Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

Implementation Of Plant Disease Detection Using CNN Algorithm

[1]Mrs Keerthana M M, [2]Mrs Hamsa A S

[1] [2] Assistant Professor, Department of Computer Science and Engineering, ATME College of Engineering, Mysuru, Karnataka, India

Abstract: This paper presents an android application to identify the plant by leaf using CNN and Machine learning techniques. The Application detects the crop and disease associated with the crops using Deep Learning. Plant identification based on the leaf is becoming one of the most popular and interesting trends nowadays. Each leaf carries unique information, which can be used for identification of plants. The paper aims to help farmers in agriculture field to improve their farming techniques leading to better crop yields as well as better stewardship of the environment. The early detection of disease in plants helps farmer to use appropriate pesticides so proper growth of the plants and yields can be improved. Computer vision has grown recently in the past few years with advent of Deep Learning algorithms. Convolutional neural networks have become the state-of-the-art algorithm for image detection and classification tasks. With the help of popular modern deep learning frameworks Keras,

Keywords—CNN, Machine Learning, Deep Learning, Keras

1. Introduction

Plant is the basic essential source for living organisms on the earth. It is the backbone of the food chain. Plants are significantly vital safeguarding of the environment. Nevertheless, it is tricky and chief mission to identify the plant varieties on earth. The critical conditions is that many plants are at the threat of extermination. According to the hypothesis of plant categorization, plants are fundamentally classified based on shapes of leaves, flowers and fruits. The illness on the plant can affect the measure of a horticultural thing. This can make a negative impact on the economy of the country whose basic strength is the horticulture. From this the acknowledgment of the disorder in the earlier stages is essential to avoid the hardships of the finance. Commonly the strategies that are embraced for checking and the chiefs of plant leaf disorder are manual. One such critical strategy is independent eye insight. The essence of this procedure is steady seeing of the field by an individual having pervasive data about the plants and their relating sickness. Plant illness discovery is the method that is applied to distinguish the sickness from the information of plant leaf. The properties of the input picture can be examined into shading and textural structure, the shading process contribute to terms red, green and blue. The textural highlights of the picture of the pack of pixels. In any case the various kinds of AI calculations are used in identifying plant leaf infections and provide the solutions.

2. Literature Survey

[1] V J Madhumati-Plant identification based on leaf is becoming one of the most interesting and a popular trend. Each leaf carries unique information that can be used in the identification of plants. In this process, the leaf image needs to be pre-processed accordingly to extract the various features of the plant leaf. In this paper, they present the identification of plants based on leaf features using Multiclass SVM (MSVM) as a classifier. Publisher-IEEE, Year: 2017

[2] Qian Cheng-This paper presents an Android application to automatically identify plant species using a single leaf image as input. At the pre-processing phase, we proposed an improved segmentation method to eliminate the noise caused by capturing on non-uniform background so we can obtain the binary image which only contains the leaf shape. Several morphological features and moment invariants descriptors were extracted as inputs of a joint classifier which combines the back propagation neural network (BPNN) with k-nearest neighbour (KNN) to distinguish 220 species of plants. The output of the classifiers are the top species that best match the query leaf image. Publisher-IEEE, Year: 2017.

[3] S Santhosh, B Gokilavani-There are several million species look similar, but they are actually not genetically the same, may belong to different families. It is very important to identify the plant species. Often taxonomist face difficulties while grouping plant species. The key focus of this paper is to identify the variety of species by classifying the leaves. This paper takes leaf images features to classify the species by using Support Vector Machines (SVM). This classification helps to recognize and identify the population of endangered and extinct species for preservation. Publisher: IEEE, Year: 2019.

[4] Tatsuhiro Akiyama-The ability to identify plant types is important when conducting vegetation surveys. We propose a mobile application using Convolution neural networks (CNN) that will help beginners identify plant species. In this we compare three CNN models, VGG19, Mobile Net and Mobile NetV2. Publisher: IEEE ,Year: 2019.

[5] Md Maiedul Islam, A K M Shahahriar Azad Rabby-Plants are everywhere around the Earth. Plant identification is very important problem for environment protection and exploration. But getting to know various plants requires considering a large number of features which might be easy for a botanist but not easy for common people. Plant identifier which uses a leaf to identifying plants will help many people. In this paper Convolution Neural Network (CNN) technique is used to classify plants using leaf images. Using Adam optimizer and Automatic learning rate reduction technique the model gave promising accuracy. This system was trained on 3600 RGB leaf images of 2categories for 6 different plant species. The model reported promising results with validation accuracy was 95.86% and training accuracy was 96.54%. Different preprocessing techniques such as background whitening, noise removal are used. Publisher: IEEE ,Year: 2019

3. Existing System

The identification of plants by conventional keys is complex, time consuming and due to the use of specific botanical terms frustrating for non-experts. This creates a hard to overcome hurdle for people interested in acquiring species knowledge. Today, there is an interest in automating the process of species identification. There are many plant species and crop identification applications that are existing and the technology stack they are using is based on image processing. Image based methods are considered a promising approach for plant identification. A user can take the picture of a plant on the spot with the build in camera of his/her smartphone device and analyse it with an installed recognition application to identify the species or at least to receive a list of possible species if single match is not detected. But there are many more affecting factors that would make the image-based classification weaker with unpredictable output.

4. Proposed System

We present a plant identification system for automatically identifying the plant in each image. In addition to common difficulties faced in object recognition, such as light, pose and orientation variations there are further difficulties particular to this problem, such as changing leaf shapes according to plant age and changes in the overall shape due to leaf composition. Our system uses a rich variety of shape, texture and colour features, some being specific to the plant domain. The proposed technique involves three major steps, such as pre-processing, feature importance, and classification. Initially, the data will be pre-processed using the resampling technique. The main objective of this project is to classify a plant seedling and it's health status from weed using Convolutional Neural Networks and develop an Android Application that has ability to classify it effectively. Also Plant disease has been one of the major threats to food security because it dramatically reduces the crop yield and compromises its quality. Accurate and precise diagnosis of diseases has been a significant challenge. According to the Food and Agriculture Organizations of the United Nations (UN), transboundary plant pests and diseases affect food crops, causing significant losses to farmers and threatening food security. We detect if there are any plant diseases that effects the crop growth and suggest the pesticides which are suitable for the crops that are affected.

5. Methodology

A Convolutional Neural Network is a Deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a convNet is much lower as compared to other classification

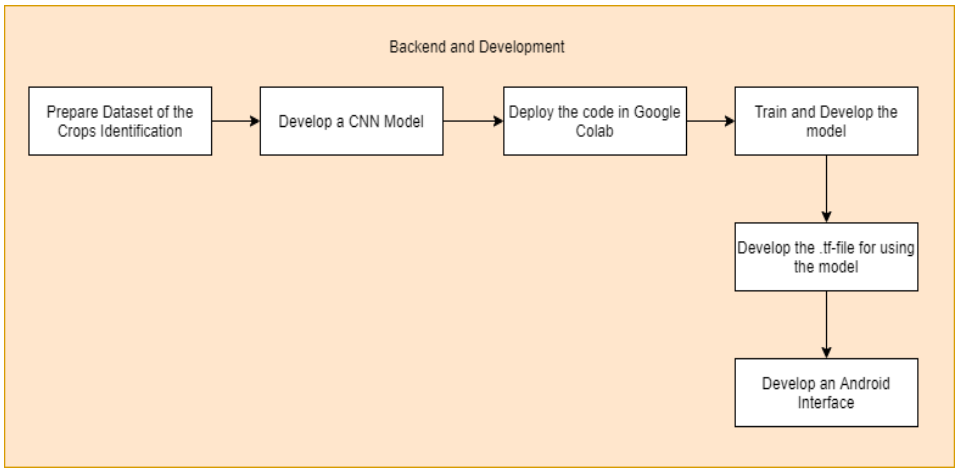
algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets can learn these filters/ characteristics.

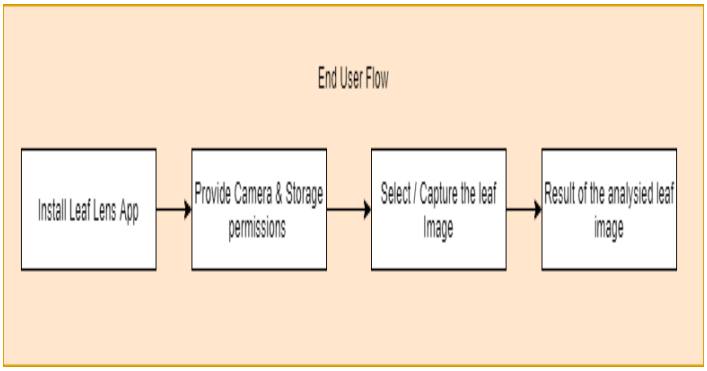
The architecture of a ConvNet is analogous to that of the connectivity pattern of neurons in the human brain and was inspired by the organization of the visual cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the respective field. A collection of such fields overlap to cover the entire visual area.

Features:

1. Prepare the dataset of the plant images to be specific plants leaves.
2. Flatten the input image dimensions to 1D. (width pixels x height pixels)
3. Normalize the image pixel values. (Divide by 255)
4. One-Hot Encode the categorical column.
5. Build a model architecture (Sequential) with Dense layers.
6. Train the model and make predictions.

We will be using CNN model algorithms to classify the plants leaves.





System Design

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phases particularly testing and maintenance.

The design activity often results in three separate outputs –

- Architecture design.

- High level design.
- Detailed design.

6. Architectural Design

Architecture focuses on looking at a system as a combination of many different components, and how they interact with each other to produce the desired result. The focus is on identifying components or subsystems and how they connect. In other words, the focus is on what major components are needed.

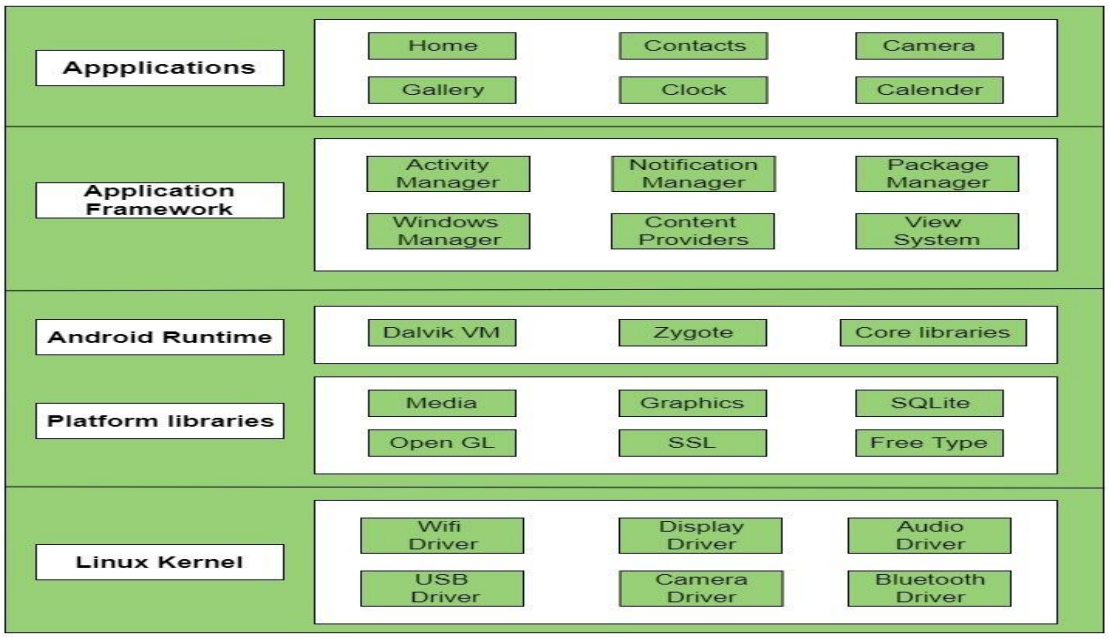


Fig 1: Architectural Design

High Level Design

High-Level Design (HLD) Document is to add the necessary detail to the current project, description to represent a suitable model for coding.

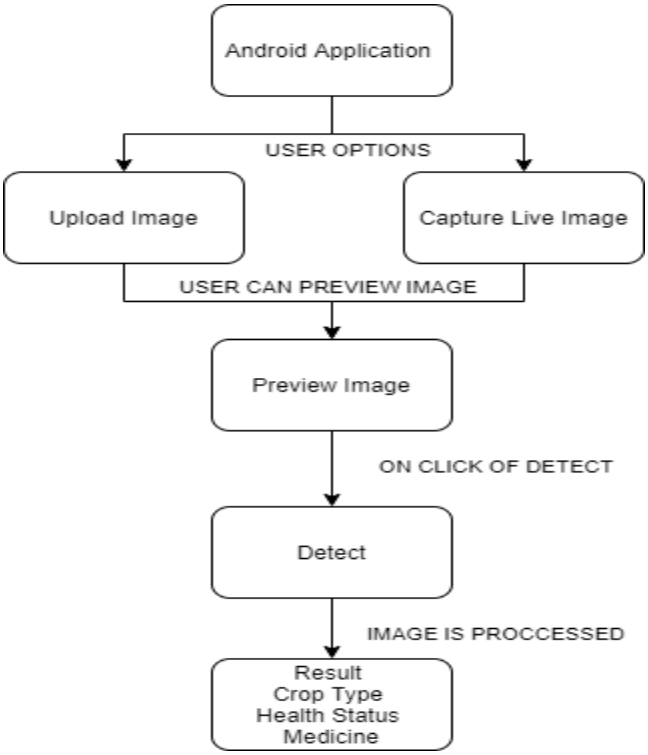


Fig 2: Flow chart of the application

The data flow of our Android application works in the backend with the flow shown below.

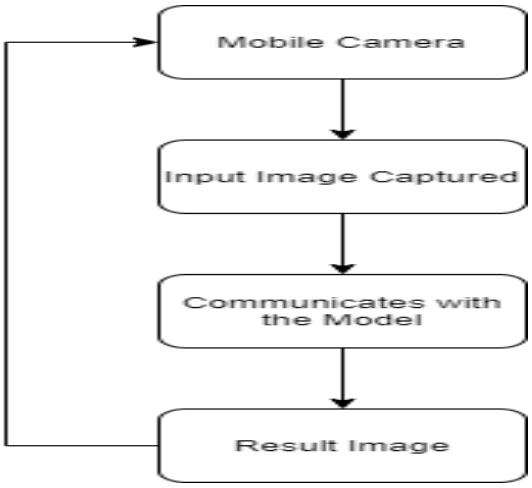


Fig 3: Data Flow Diagram

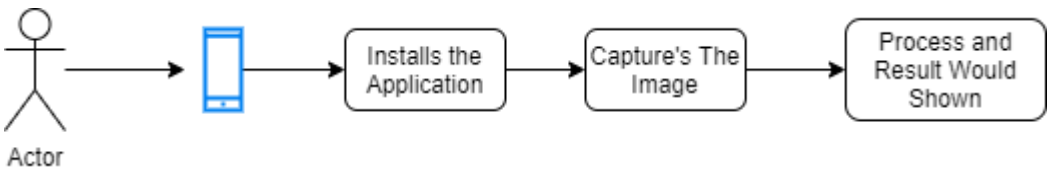


Fig 4: Activity Diagram for Plant Doctor

Preparing the Machine Learning Model

The dataset was taken from the open source https://www.kaggle.com/emmarex/plantdisease where we have collected all dataset which was needed to develop the calcification model. We have used the CNN machine learning concept to develop our model. The Inception network was an important milestone in the development of CNN classifiers. The Inception network on the other hand, was complex (heavily engineered). It used a lot of tricks to push performance; both in terms of speed and accuracy. Its constant evolution lead to the creation of several versions of the network. We have used the inception model 3.

This image represents the placement of the input devices and demonstrates the input received.

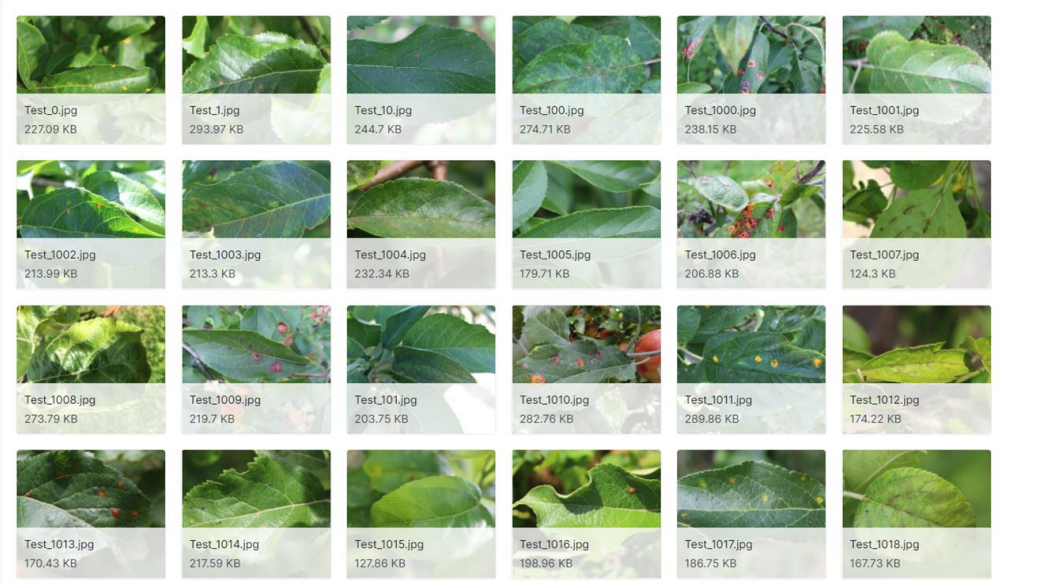


Fig 5: Plant Leaf image

```
[ ]  print("Building model with", MODULE_HANDLE)
     model = tf.keras.Sequential([
         feature_extractor,
         tf.keras.layers.Flatten(),
         tf.keras.layers.Dense(512, activation='relu'),
         tf.keras.layers.Dropout(rate=0.2),
         tf.keras.layers.Dense(train_generator.num_classes, activation='softmax',
                               kernel_regularizer=tf.keras.regularizers.l2(0.0001))
     ])
     #model.build((None,)+IMAGE_SIZE+(3,))

     model.summary()
```

```
Building model with https://tfhub.dev/google/tf2-preview/inception_v3/feature_vector/2
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
keras_layer (KerasLayer)     (None, 2048)              21802784
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 512)               1049088
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 38)                19494
=================================================================
Total params: 22,871,366
Trainable params: 1,068,582
Non-trainable params: 21,802,784
_________________________________________________________________
```

Development of the Android Application for the interface.

TensorFlow Lite Converter converts a TensorFlow model to TensorFlow Lite flat buffer file(. tflite). TensorFlow Lite flat buffer file is deployed to the client, which in our cases can be a mobile device running on iOS or Android or an embedded device. Thereby the above developed model is been converted to a .tflite file which is been used to embed or interact with our Android Application.

```
class MainActivity : AppCompatActivity() {
    private lateinit var mClassifier: Classifier
    private lateinit var mBitmap: Bitmap

    private val mCameraRequestCode = 0
    private val mGalleryRequestCode = 2

    private val mInputSize = 224
    private val mModelPath = "model.tflite"
    private val mLabelPath = "plant_labels.txt"
    private val mSamplePath = "soybean.JPG"
```

7. Implementation

Working of a Machine Learning Model

A Machine Learning system learns from historical data, builds prediction models, and whenever it receives new data, predicts the output for it. The accuracy of the predicted output depends up on the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately. Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it we just need to feed the data to generic algorithms, and with the help of these algorithms, the machine builds the logic as per the data and predicts the output.

Kaggel

Kaggle, a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges. One of the biggest hurdles of democratizing Deep Learning has been the need for a GPU accelerated environment. If we want to spend our time efficiently testing and learning across problem statements we need a GPU-based server. But these GPU based environments across platforms like GCP, AWS, Azure, Paper-space can amount to a huge cost. This cost when combined with the cost of storage and if we are using instances that are persistent may really add to the burden on our money bag. But some amazing news came in when Kaggle and Google Colab introduced GPU-enabled kernels which can be now used to decipher and solve deep learning problems across the problem statements without the need to buy a GPU.

Python Language

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides.

Kotlin Android Development

Kotlin is a general purpose, free, open source, statically typed “pragmatic” programming language initially designed for the JVM (Java Virtual Machine) and Android that combines object-oriented and functional programming features. It is focused on interoperability, safety, clarity, and tooling support. Versions of Kotlin targeting JavaScript ES5.1 and native code (using LLVM) for a number of processors are in production as well.

Jupyter Notebook

The IPython Notebook is now known as the Jupyter Notebook. It is an interactive computational environment, in which you can combine code execution, rich text, mathematics, plots and rich media. The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. The IPython notebook combines two components: A web application: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations and their rich media output.

8. Results

An Offline Mobile Application for Crop Identification checks the disease along and recommends Fertilizers. This is our application's landing page. Where user has the options to interact with our application.

1. User can Select the existing image from the gallery
2. User can capture real Time Image and upload
3. Once the image is uploaded/Selected User can detect
4. Preview Screen of the image uploaded/selected

Fig 6: Front Screen Plant Doctor

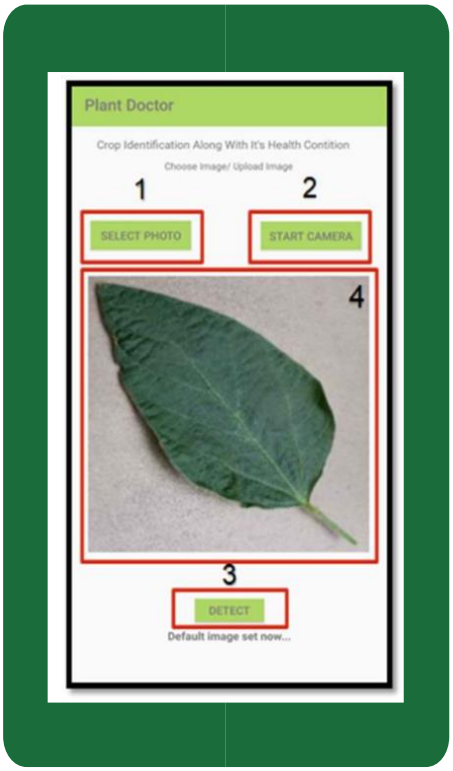


Fig 7: Loading a leaf image

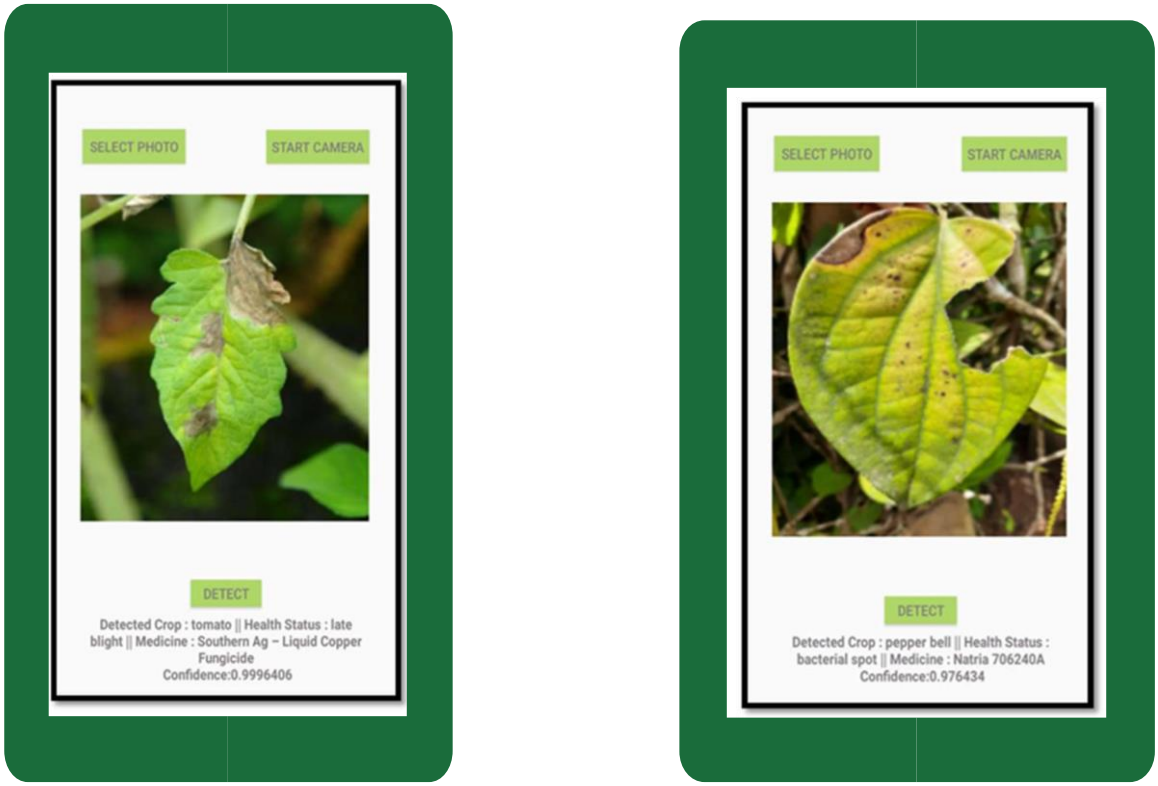


Fig 8 : Results of two leaf images which with name, health status and suggested medicine

Example of Captured Image Live:

In Fig 8, we have uploaded the image using the start camera option and captured a leaf (Pepper Plant's)

Result:
Detected Crop : Pepper Bell
Health Status : Bacterial Spot
Medicine : Natria 706240A

Example of Existing Image

Here, we have uploaded the image using the existing image from the gallery. Uploaded Image (Tomato Plant's)

Result:
Detected Crop : Tomato
Health Status : Late blight
Medicine : Liquid Copper Fungicide

Example of Non leafy image:

Here, we have uploaded the image of a non-leaf. If they are no Leaf in the uploaded image this would be outputted.

Fig 9 : Non leaf image as input

Result:
Background- No leaf detected.

9. Conclusion

In this paper, we include a plant identification system for automatically identifying the plant using leaf image by using CNN and Machine learning techniques. In addition to common difficulties faced, our system uses a rich variety of shape, texture and colour features, some being specific to the plant domain. Plant disease has been one of the major threats to food security because it intensely reduces the crop yield. Therefore, accurate and precise diagnosis of diseases has been done in the early stages and proper pesticides are suggested so it does not affect food crops, causing significant losses to farmers and threatening food security. We are using local datasets to train our model where the user provides the image of the plant and with the click of the button; the app feeds the image to the trained model and gives the result.

Reference

[1] Wenjiang Hung, Qingsong Guan, Juua luo, Jingcheng Zhang, Jinling Zhao, Dong Liang, Linsheng Huang, and Dongyan Zhang, "New Optimized Spectral Indices for Identifying and Monitoring Winter Wheat Disease", IEEE journal of selected topics in applied earth observation and remote sensing, Vol. 7, No. 6, June 2018.

[2] Dr K Thangadurai, K Padmavathi, "Computer Vision image Enhancement for Plant Leaves Disease Detection", 2017 World Congress on Computing and Communication Technologies.

[3] Monica Jhuria, Ashwani Kumar and Rushikesh Borse, "Image Processing For Smart Farming: Detection Of Disease and Fruit Grading", Proceeding of the 2017 IEEE Second International Conference on Image.

[4] Zulkifli Bin Husin, Abdul Hallis Bin Abdul Aziz, Ali Yeon Bin, Md Shakaff Rohani Binti S Mohammed Farook, "Feasibility Study on Plant Chili Disease Detection Using Image Tecniques",2015 Third International Conference on Intelligent Systems Modelling and Simulation.

[5] Mrunalin R Badnakhe, Prashant R Deshmukh, " Infected Leaf Analysis and Comparison by Otsu Threshold and k-Means Clustering", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, March 2017.