow, Keras, Machine Learning, Convolution Neural Networks.*

Index Terms – *Piezoelectric sensors, Arduino, Atmega 328 Microcontroller, RFID Reader, RFID Tags*

I. INTRODUCTION

Botanists typically classify plants based on their floral components, such as their blooms, fruits, and leaves. Flowers and seeds should not be used to identify plants because of their sporadic appearance. On the other hand, leaves are a good option for automatic identification of plants because they are readily accessible, last a longer time, and are plentiful. Even for botanists, identifying plants can be challenging due to the vast number of varieties. A plant identification tool can be useful in this situation. Accurate plant identification can help farmers and gardeners learn about the species, including how it develops and how to take care of and protect it from pests and diseases.

II. EXISTING SYSTEM

The current systems are based on medicinal plants and flowers, and the flower-based identification system gives the top three flower images that are most similar to each other. If their colour is distinct, the accuracy is good. However, if colours are identical, it might be inaccurate to classify the picture. High precision is achieved when identifying medicinal plants, but leaf vein and edge characteristics are not taken into account.

III PROPOSED SYSTEM

We used the Mendeley dataset, which includes high-quality images of 10 different species with multiple samples per species, for the suggested plant identification method. With the help of the Flask framework, we created a web application that allows users to submit and upload plant leaves. The system gives plant information for that leaf after classifying the input leaf image. In order to accomplish this, a Convolutional Neural Network (CNN) model is trained to correctly identify the plant.

IV BLOCK DIAGRAM

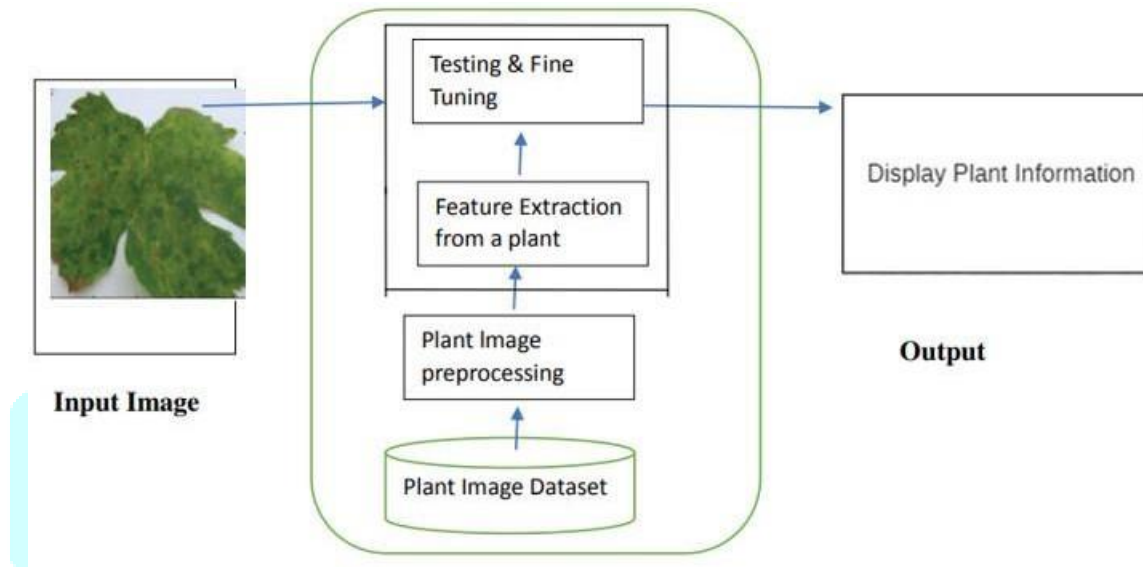


Fig 1: BLOCK DIAGRAM

Here, we offered a technique for determining the name of the plant from its leaf. The suggested method divides the leaf image data into labels based on the vein and leaf edge characteristics. The label of the user-uploaded leaf image is predicted by the CNN algorithm. As output, the label's details and the plant name are shown.

V SEQUENCE DIAGRAM

In the Unified Modeling Language (UML), a sequence diagram is a type of interaction diagram that demonstrates how and in what order processes engage with one another. It is a Message Sequence Chart design. Event diagrams, event situations, and timing diagrams are other names for sequence diagrams.

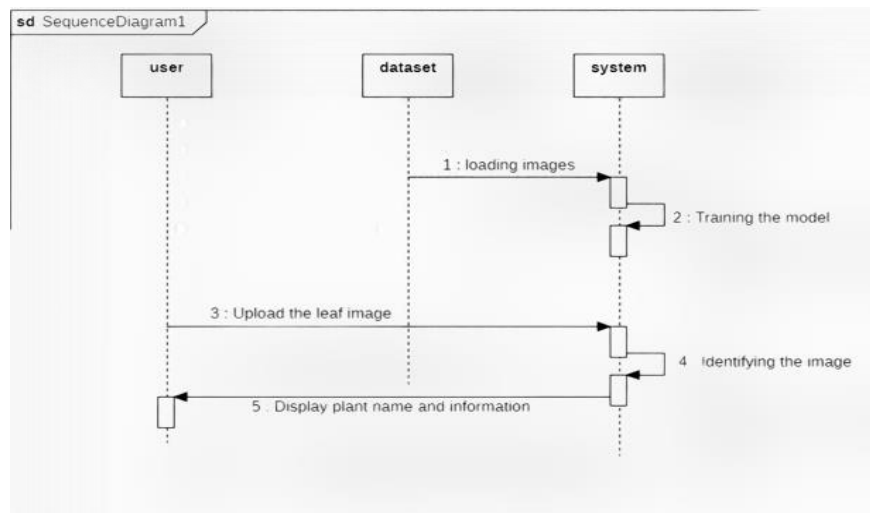


Fig 3: Sequence Diagram

VI METHODOLOGY

Module Description:

Here there are two modules in this project

1. User
2. System

USER

Users are the farmers and garden enthusiasts who can upload the leaf and acquire the species knowledge.

SYSTEM

The system will identify and train the images. It recognises the label on the leaf that the user has submitted and provides information about the plant associated with that label.

We used the Mendeley dataset, which contains high-quality images of 10 distinct species with multiple samples per species, for this plant identification method. With the help of the Flask framework, we created a web application that allows users to submit and upload plant leaves. The system gives plant information for that leaf after classifying the input leaf image. For this, an accurate plant identification model for the Convolutional Neural Network (CNN) is created.

VII Working:

Algorithm:

- Step 1: Import Required libraries.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf
```

- Step 2: Initializing CNN and adding a convolution layer.

```
# Initialising the CNN
model = Sequential()

# Step 1 - Convolution
model.add(Conv2D(32, (3, 3), input_shape=(128, 128, 3), activation='relu'))
```

Since there are several levels that must be built into a CNN in order, we must start with a sequential class.

The first convolutional layer is then added.

- Relu activation function: Since pictures are non-linear, relu activation function is used to introduce non-linearity after convolutional operation. Rectified linear activation function is known as Relu. If the input is positive, the relu function will output the value immediately; otherwise, it will output 0.
- Input Shape: The picture size indicated by this argument is 120*120*3.

➤ Step 3: Pooling Operation

```
model.add(MaxPooling2D(pool_size = (2, 2)))
```

Following CNN initialization, the pooling procedure must be used. Downsampling the picture is what pooling entails. The feature maps' proportions are decreased by using the pooling layer. As a result, the neural network's computation is reduced and the number of factors to learn is reduced thanks to the pooling layer.

➤ Step 4: To add a second convolution layer, follow steps 2 and 3 again.

➤ Step 5: Flattening Operation

```
model.add(Flatten())
```

- The flattening process turns the dataset into a 1-D array for input into the fully connected layer, the following layer.

➤ Step 6: Fully connected layer

```
model.add(Dense(units = 128, activation = 'relu'))
model.add(Dense(units = 10, activation = 'sigmoid'))
```

- The neural network's input is the flattening operation's result. Here, we're using a dense class from the Keras library to build an output layer and completely connected layer.

➤ Compiling CNN Model

```
# Compiling the CNN
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
```

- **Loss function:** The categorical_crossentropy loss function, which is employed in the classification assignment, is what we are using.
- **Optimizer:** To change the neural network weights and learning rate, we are using the Adam Optimizer. By minimising the function, optimizers can tackle optimisation problems.
- **Metrics:** To assess the effectiveness of the Convolutional neural network algorithm, we use Accuracy as a measure.

Fitting Model on Training

Model Fitting on Training In 50 iterations, we are fitting the CNN model on the training dataset. The training and evaluating stages vary depending on the size of the test and training sets in each iteration.

```
model.fit(training_set, steps_per_epoch = 20,
          epochs = 50,
          validation_data = valid_se
```

1. Convolutional Neural Network Step 1: convolutional operation

The convolution procedure is the first component of our strategy. We will discuss feature detectors, the neural network's filters, in this stage. Additionally, we'll talk about feature maps, their parameters, the layers of pattern recognition, and how the results are mapped out.

The Rectified Linear Unit or ReLU will be used in the second portion of this process. We will discuss ReLU layers and examine the role of linearity in Convolutional Neural Networks.

Although it's not essential to comprehend CNN's, it wouldn't hurt to take a quick lesson to advance your

knowledge.

Step 2: Pooling Layer

We'll discuss pooling in this section and learn precisely how it typically operates. But max sharing will be the central concept in this situation. However, we'll discuss a variety of strategies, including mean (or sum) sharing. This section will conclude with an interactive visual example that will undoubtedly clarify the entire idea for you.

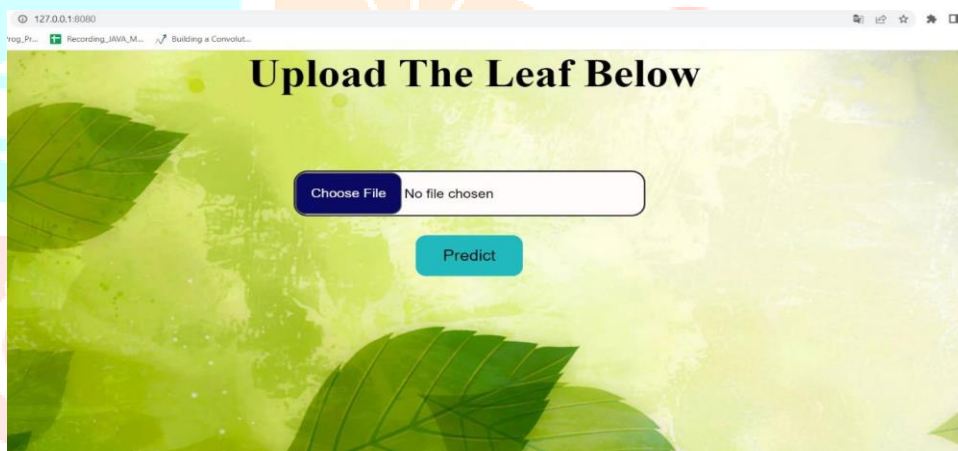
Step 3: Flattening

Here is a short explanation of the flattening procedure and how, when using convolutional neural networks, we go from pooled to flattened layers. The flattening process turns the dataset into a 1-D array for input into the fully connected layer, the following layer.

Now that we have completed the three stages, we have a pooled feature map. After two stages, we are now flattening our output into a column. because we must add this 1-D input to a layer of an artificial neural network.

Step 4: Fully connected layer and output layer

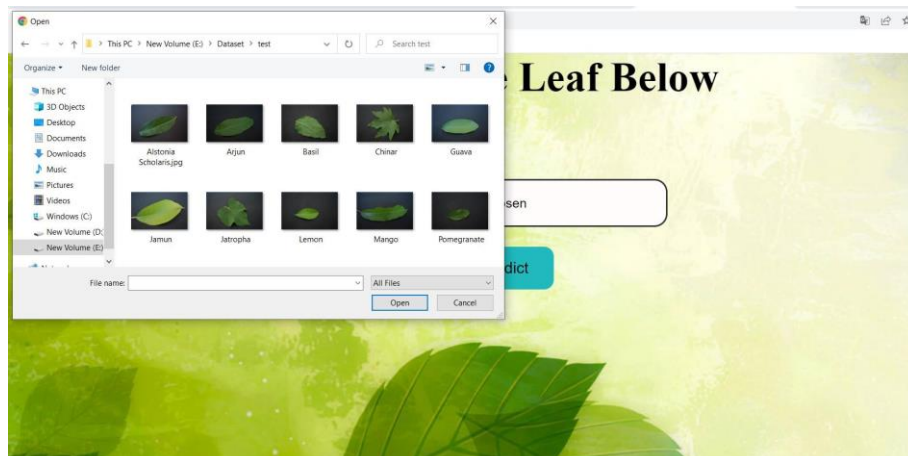
The neural network's input is the flattening operation's result. The convolutional neural network becomes more sophisticated and capable of classifying pictures as a result of the artificial neural network's goal.



RESULT:

Fig:9 Home Page

Above screen is the home page of the web application where user can upload image. By clicking on "Choose



File" button user will directed to the screen below.

Fig:10 Upload image

Here user can browse the image from internal storage and upload the file and then click on “Predict”.

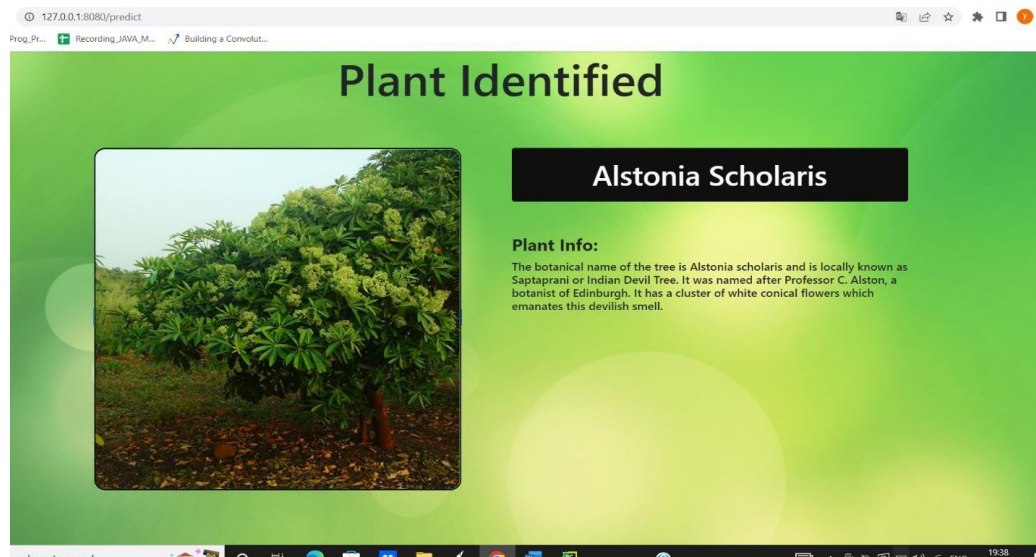


Fig:11 Alstonia Scholaris

After uploading the image and by clicking on “Predict” button user will get the information according to the image uploaded.

Conclusion and Future Enhancements

A web application for identifying leaf species has been demonstrated in this project. It is founded on a collection of leaf characteristics that have shown promise in leaf datasets. 90% greater accuracy was attained. This programme is helpful to both experts and non-experts interested in learning about plants because of the accuracy of the identification.

Future Enhancements:

We would like to develop an user friendly Android Application.

References:

- [1] N. Boujemaa, J. Fauqueur, M. Ferecatu, F. Fleuret, V. Gouet, B. Lesaux, and H. Sahbi. Ikona: Interactive specific and generic image retrieval. In International workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR), 2001.
- [2] H. Goëau, P. Bonnet, A. Joly, I. Yahiaoui, D. Barthélemy, N. Boujemaa, and J.-F. Molino. The IMAGECLEF 2012 Plant identification Task. In CLEF 2012, Rome, Italy, Sept. 2012.
- [3] A. Joly and O. Buisson. Random maximum margin hashing. In CVPR, pages 873–880, 2011.
- [4] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In 12th European Conference on Computer Vision (ECCV 2012), pages 502–516, Florence, Italy, Oct. 2012.
- [5] S. Mouine, I. Yahiaoui, and A. Verroust-Blondet. Advanced shape context for plant species identification using leaf image retrieval. In Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, pages 49:1–49:8, 20