Potato Leaf disease Image classification using Deep Learning

A MINI-PROJECT REPORT

By

•

Shreya Bhujbal
Juhi Checker
Anne Rajan

Under the guidance of
Prof. Dipali Koshti

DEPARTMENT OF COMPUTER ENGINEERING
FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING
FR. AGNEL ASHRAM, BANDRA (W),
MUMBAI - 400 050.

UNIVERSITY OF MUMBAI
(2018 – 2019)

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING
FR. AGNEL ASHRAM, BANDRA (W),
MUMBAI - 400 050.

CERTIFICATE

This is to certify that the following students working on the project "Potato and Strawberry leaf disease detection" have satisfactorily completed the requirements of the project in fulfillment of the course T.E in Computer Engineering of the University of Mumbai during academic year 2018-2019 under the guidance of "Prof. Dipali Koshti".

Submitted By: **Shreya Bhujbal (7919)**
Juhi Checker (7920)
Anne Rajan (8160)

Prof. Dipali Koshti
Guide

Dr. Sunil Surve
Head of the Department

_____
Principal

CERTIFICATE

This is to certify that the project synopsis entitled **"Potato Leaf disease Image classification using Deep Learning "** submitted by the following students is found to be satisfactory and the report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed for the course.

Shreya Bhujbal
Juhi Checker
Anne Rajan

Internal Examiner	**External Examiner**
(Signature)	(Signature)
Name:	Name:
Date:	Date:

Seal of the Institute

DECLARATION OF THE STUDENT

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea / data / fact / source in my submission.

We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the student **Date:**

(Shreya Bhujbal)
(Roll no: 7919)

Signature of the student **Date:**

(Juhi Checker)
(Roll no: 7920)

Signature of the student **Date:**

(Anne Rajan)
(Roll no: 8160)

ACKNOWLEDGEMENT

We are proud to present our report on "**Potato Leaf disease Image classification using Deep Learning**". Before elaborating on the technical details of the project, we would like to take this opportunity to express our gratitude towards our guide, Prof. Dipali Koshti, C.R.C.E., Bandra (W), Mumbai, for providing the required technical guidelines and suggestions regarding our Mini Project. She has helped us solve all our difficulties throughout the course of the project and has helped us become erudite in the subject.

We would also like to thank Prof. Mahesh Sharma, Dr Sunil Surve (Head of Computer Department), Dr.Srija Unnikrishnan (Principal) and the management of C.R.C.E., Mumbai for their cooperation and support, and for providing the necessary infrastructure for pursuing and implementing this project.

Shreya Bhujbal(Roll no.7919)

Juhi Checker (Roll no.7920)

Anne Rajan (Roll no.8160)

Date:

ABSTRACT

Crop cultivation plays an essential role in the agricultural field. So, if plant leaf are affected by the diseases, it may affect the production as well as the economy of the country. To identify the plant leaf diseases at an ultimate phase is not yet explored. In order to find out which disease affect the leaf, the farmer need to contact the expert for the detection of disease. The expert provides the suggestions which is based on its knowledge and information whereas sometimes searching the expert suggestion is time consuming, expensive and may be not precise. Therefore, to resolve this problem, Image processing techniques can be used which provides the accurate and fast solution. In this project we have made a mobile application which uses MobileNet and Inception Model, enhanced convolutional neural network algorithms to predict the infected area of the leaves. A color based segmentation model is defined to segment the infected region and placing it to its relevant classes. Our project is used to detect the leaf diseases of Potato and strawberry.

Introduction

India is an agricultural country and about 70% of the population depends on agriculture. Farmers have large range of diversity for selecting various suitable crops and finding the suitable pesticides for plant. Diseases on plant leads to the significant reduction in both the quality and quantity of agricultural products. Monitoring of health and disease on plant plays an important role in successful cultivation of crops and plant growth in the farm.

The image processing techniques can be used in the plant disease detection. A neural network learns how to extract features while training. CNN being a multi-layer feed-forward neural network, is the popular deep learning model. Image recognition of plant diseases is to extract the characteristic feature information from the diseased regions in the obtained images by using image processing techniques, and then to achieve disease recognition by using pattern recognition methods. Generally, the extracted features from the images of plant diseases include color features, shape features, texture features, and so on. It is very important to extract the effective characteristic features for the image recognition of plant diseases.

Problem Statement

The problem statement is ‘To analyse the leaves of potato and strawberry plant and identify if the plant has a disease and diagnose using deep learning methodologies’. The application to be proposed must be able to process and analyse images of leaves of potato and strawberry plant and detect and diagnose the diseases affecting the plant. The application would be used in a field setup and it has help and aid farmers and cultivators to identify diseases and seek remedies for it.

Project Scope

1. To help farmers identify disease of crop by simply uploading leaf image.
2. To get remedy / solution for a problem easily.
3. To help farmers to get their queries / problems faced solved by experts
4. To facilitate farmers community to come together and help each other.
5. To make farmers aware of schemes and benefits announced by government.
6. To break the language and technology barrier between farmer and solution.

Table of Contents

1. Introduction

In early days, the monitoring and analysis of plant diseases were done manually by the expertise person in that field. This requires tremendous amount of work and also requires excessive processing time. Also identifying plant disease incorrectly leads to huge loss of yield, time, money and quality of product.

Plant disease diagnosis through optical observation of the symptoms on plant leaves, incorporates a significantly high degree of complexity. Due to this complexity and to the large number of cultivated plants and their existing phytopathological problems, even experienced agronomists and plant pathologists often fail to successfully diagnose specific diseases, and are consequently led to mistaken conclusions and treatments.

To overcome this, we have proposed an mobile application '**Farmitra**' through which we can detect the plant leaf diseases and offer a corresponding treatment measures as well as the farmers can post a comment or issues in the forum. The Plant leaf disease detection is done by using image processing . Image recognition of plant diseases is to extract the characteristic feature information from the diseased regions in the obtained images by using image processing techniques, and then to achieve disease recognition by using pattern recognition methods such as neural networks.

Deep learning can be thought as a learning method on neural networks. Due to image recognition, the Farmitra app is able to identify the plant type - as well as the appearance of a possible disease. The Deep Learning Model we have proposed is the MobileNet and Inception Model, which are widely used Image Recognition models and with a greater accuracy rate of around 99.7% as compared with the other Deep Learning Models such as Vgg(95%) and Resnet(97%).

React native, an upcoming cross platform app development which provides optimal performance, simple user interface and support third party plugins makes a very efficient front-end. In our product we have built our front-end on react native so that the farmers find it easy to use.

2. Review of Literature

In the related works we came across, several Neural Networks were used for plant disease prediction like the basic Convolutional Neural Network(CNN) and other Artificial Neural Networks(ANN) like VGG, Alexnet and Resnet. In order to understand their accuracy levels and the losses incurred, these models were trained against our dataset. Numerous approaches have been proposed over the years.

In traditional systems approach for detection and differentiation of plant diseases can be achieved using Support Vector Machine algorithms. But the problem with SVM is that its not suitable for training large datasets as the complexity is dependent on the size of the dataset. Other drawbacks are that the training time needed is high and the accuracy is less. Another approach based on leaf images and using ANNs as a technique for an automatic detection and classification of plant diseases was used with K-means as a clustering procedure but that gives an accuracy of somewhere between 65% to 90%. As mentioned above, we tried the following models and these were our observations.

CNN

Convolutional Neural Networks are regularized versions of multilayer perceptrons. The "fully-connectedness" of these networks (each neuron in one layer is connected to all neurons in the next layer), make them prone to overfitting data. Typical ways of regularization includes adding some form of magnitude measurement of weights to the loss function. However, CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme. In this model, a total of 7 layers are involved. CNN is better for large datasets unlike SVM which is used for small data sets with fewer outliers.Another advantage is that we had to worry less about the feature engineering part.

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 224, 224, 32)	896
activation_7 (Activation)	(None, 224, 224, 32)	0
batch_normalization_v1_6 (Ba	(None, 224, 224, 32)	128
max_pooling2d_5 (MaxPooling2	(None, 74, 74, 32)	0
dropout_5 (Dropout)	(None, 74, 74, 32)	0
conv2d_6 (Conv2D)	(None, 74, 74, 64)	18496
activation_8 (Activation)	(None, 74, 74, 64)	0

batch_normalization_v1_7 (Ba (None, 74, 74, 64) 256

max_pooling2d_6 (MaxPooling2 (None, 24, 24, 64) 0

dropout_6 (Dropout) (None, 24, 24, 64) 0

conv2d_7 (Conv2D) (None, 24, 24, 64) 36928

activation_9 (Activation) (None, 24, 24, 64) 0

batch_normalization_v1_8 (Ba (None, 24, 24, 64) 256

max_pooling2d_7 (MaxPooling2 (None, 12, 12, 64) 0

dropout_7 (Dropout) (None, 12, 12, 64) 0

conv2d_8 (Conv2D) (None, 12, 12, 128) 73856

activation_10 (Activation) (None, 12, 12, 128) 0

batch_normalization_v1_9 (Ba (None, 12, 12, 128) 512

max_pooling2d_8 (MaxPooling2 (None, 6, 6, 128) 0

conv2d_9 (Conv2D) (None, 6, 6, 128) 147584

activation_11 (Activation) (None, 6, 6, 128) 0

batch_normalization_v1_10 (B (None, 6, 6, 128) 512

max_pooling2d_9 (MaxPooling2 (None, 3, 3, 128) 0

dropout_8 (Dropout) (None, 3, 3, 128) 0

flatten_1 (Flatten) (None, 1152) 0

dense_2 (Dense) (None, 1024) 1180672

activation_12 (Activation) (None, 1024) 0

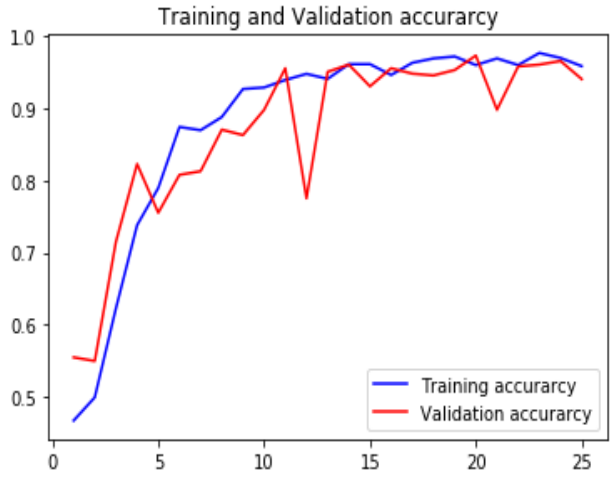
batch_normalization_v1_11 (B (None, 1024) 4096

dropout_9 (Dropout) (None, 1024) 0

dense_3 (Dense) (None, 2) 2050

activation_13 (Activation) (None, 2) 0

Total params: 1,466,242
Trainable params: 1,463,362
Non-trainable params: 2,880



- After testing on 2743 training datasets, we have achieved 97.94% accuracy
- After testing on 680 validation datasets, we have achieved 96.50% accuracy
- Test Accuracy: 96.62%

VGG

VGG refers to a deep convolutional network for object recognition developed and trained by Oxford's renowned Visual Geometry Group (**VGG**). VGG is one of the models we used to predict a particular plant disease. The model is pre-trained model on keras called VGG16. Keras provides both the 16-layer and 19-layer version via the VGG16 and VGG19 classes. For this model, we trained and tested the model on our dataset. The summary for the model is as following

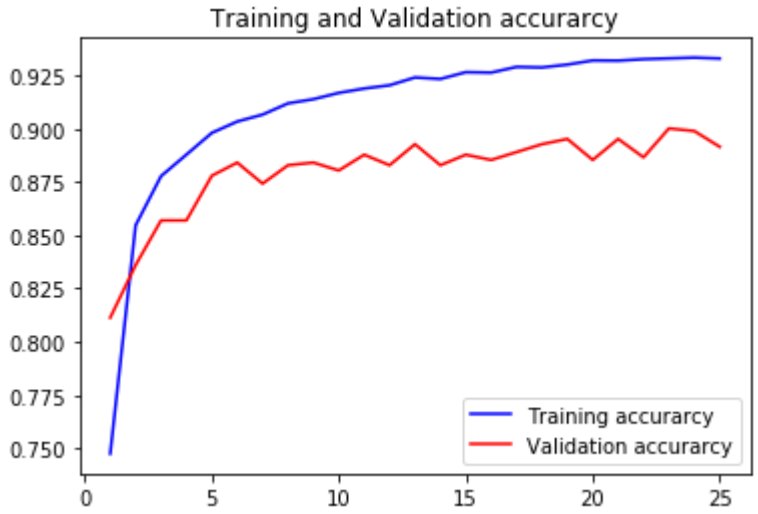
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 64, 64, 3)	0
block1_conv1 (Conv2D)	(None, 64, 64, 64)	1792
block1_conv2 (Conv2D)	(None, 64, 64, 64)	36928

block1_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block2_conv1 (Conv2D)	(None, 32, 32, 128)	73856
block2_conv2 (Conv2D)	(None, 32, 32, 128)	147584
block2_pool (MaxPooling2D)	(None, 16, 16, 128)	0
block3_conv1 (Conv2D)	(None, 16, 16, 256)	295168
block3_conv2 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv3 (Conv2D)	(None, 16, 16, 256)	590080
block3_pool (MaxPooling2D)	(None, 8, 8, 256)	0
block4_conv1 (Conv2D)	(None, 8, 8, 512)	1180160
block4_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block4_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block4_pool (MaxPooling2D)	(None, 4, 4, 512)	0
block5_conv1 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv2 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv3 (Conv2D)	(None, 4, 4, 512)	2359808
block5_pool (MaxPooling2D)	(None, 2, 2, 512)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 0)	0

==========

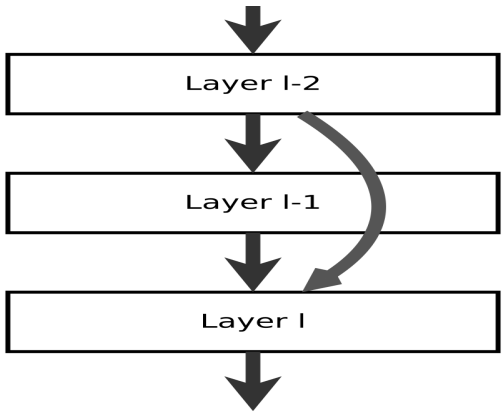
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688

Time taken to train and validate 3251 and 811 images with 25 epochs and steps per epoch were 300(usually taken equal to number of images in training set but the training time for that was predicted to be 8 hours) was 3 hours and 4 mins .
The training accuracy received was 93.27% and validation accuracy was 89.15%. Also while testing, the model was unable to differentiate between Esca and Black Rot due to their similar patterns. Hence we decided to not go further with model. But if we train VGG model for two to three weeks the prediction rate will be commendable.

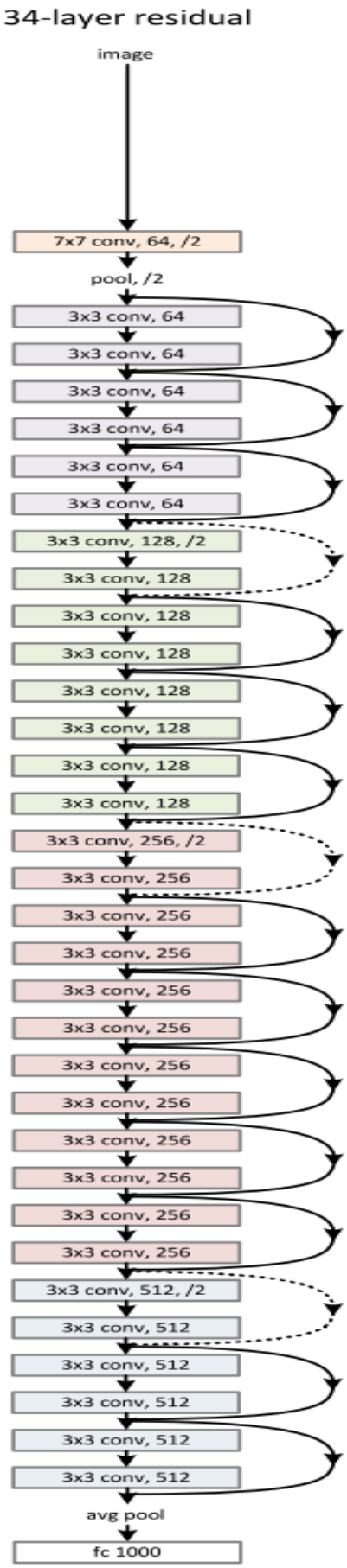


ResNet

A **residual neural network** (**ResNet**) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing skip connections, or short-cuts to jump over some layers. Typical ResNet models are implemented with single-layer skips.

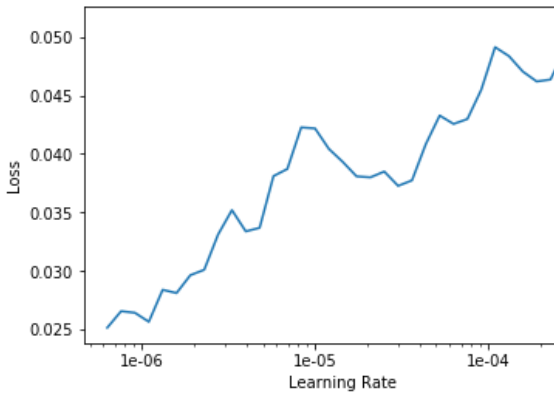


There are several variants of the ResNet model. The one we used has 34 blocks of layers, with each block having 2-4 layers.

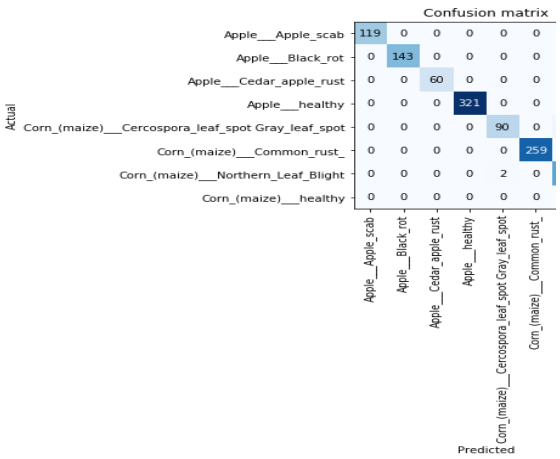


The total time taken to train and validate 7032 images for 10 epochs using GPU was 16 mins 43 secs.

An accuracy of **98.8%** was achieved against 10 folds of testing and validation. We normalized the images and the model was also able to predict accurately for uncropped images, i.e. without secluding one particular leaf from the others.



The validation loss incurred can be seen as above.



ResNet had turned out to be a very good and efficient model, but, as higher accuracy was achieved using InceptionNet, we went along with that.

3. Report on the proposed system

3.1. Problem Statement Analysis

Problem Statement:

The problem statement is 'To analyse the leaves of potato and strawberry plant and identify if the plant has a disease and diagnose using deep learning methodologies'. The application to be proposed must be able to process and analyse images of leaves of potato and strawberry plant and detect and diagnose the diseases affecting the plant. The application would be used in a field setup and it has help and aid farmers and cultivators to identify diseases and seek remedies for it.

Analysis:

The problem statement stated above specifies the need and urgency of a technological system for analyzing and monitoring the crop health and thus providing a helping hand to the farmers community to solve issues that they face in their daily life. In India agricultural development is one of the most important topic and needs to be taken care of. With statistics revealing that 45 farmers committing suicides in India every day, there is a need of crucial steps to be taken in this sector. A high percentage of farmers complained of repeated losses; 70 percent of respondents said their crops were destroyed because of unseasonal rains, drought, floods and pest attack. Although there are specialized help centres like Kisan call centres available according to a survey 70 percent farmers never contacted these centres. The farmers face with the problem of curbing the disease affecting the crops in the crucial stages. Sometimes lack of knowledge lead to ignorance and the neighbouring cultivation are also affected which further lead to unaffordable losses. Today Technology is proliferating into all aspects of business, and agriculture is no exception. Farms are becoming increasingly automated and data-driven. Wireless and GPS technology now helps farmers monitor their crops and livestock; it also aggregates and analyzes data to help them make better decisions. There is a need for a system which would automatically detect the disease that the crop is suffering with and provide solution and steps to curb the disease. As the farmers are not equipped with high end technology but are limited to resources like a mobile phone the designed system should still be able to address the issue with low end devices. The system must also pose an alternate solution in case of any connectivity issues that the farmers may face due to less availability of Internet and other such resources. The proposed system should also be capable of serving in different dialects as the farmers may not be accustomed to the English language alone.

3.2. Design and Methodology of proposed system

We have developed an mobile application named 'Farmitra' that will help farmers of India to detect plant leaf diseases. This will be done using a convolutional neural network model in the backend and React Native in the frontend. In this application, farmer would have to click a picture on his/her phone and the CNN algorithm will detect which plant leaf disease it is along with it's remedies. Our product also

provides a portal for farmers to post their problems related to farming and experts around the world would answer them.

Use Case Diagram:

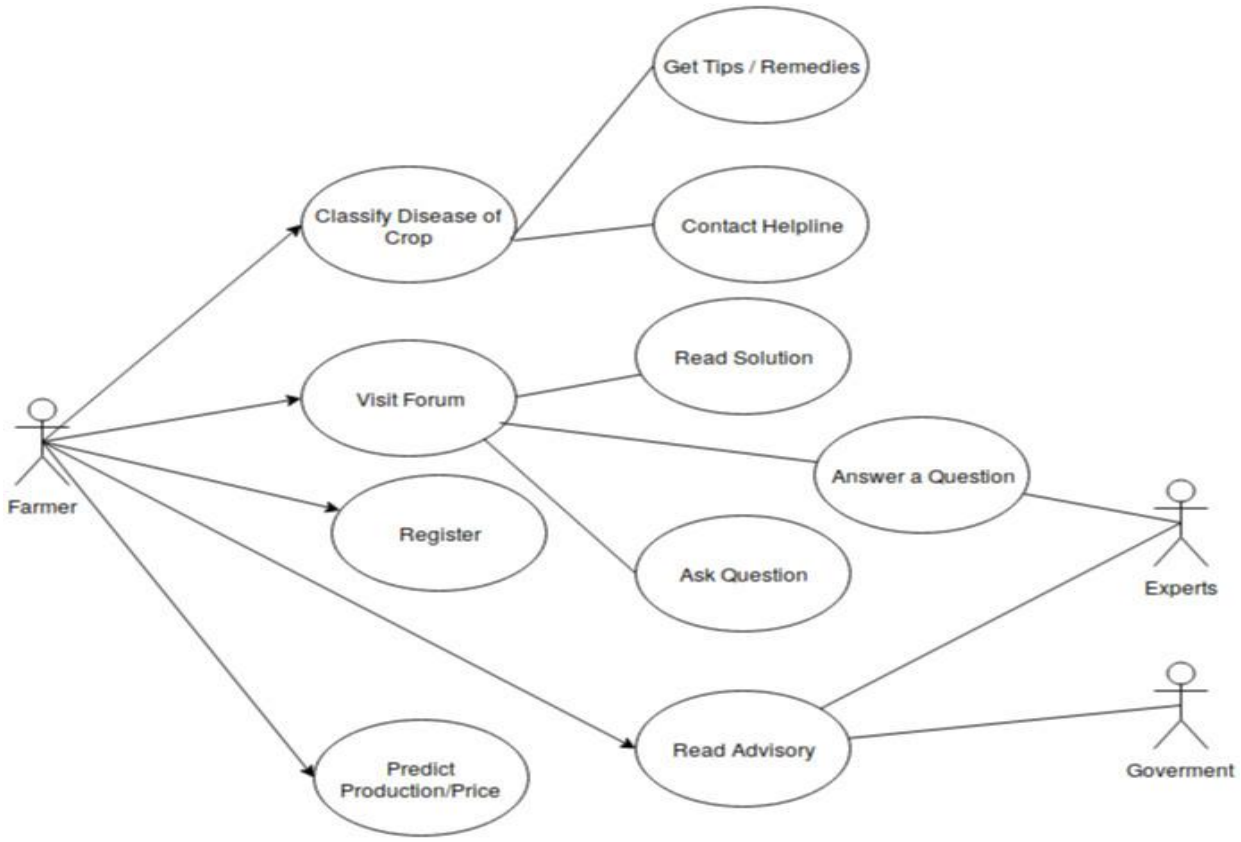
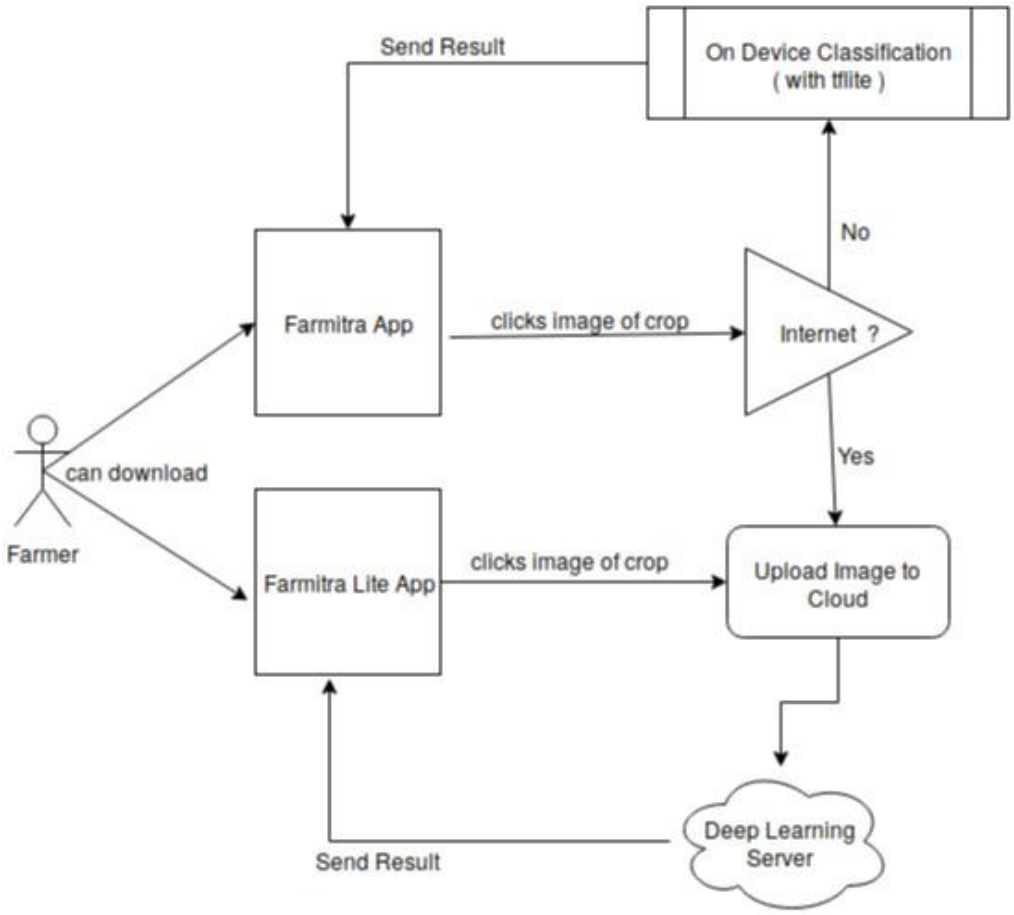


Figure 3.2.1. Block Diagram of Proposed system

Workflow of classification:



4. Hardware software requirements and Implementation

Hardware Requirements:

Client Side

OS	**Version**
Android	Android 6+ (Marshmallow and above)
iOS	ios 9+

Mobile Requirements:

Feature	**Minimum Requirement**
Camera	5 megapixel
Internet Connectivity	2G (GPRS)
RAM	512mb(Lite Version), 1GB (normal version)
Free Memory	50mb(Lite Version), 160mb(normal version)

Server Side

Server	Windows / Linux
Database	Postgresql / MySql
RAM	2GB
MEMORY	5GB

Software Requirements:

Client Side:

Only the Krishi Mitra or Krishi Mitra Lite App installed on the mobile device.

Server Side:

Software	**Version**
Nodejs	10.x.x +
Postgresql	9.x.x +
Python	3.5.3
Tensorflow	1.13.1
Flask	1.0.2
Imageai	2.0.2
opencv-python	4.10.1
h5py	2.9.0

Implementation:

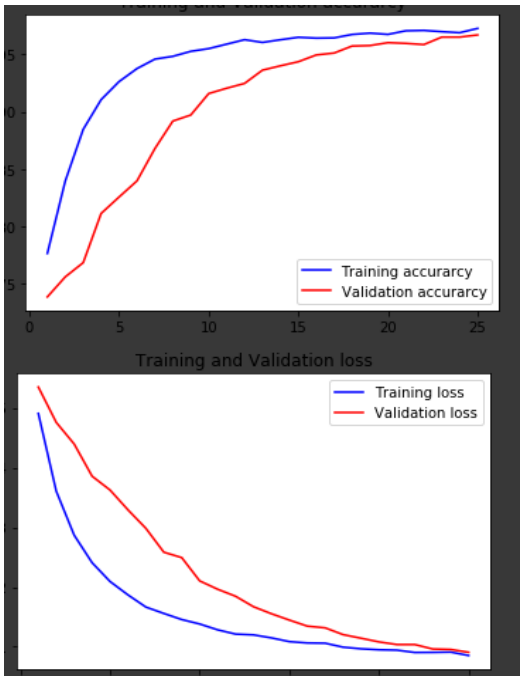
Backend:

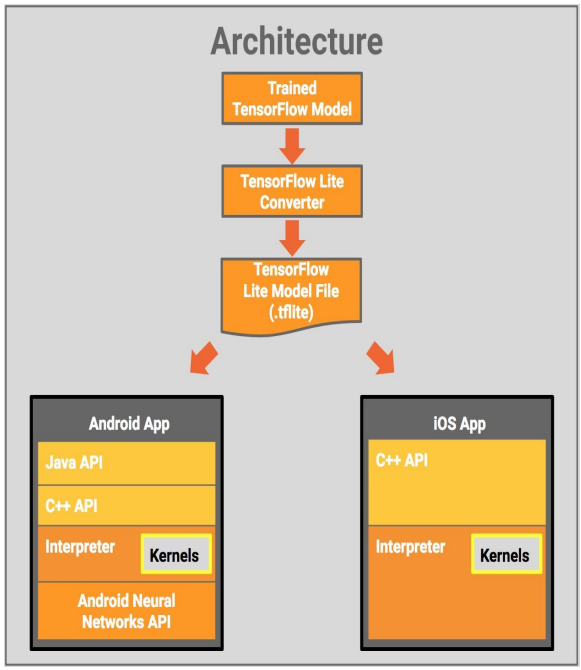
MobileNet V2

- We have created the base model from the **MobileNet V2** model developed at Google, and pre-trained on the ImageNet dataset, a large dataset of 1.4M images and 1000 classes of web images. This is a powerful model.
- Due to lack of computation power we decided to create 4 different models of tflite for disease prediction. Each model consists of 2 crops. Since MobileNet is specifically designed to be deployed inside mobile, we chose to use MobileNet model with Transfer Learning. One Average Pooling layer and one Dense layer is added after MobileNet model to predict our classes. Each model is trained roughly with 5000 images which gives accuracy of 97% +-2.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
mobilenetv2_1.00_224 (Model) (None, 7, 7, 1280)        2257984
_________________________________________________________________
global_average_pooling2d (Gl (None, 1280)              0
_________________________________________________________________
dense (Dense)                (None, 5)                 6405
=================================================================
Total params: 2,264,389
Trainable params: 6,405
Non-trainable params: 2,257,984
_________________________________________________________________
```

- As seen from graph,training accuracy gradually increases with the validation accuracy and similarly training loss decreases with same rate as validation loss.Since there are no spikes in the graph we can conclude that model is learning perfectly with gradual rate.





- We convert the model to a **TensorFLow Lite** format, and finally, integrate the model in our app.TensorFlow Lite is TensorFlow's lightweight solution for mobile and embedded devices.
- Tflite is currently in beta, and hence is not supported for all Deep Learning models. After we create a deep learning, we need to save it in h5 file format. Tflite converter then helps to convert h5 file to tflite file which is deployable inside mobile. While deploying tflite we need to make sure we don't compress the file and save it as it is, else there might be data loss and accuracy will be affected during prediction.

RENDER

We have hosted our app on Render.It is a cloud provider that instantly deploys our code directly from GitHub.

FLASK

Our web application uses Flask web framework.
Flask is a micro web framework written in Python.

GUNICORN

We have used gunicorn as our server.Gunicorn 'Green Unicorn' is a Python WSGI HTTP Server.

Inception v-3 :

We implemented, transfer learning, a machine learning method which utilizes a pre-trained neural network. The image recognition model is one of them called Inception-v3 consisting of two parts:

- Feature extraction part with a convolutional neural network.
- Classification part with fully-connected and softmax layers.

The pre-trained Inception-v3 model achieves state-of-the-art accuracy for recognizing general objects with 1000 classes, like "Zebra", "Dalmatian", and "Dishwasher". The model extracts general features from input images in the first part and classifies them based on those features in the second part.

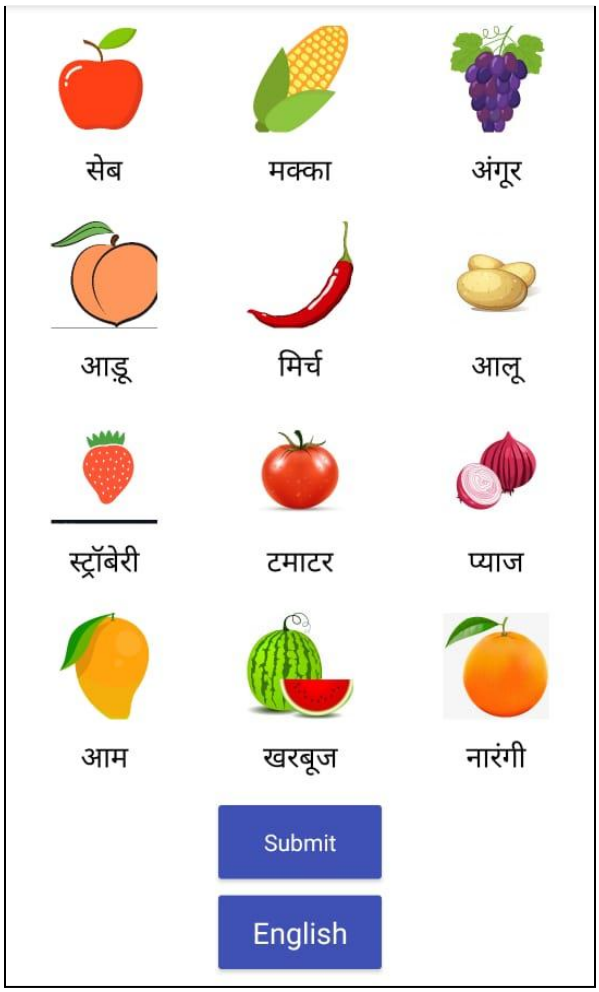
In transfer learning, when you build a new model to classify your original dataset, you reuse the feature extraction part and re-train the classification part with your dataset. Since we don't have to train the feature extraction part (which is the most complex part of the model), we could train the model with less computational resources and training time.

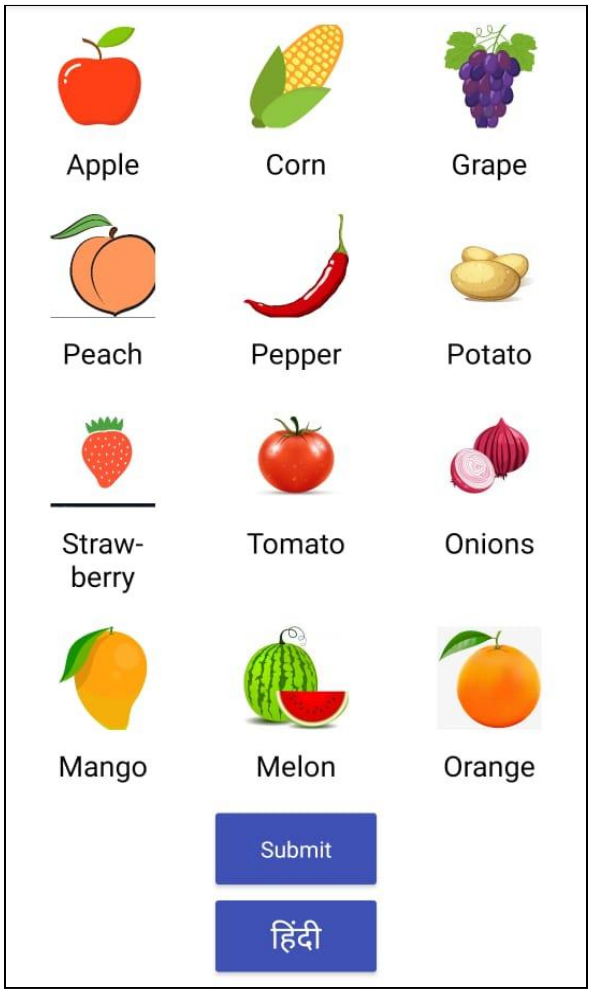
We have used ImageAI, a python library to build applications and systems with self-contained Deep Learning and Computer Vision capabilities using simple and few lines of code. We were able to classify our images using ImageAI's inception v-3 model which gave us the accuracy of 99.7%.

```
__________________________________________________________________________________________________
concatenate_5 (Concatenate)     (None, 5, 5, 768)    0           activation_279[0][0]
                                                                 activation_280[0][0]
__________________________________________________________________________________________________
activation_281 (Activation)     (None, 5, 5, 192)    0           batch_normalization_v1_281[0][0]
__________________________________________________________________________________________________
mixed10 (Concatenate)           (None, 5, 5, 2048)   0           activation_273[0][0]
                                                                 mixed9_1[0][0]
                                                                 concatenate_5[0][0]
                                                                 activation_281[0][0]
__________________________________________________________________________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)         0           mixed10[0][0]
__________________________________________________________________________________________________
predictions (Dense)             (None, 31)           63519       avg_pool[0][0]
==================================================================================================
Total params: 21,866,303
Trainable params: 21,831,871
Non-trainable params: 34,432
```

Frontend:

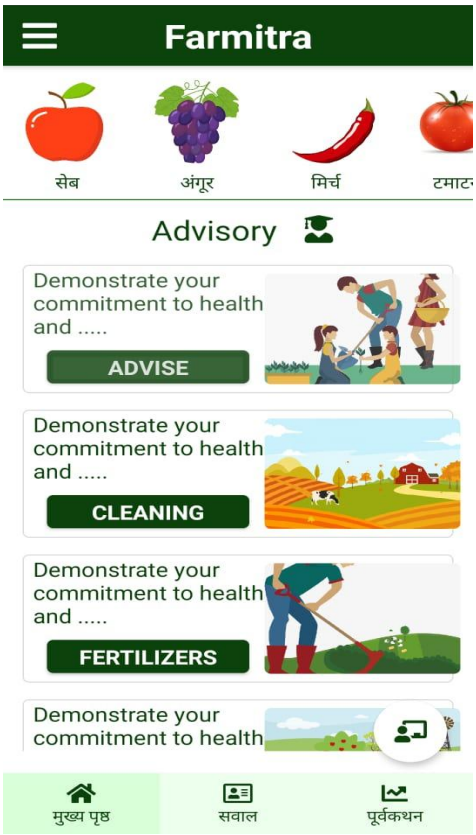
Crop Selection Page:





First page of our app will be crop selection page, here farmer can select on the crops which he has in his farm for clutterfree experience. Moreover, our app is available in two languages, English and Hindi, farmer can select any one if the choice here.

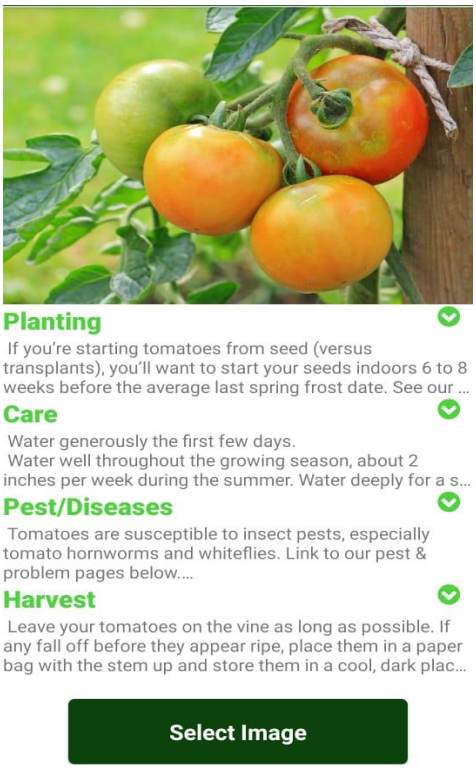
Home Page:



In Home Page, Farmer can easily see the crops list which he had selected. Below which is the Advisory Feed which will be created specially for farmer according to his location and types of crop selected.

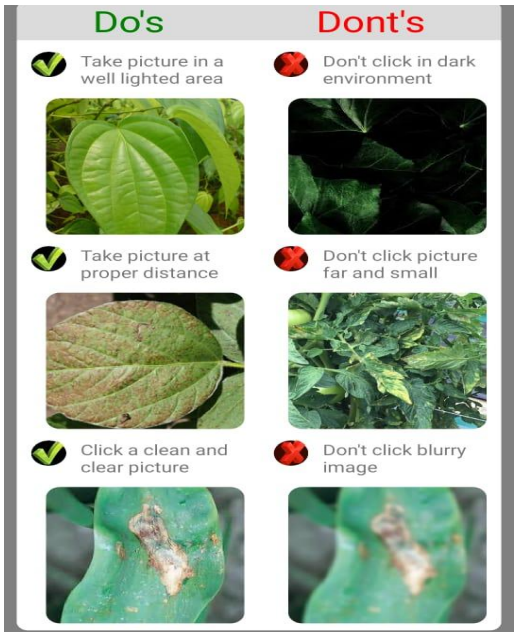
This content will be created by experts in field of agriculture, professors of institute and even government for announcing new schemes and benefits for farmers. With the help of this portal, we will bridge the gap between farmers and information meant for him.

Crop Detail Page:



In Crop Detail Page, extra information about the crop will be displayed. Like Planting instructions, Care, Pest/Diseases and Harvest Instruction. This page will be really helpful if farmer is planting some crop for first time. If he has some issue with the crop, he can click image and check which disease the crop has been infected with.

Do's & Don't Prompt:

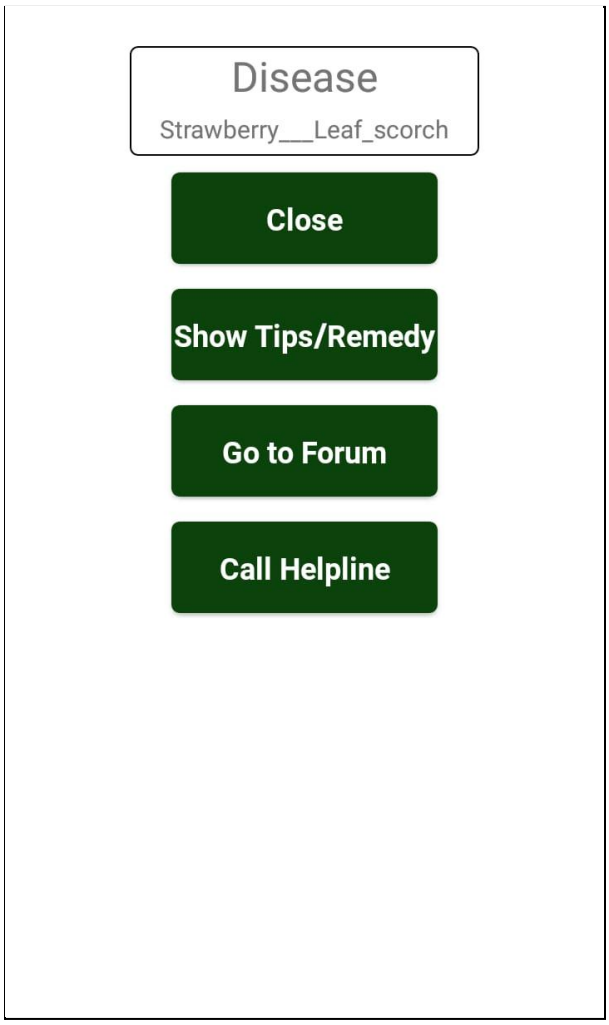


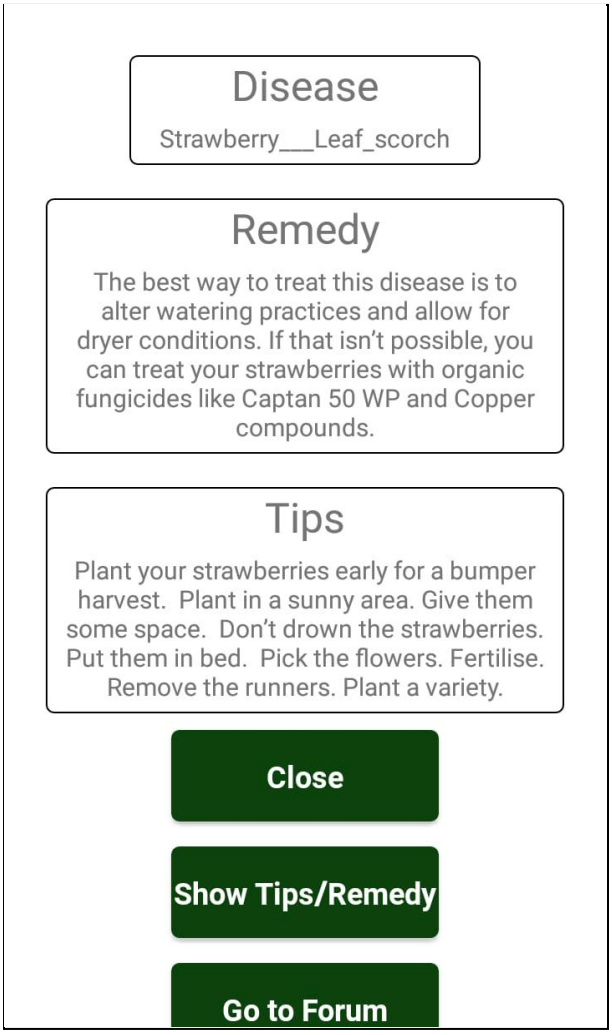
Before farmer clicks an image, he will be prompted with Do's and Don't screen, so that we clicks the correct picture with which our model can predict disease with high accuracy

Image Upload Page:

Here, farmer can select the leaf image for prediction.

Result Page:



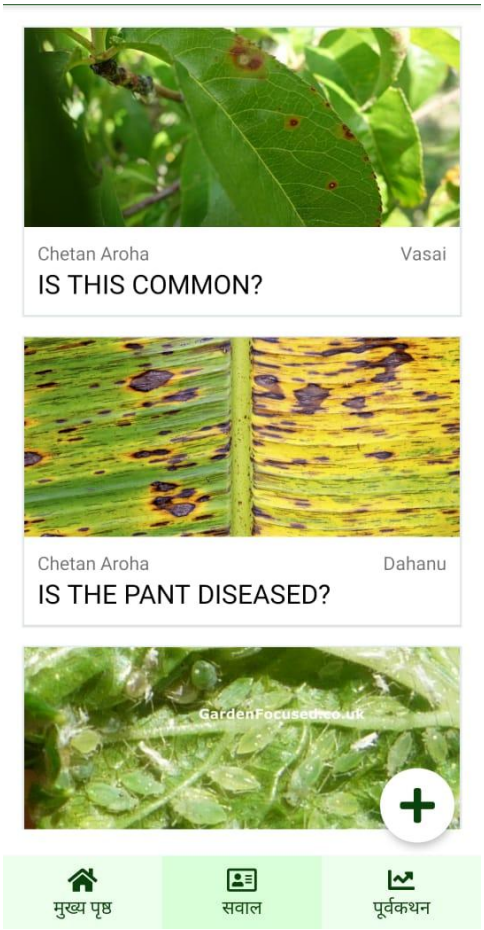


This page will show the result of prediction. Based upon the app (lite or regular) and internet connection. Prediction will be processed either from on-device deep learning model(offline version) or from the model deployed in cloud server.

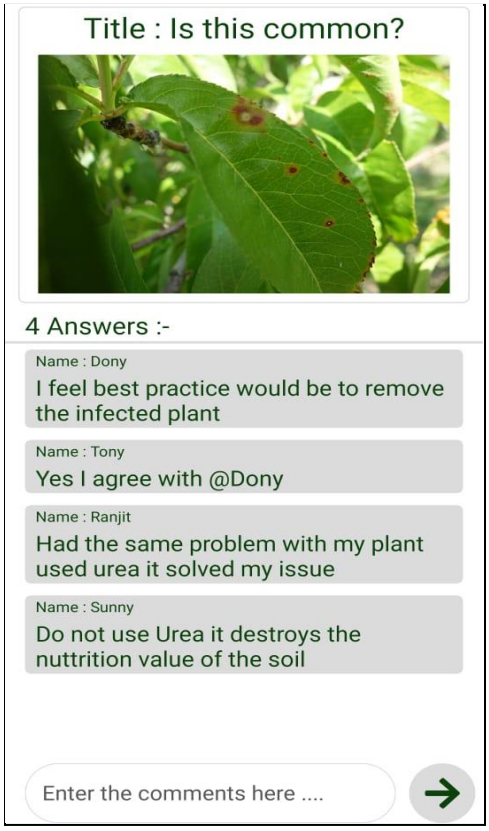
Based upon the disease predicted, the farmer will also displayed Tips and Remedy for the problem.

If still he don't find any help in that, they can find more solutions on our Forum or he can directly call the Farmer's Helpline from here.

Forum Page:



Question Form



Answer Page

Forum Page is common platform for all the farmers to share their problems and get the solution from others. Main motto of this platform is “A farmer can help a farmer the best” .

6 .References

[1]	https://www.ijareeie.com/upload/2016/march/25_Plant.pdf
[2]	https://acadpubl.eu/hub/2018-119-14/articles/2/109.pdf
[3]	https://www.researchgate.net/publication/315136264_Detection_of_Plant_Leaf_Disease_Employing_Image_Processing_and_Gaussian_Smoothing_Approach
[4]	http://www.ijsret.org/pdf/121761.pdf
[5]	H. Wang, G. Li, Z. Ma and X. Li, "Image recognition of plant diseases based on principal component analysis and neural networks," *2012 8th International Conference on Natural Computation*, Chongqing, 2012, pp. 246-251.doi: 10.1109/ICNC.2012.6234701
[6]	S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," *2015 International Conference on Computing Communication Control and Automation*, Pune, 2015, pp. 768-771.doi: 10.1109/ICCUBEA.2015.153
[7]	H. Sabrol and K. Satish, "Tomato plant disease classification in digital images using classification tree," *2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, 2016, pp. 1242-1246. doi: 10.1109/ICCSP.2016.7754351
[8]	Konstantinos P.Ferentinos, “Deep learning models for plant disease detection and diagnosis “,https://doi.org/10.1016/j.compag.2018.01.009.
[9]	Plantix . https://plantix.net/
[10]	https://plantvillage.psu.edu/
[11]	Vijai Singh a, A.K. Misra,” Detection of plant leaf diseases using image segmentation and soft computing techniques”, Science Direct. https://doi.org/10.1016/j.inpa.2016.10.005