

# Android Application on E-commerce & Plant Disease Identification using Tensor Flow

Shrikant S. Gurav, Snehal A. Patil, Shubhangi L. Patil, Priti P. Desai, Prajakta B. Badkar,  
Department of Computer Science & Engineering  
Sant Gajanan Maharaj College of Engineering, Gadhinglaj, Kolhapur

**Abstract:-** In this Digitized world farmers can do day-to-day operations using online means such as buy and sell of saplings. The Modern world can connect back to the motherland using PlantIP app which helps farmers to digitize themselves and form a community. To identify and detect plant diseases, which are a major factor in the loss of crop yield in agriculture and crop production, we employ a powerful ML library provided by Google. The project uses live weather forecast for a week with updated GPS tracks using satellite data from Openweathermap. Parts of the distributed run-time system for the plant disease detector are arranged to execute on user-side mobile devices and cloud-side centralized servers. The CNN deep learning model and the Intermediate Representation (IR) model that operate on mobile devices are both described in Layer 1 of the system. The user interface is depicted in Layer 2 and created as an Android app to make it easy for system users (shown in Layer 3) to engage with the system.

**Keywords:-** Image processing, image acquisition, classification, segmentation.

## I. INTRODUCTION

PlantIP with the approach of image processing Technology and connecting to Community of agriculture experts’ know-how or help fellow farmers with their knowledge and experience: Which will make the PlantIP Community, the largest network for farmers worldwide. Where they can talk about new methodologies and get good yields with our App.

This paper consists of details about the app and how it helps farmers to reach out to customers as well as form a community by providing help in live weather tracking and week planners.

Parts of the distributed run-time system for the plant disease detector are arranged to execute on user-side mobile devices and cloud-side centralised servers. The CNN deep learning model and the Intermediate Representation (IR) model that operate on mobile devices are both described in Layer 1 of the system. The user interface is depicted in Layer 2 and created as an Android app to make it easy for system users (shown in Layer 3) to engage with the system. literature survey & use cases

After the recent Lockdown world has turned its clock to go faster in world of digital media, People are more familiar and comfortable ordering online goods. Lots of farmers like trying new things to have more crop yield. The field of agriculture is in decline, yet there have been a number of

advancements that have brought about genuine financial advantage for the farmers and one in every one of them can use apps like PlantIP for increasing their reach to customers and selling veggies and fertilizers, small gardening plants and much more using easy customization of selling products.

The new era of Agriculture brought lots of new advancement in technologies as well as lots of untrack diseases which hard to keep it up with here comes the use of PlantIP the one stop solution

To detect any disease stored on cloud and processed using google Teachable machine [1].

The neural network is formed to optimize the shortest path [2] to guess the disease and finding a cure and information to handle disease is provided using PlantIP.

We have used more than thirty-eight types of disease which are more common for basic farming and home gardening yields.

We use Open weather map for live updates of weather in the app as nature is so unpredictable these days, the increasing global warming needs proper plans for the week without interposition of any preventing measures for farming.

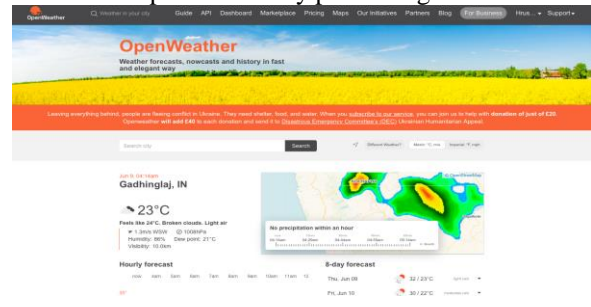


Fig. 1

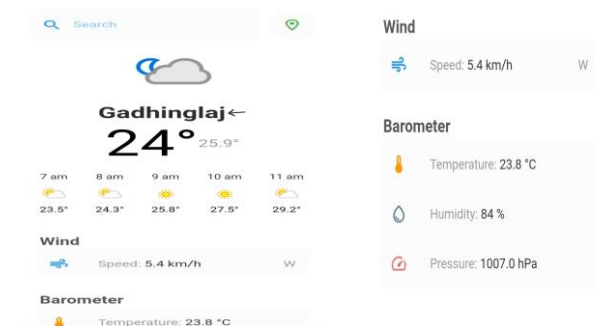


Fig. 2

- A UI of use case of Open weather map in Plantip App.
- We use a scheduler planner to remind important updates regarding the weather forecast and notify using OneSignal notification.
- The users can never miss what's important using plantip as the scheduler is highly customizable for location-based notifications from user frequently used locations.

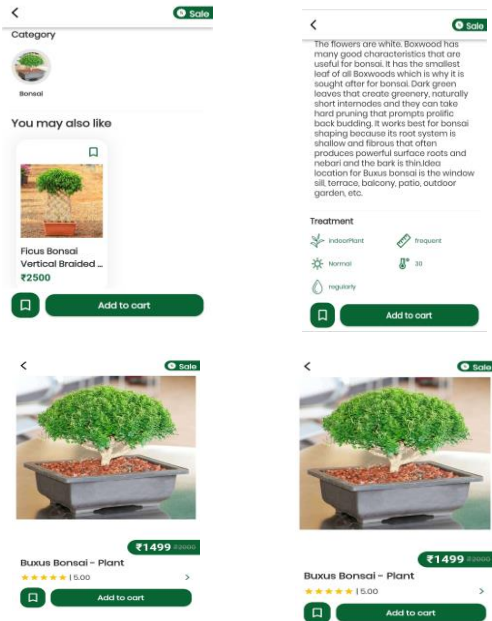


Fig. 3: Plant IP app screenshots

- The E-commerce buy and sell helps direct relations between farmers and customers selling 100 of categories including plants, fertilizers, small farming equipment etc.
- The smart suggestions help buying more easily in plantip as users can perform all operations like review and request customizations.

**II. DISEASE TYPES & CLASSIFICATION**

**A. Tomato-Early blight**



Fig. 4: Tomato Early blight

The fungus *Alternaria solani* is the source of the widespread tomato disease known as early blight. Almost all tomato plant parts, including the leaves, stems, and fruits, are susceptible to it. Although the plants might not perish, they will likely be weaker and produce fewer tomatoes than usual. Although it can affect seedlings as well, early blight often affects older plants.

**B. Late Blight Potato Leaf**



Fig. 5: Late Blight potato Leaf

The oomycete microbe *Phytophthora infestans*, which resembles a fungus, causes late blight in potato plants. This unquestionably dangerous disease can affect potato leaves and tubers at any stage of crop growth.

**C. Tomato Bacterial Spot**



Fig. 6: Tomato Bacterial Spot

*Xanthomonas vesicatoria*, *Xanthomonas euvesicatoria*, *Xanthomonas gardneri*, and *Xanthomonas perforans* are the culprits behind tomato bacterial spot. On contaminated seed and transplants, which may or may not exhibit symptoms, these bacterial infections can be introduced into a garden.

Dataset used to train model:

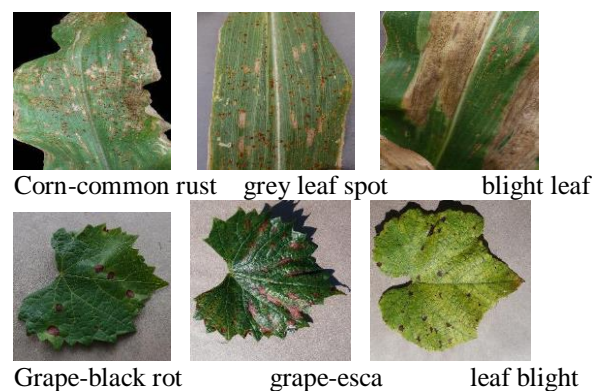


Fig. 7

### III. PROPOSED WORK & ML USE

PlantIP with the approach of image processing Technology and connecting to a network of agriculture experts' know-how or help fellow farmers with your knowledge and experience. Where they can talk about new methodologies and get good yields with our App [3].

Parts of the distributed run-time system for the plant disease detector are arranged to execute on user-side mobile devices and cloud-side centralized servers.

The system's deep learning model and the mobile device's Intermediate Representation model are both described in Layer 1.

The user interface is depicted in Layer 2 and created as an Android app to make it easy for system users (shown in Layer 3) to engage with the system.

In the first step of the system we propose, image acquisition, we take pictures with a digital camera.

Second, to enhance the leaf's damaged region and remove noise from the provided image, image processing techniques like image enhancement and image segmentation are applied to the leaf.

Finally, Teachable Machine's convolution neural network is used to complete the classification of the sick leaves. The processes below are repeatedly carried out for each image in the data collection.

The dataset's images are first imported into memory, then fed to the CNN model with the proper input sizes. After the CNN model computes and processes the image's information, it then outputs the image's classification.

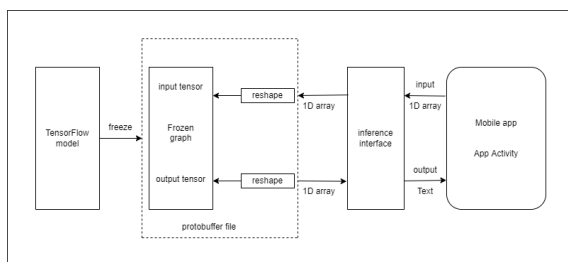


Fig. 8

The above figure represents the architecture of the deployment of the trained CNN model to the Android app using TensorFlow Lite.

After creating the CNN model, we first store it in the.h5 format, which freezes or saves the trained weights along with the model. Using the TocoConverter tool, we can transform our TensorFlow model into a TensorFlow Lite model, which can then be included in our Android app. This model can complete tasks without an internet connection thanks to its mobile device optimization.

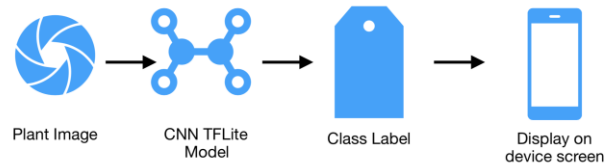


Fig. 9

First, the user launches the app, selects a plant image, and then clicks a button. The image is sent to the TensorFlow Lite model included with the app after pressing the button, which produces a class label. The screen of the device merely shows this class label.

### IV. ARCHITECTURE

- **STEP 1: Android Application Setup:**  
The application requires the user to provide correct credentials for the user to register or log in and to provide the correct permissions for the application. The Android application uses Tensor Flow, which is integrated, to leverage trained models.
- **STEP 2: Making a learnt model from a dataset:**  
Using the correct Tensor Flow methods, such as clustering, a model is made that has already been pre-trained from a collection of plants.

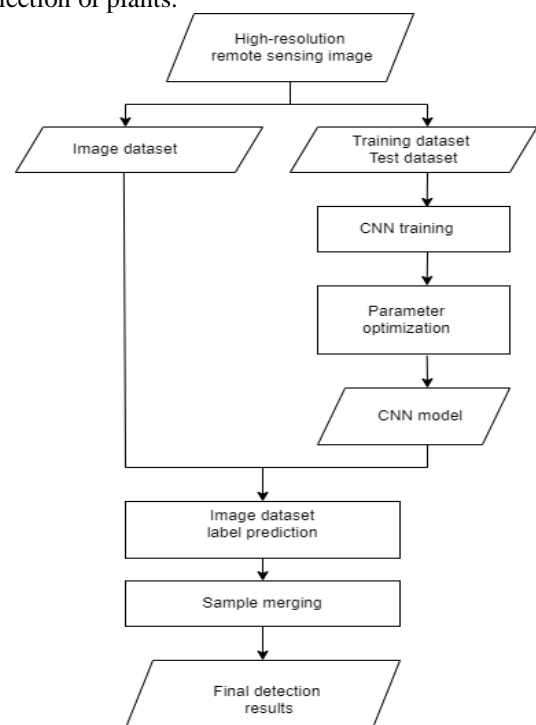


Fig. 10: Flowchart

- **STEP 3: Datasets for training and testing:**  
The TensorFlow comprises information about the plant disease and the model developed using this data. The ratio of training to testing sets is 10:1, correspondingly. The TensorFlow framework uses the prewritten clustering algorithm for machine learning. [4].

- **STEP 4: Testing data:**  
The "Identify the illness" option is available in the application. This further enables the algorithm to forecast illness on the fresh plant sample biased on the information supplied by farmers about the plants, if needed.

- **STEP 5: Optimization techniques:**  
The model is improved using TensorFlow's optimization methods, such as model [5]. enhancing, cutting, and stemming. The user chooses the display settings from the regional language or as selected by the user at login.

- **Neural Network**

A Neural Network is a scientific paradigm that is inspired by the ways in which organic nervous systems, such as the brain, process data and pictures into an easily processed format while retaining the qualities that are essential for getting superior outcomes.

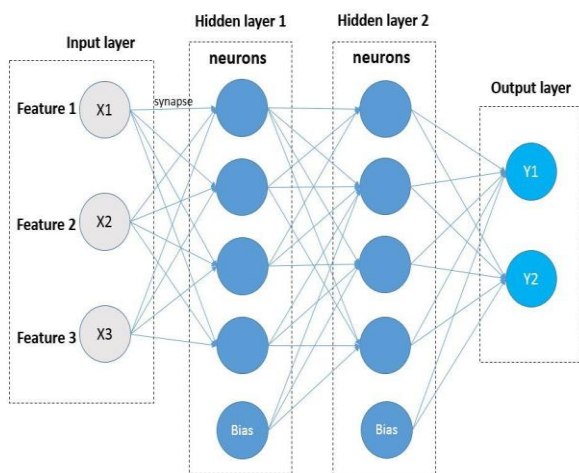


Fig. 11: Neural Network

**Input Layers:** In this layer, the model receives input. The entire number of characteristics in our data is equal to the number of neurons in this layer (number of pixels in case of an image).

- **Hidden Layer:** The concealed layer is then updated with the information from the. Depending on our model and the amount of input, hidden layers change. The number of neurons in each hidden layer might vary, although they are often more than the number of features. The output of each layer is derived by matrix multiplying the output of the layer before it by the learnable weights of that layer, and then by adding the learnable biases by the activation function, which makes the network nonlinear.

- **Output Layer:** The output of each class is then converted into the probability score for each class using a logistic function, such as sigmoid or SoftMax, using the data from the hidden layer as input.

- Draw the model Metrix graph

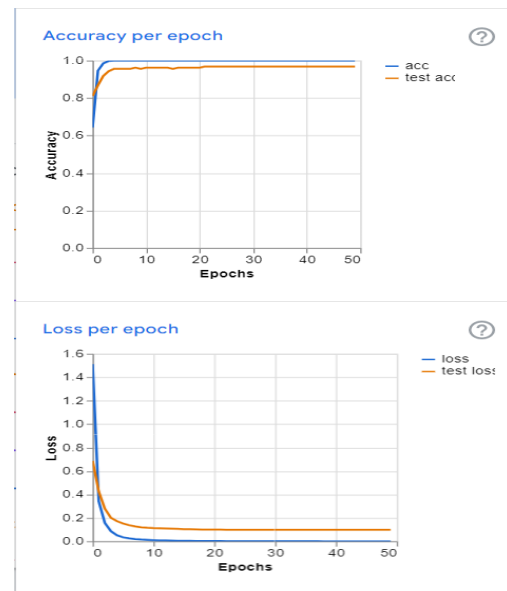


Fig. 12: Generated image from TensorFlow

"Loss" is the value of the loss over the training data at the end of each epoch. With training, the optimization process seeks to decrease this, therefore the lower the number, the better.

The proportion of accurate predictions to all other predictions in the training data is referred to as accuracy. Better if it's higher. Normally, but not always, the loss is negatively connected with this.

Epoch	Train loss	Valid loss	Time
0	282.509796	284.545929	06:33
1	219.822098	216.609177	06:11
2	184.608017	179.594299	06:16
3	157.201462	152.107513	06:13
4	146.833130	143.230316	06:08
5	142.532150	140.502411	06:06
6	130.854355	128.107193	06:13
7	123.135384	120.282646	06:16
8	122.825447	121.192963	06:15
9	113.097366	110.876205	06:09
10	110.630867	107.839455	06:05
11	105.667632	102.160103	06:08
12	104.367531	101.760544	06:11
13	96.9w82460	92.826294	06:10
14	94.381241	90.038216	06:11
15	91.442261	87.211021	06:12
16	89.456108	84.471718	06:19
17	88.846085	85.127541	06:02
18	85.060516	80.282585	06:05
19	83.723434	80.030220	06:20

Table 1



## V. CONCLUSION

By this Methodology, we can assist with managing all sorts of identification of problems such as assembly line product identification, fruits, or vegetables disease identification. It reduces the number of plants dying due to improper identification of disease.

## REFERENCES

- [1.] Savita N. Ghaiwat, Parul Arora, “Detection and Classification of Plant Leaf Diseases Using Image processing Techniques: A Review” 3, 2014.
- [2.] Renuka Rajendra Kajale, “Detection and Reorganization of Plant Leaf Disease Using Image Processing and Android O.S.” March-April 2015.
- [3.] A.S.Deokar, Akshay Pophale, Swapnil Patil, Prajakta Nazarkar, Sukanya Mungase, Harshad Shetye, Tejas Rane, Tanmay Pawar, Prof. Anuradha Dandwate“ An Analysis of Methodologies For Leaf Disease Detection Techniques ” February 2016.
- [4.] V Pooja; Rahul Das ; V Kanchana, “Identification of plant leaf diseases using image processing techniques”, 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), April 2018
- [5.] Boikobo Tlhobogang ; Muhammad Wannous, “Design of plant disease detection system: A transfer learning approach work in progress”, IEEE International Conference on Applied System Invention (ICASI), 2018,