Project Report on

“Crop Disease Prediction”

Submitted By

SNEHA KADAM	**(BE15F03F029)**
MANALI SHAH	**(BE15F03F051)**
PRACHI NAIK	**(BE15F06F039)**
VRUSHALI POTDAR	**(BE15F06F042)**

Guided By

Prof. Komal S. Gandle

In partial fulfillment for the award of

Bachelor of Engineering

(Information Technology)

“In Pursuit of Global Competitiveness”

Department of Information Technology

Government College of Engineering, Aurangabad

(2018-19)

CERTIFICATE

This is to certify that, the project report entitled "**Crop Disease Prediction**" which is being submitted here with for the award of the "**Bachelor of Engineering"** in "**Information Technology**" for the academic year 2018-19 Government College of Engineering, Aurangabad.

Sneha Kadam	(BE15F03F029)
Manali Shah	(BE15F03F051)
Prachi Naik	(BE15F06F039)
Vrushali Potdar	(BE15F06F042)

Place: Aurangabad

Date:

Prof. K. S. Gandle
(Guide)

Prof. Chitra Gaikwad
(Head of Department)

Dr. P.B. Murnal
(Principal)

Government College of Engineering,

Aurangabad

(2018-19)

DECLARATION

We hereby declare that the project work entitles "**Crop Disease Prediction**" submitted to the Government College of Engineering, Aurangabad, is a record of an original work done by us guidance of **Prof. K. S. Gandle**, our project guide.

This project work is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Information Technology Department

Sneha Kadam (BE15F03F029)

Manali Shah (BE15F03F051)

Prachi Naik (BE15F06F039)

Vrushali Potdar (BE15F06F042)

Place: Aurangabad

Date:

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Despite having seen many improvements in the mass production and accessibility of food, food security remains threatened by a variety of factors such as the decline of pollinators and plant diseases. In the developing world, more than 80 percent of the agricultural production is generated by smallholder farmers, and reports of yield loss of more than 50% due to pests and diseases are common. Furthermore, the majority of individuals suffering from hunger live in smallholder farming households. Fortunately, diseases can be managed by identifying the diseases as soon as it appears on the plant. In addition, with the rise of the internet and mobile technology worldwide, it is easy to access diagnosis information on a particular type of disease. As a result, the prevalence of smartphones with powerful cameras can help to scale up any type of solution that involves crop detection feasible and practical.

Smartphones in particular offer very novel approaches to help identify diseases because of their computing power, high-resolution displays, and extensive built-in sets of accessories, such as advanced HD cameras. In fact it is estimated that around 6 billion phones would be available around 2050.

1.2 Necessity

Crop diseases serve as a major threat to food supply. As a result of the growing of smartphone technology throughout the world, it has now become technical feasible to leverage image processing techniques to identity type of plant disease from a simple photo. Identifying disease can lead to quicker interventions that can be implemented to reduce the effects of crop diseases on food supply. Using a public dataset of images of diseased and healthy plants, a deep convolutional network and semi supervised methods are trained to classify crop species and disease status of 57 different classes.

1.2 Objectives

- Providing an efficient way for detection of crops.
- Providing immediate solution for the diseased crop.
- GUI of application will be as simple as possible for handling by end users.

CHAPTER 2

LITERATURE SURVEY

2.1 Image classification using Deep learning

M Manoj krishna , M Neelima , M Harshali , M Venu Gopala Rao

International Journal of Engineering & Technology, 7 (2.7) (2018)

The image classification is a classical problem of image processing, computer vision and machine learning fields. In this paper we study the image classification using deep learning. In this, AlexNet fields. In this paper we study the image classification using deep learning. In this, AlexNet architecture with convolutional neural networks is used. Four test images are selected from the ImageNet database for the classification purpose. We cropped the images for various portion areas and conducted experiments. The results show the effectiveness of deep learning based image classification using AlexNet.

Classification is a systematic arrangement in groups and categories based on its features. Image classification came into existence for decreasing the gap between the computer vision and human vision by training the computer with the data. The image classification is achieved by differentiating the image into the prescribed category based on the content of the vision. The conventional methods used for image classifying is part and piece of the field of artificial intelligence (AI) formally called as machine learning. The machine learning consists of feature extraction module that extracts the important features such as edges, textures etc and a classification module that classify based on the features extracted. The main limitation of machine learning is, while separating, it can only extract certain set of features on images and unable to extract differentiating features from the training set of data. This disadvantage is rectified by using the deep learning.

Deep learning (DL) is a sub field to the machine learning, capable of learning through its own method of computing. A deep learning model is introduced to persistently break down information with a homogeneous structure like how a human would make determinations. To accomplish this, deep learning utilizes a layered structure of several algorithms expressed as an artificial neural system (ANN). The architecture of an ANN is simulated with the help of the biological neural network of the human brain. This makes the deep learning most capable than the standard machine learning models . In deep learning, we consider the neural networks that identify the image based on its features. This is accomplished for the building of a complete feature extraction model which is capable of solving the difficulties faced due to the conventional methods. The extractor of the integrated model should be able to learn extracting the differentiating features from the training set of images accurately. Many methods like GIST, histogram of gradient oriented and Local Binary Patterns, SIFT are used to classify the feature descriptors from the image.

Artificial Neural Networks

A neural network is a combination of hardware bonded or separated by the software system which operates on the small part in the human brain called as neuron. A multi layered neural network can be proposed as an alternative of the above case. The training image samples should be more than nine times the number of parameters essential for tuning the classical classification under very good resolution. The multi-layered neural network is so complicated task with respect to its architecture in the real world implementations. The multi-layered neural network is at present expressed as the Deep Learning.

Neural networks are expressed in terms of number of layers involved for producing the inputs and outputs and the depth of the neural network. Neural networks are involved in many principles like fuzzy logic, genetic algorithms and Bavesian methods. These layers are generally referred to as hidden layers. They are expressed in terms of number of hidden nodes and number of inputs and outputs every node consists. The Convolutional Neural Network (ConvNet) is most popular algorithm used for implementing the deep learning technique. The ConvNet consists of Feature detection layers and classification. A ConvNet is composed of several layers, and they are convolutional layers, max pooling or average-pooling layers, and fully-connected layers.

Alexnet

The ConvNet is categorized into two types named LeNet and AlexNet. The LeNet is expressed as the Shallow Convolutional Neural Networks which is designed to classify the hand-written digits. The LeNet comprises of 2 convolutional layers, 2 subsampling layers, 2 hidden layers and 1 output layer . The AlexNet is expressed as the deep convolutional neural networks which are used for classifying the input image to one of the thousand classes.

AlexNet is used to solve many problems like indoor sense classification which is highly seen in artificial neural intelligence. It is a powerful method of knowing the features of the image with more differential vision in the computer field for the recognition of patterns. This paper discuss about the classification of a particular size of image of required choice. It can very effectively classify the training sample of images present in the AlexNet for better vision.

The AlexNet comprises of 5 convolutional layers, 3 sub sampling layers and 3 fully connected layers. The main difference between the LeNet and AlexNet are the type of Feature Extractor. Here the non-linearity in the Feature Extractor module is used in AlexNet whereas Log sinusoid is used in LeNet. AlexNet uses dropout which is not observed in any other data sets of networking.

In the first layer, there are 96 11x11 filters are used at stride 4. The output volume size is 55x55x96. The AlexNet is trained on the GPU named GTX580 which is having a small amount of 3GB of memory. So, the CONV1 output will be halved and sent to two GPU's i.e. 55x55x48 is sent to each GPU. The 2nd, 4th, and 5th convolutional layers bits are related just to the part maps in the previous layer which dwell on the same GPU said in the figure. The kernels of the 3rd convolutional layer are associated with all kernel maps in the 2nd layer. The neurons in the fully connected layers are associated with all neurons in the past layer.

The 3rd, 4th, and 5th convolutional layers are associated with each other with no interceding pooling or standardization layers. The 3rd convolutional layer has 384 parts of size $3\times 3 \times 256$ associated with the (standardized, pooled) yields of the 2nd convolutional layer.

The fourth convolutional layer has 384 kernels of size $3 \times 3 \times 192$ and the fifth convolutional layer has 256 kernels of size $3 \times 3 \times 192$.The first two fully connected layers have 4096 neurons each.

In this the local response normalization in the normalization layer is used. There are two normalization layers present in the AlexNet architecture. The Deep Neural Network with ReLU Nonlinearity can train very fast than with the identical of the function tanh units. The ReLU considers quicker and more compelling training by mapping the negative esteems to zero and keeping up positive esteems. Signifying by the movement of a neuron figured by applying kernel i at position (x, y) and after that applying the ReLU nonlinearity, the response-normalized movement is expressed as :

$$c^{i}_{(x,y)} = d^{i}_{(x,y)} / \left(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} \left(d^{j}_{(x,y)} \right)^{2} \right)^{\beta}$$

This kind of response standardization actualizes a type of parallel hindrance roused by the sort found in genuine neurons, making rivalry for huge exercises among neuron yields registered utilizing different kernels. The test images are cropped to various portion areas and applied for classification.

2.2 Plant Leaf Disease Detection using Deep Learning and Convolutional Neural Network

Anandhakrishnan MG Joel Hanson1, Annette Joy2, Jerin Francis3 Department of Computer Science Engineering SCET, India
Volume 7 Issue No.3

An automated system designed to help to identify plant diseases by the plant’s appearance and visual symptoms could be of great help to amateurs in the gardening process and also trained professionals as a verification system in disease diagnostics. Advances in computer vision present an opportunity to expand and enhance the practice of precise plant protection and extend the market of computer vision applications in the field of precision agriculture.

Exploiting common digital image processing techniques such as colour analysis and thresholding were used with the aim of detection and classification of plant diseases. In machine learning and cognitive science, ANN is an information-processing paradigm that was inspired by the way biological nervous systems, such as the brain, process information. Neural networks or connectionist systems are a computational approach used in computer science and other research disciplines, which is based on a large collection of neural units (artificial neurons), loosely mimicking the way a biological brain solves problems with large clusters of biological neurons connected by axons. Each neural unit is connected with many others, and links can be enforcing or inhibitory in their effect on the activation state of connected neural units. Each individual neural unit may have a summation function which combines the values of all its inputs together. There may be a threshold function or limiting function on each connection and on the unit itself, such that the signal must surpass the limit before propagating to other neurons. These systems are self learning and trained, rather than explicitly programmed, and excel in areas where the solution or feature detection is difficult to express in a traditional computer program. Neural networks typically consist of multiple layers or a cube design, and the signal path traverses from front to back. Back propagation is the use of forward stimulation to reset weights on the "front" neural units and this is sometimes done in combination with training where the correct result is known. More modern networks are a bit more free flowing in terms of stimulation and inhibition with connections interacting in a much more chaotic and complex fashion.

Dynamic neural networks are the most advanced, in that they dynamically can, based on rules, form new connections and even new neural units while disabling others. The goal of the neural network is to solve problems in the same way that the human brain would, although several neural networks are more abstract. Modern neural network projects typically work with a few thousand to a few million neural units and millions of connections, which are still several orders of magnitude less complex than the human brain and closer to the computing power of a worm. New brain research often stimulates new patterns in neural networks. One new approach is using connections which span much further and link processing layers rather than always being localized to adjacent neurons. Other research being explored with the different types of signal over time that axons propagate, such as Deep Learning, interpolates greater complexity than a set of Boolean variables being simply on or off. Their inputs can also take on any value between 0 and 1. Also, the neuron has weights for each input and an overall bias. The weights are real numbers expressing importance of the respective inputs to the output. The bias is used for controlling how easy the neuron is getting to output 1. For a neuron with really big bias it is easy to output 1, but when the bias is very negative then it is difficult to output 1.

Neural Network Training

Training the deep convolutional neural network for making an image classification model from a dataset was proposed. Tensor Flow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well. In machine learning, a convolutional neural network is a type of feed -forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli in a restricted region of space known as the receptive field. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a convolution operation.

Convolutional neural networks (CNNs) consist of multiple layers of receptive fields. These are small neuron collections which process portions of the input image. The outputs of these collections are then tiled so that their input regions overlap, to obtain a higher-resolution representation of the original image; this is repeated for every such layer. Tiling allows CNNs to tolerate translation of the input image. Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters. They also consist of various combinations of convolutional and fully connected layers, with point wise nonlinearity applied at the end of or after each layer. A convolution operation on small regions of input is introduced to reduce the number of free parameters and improve generalization. One major advantage of convolutional networks is the use of shared weight in convolutional layers, which means that the same filter (weights bank) is used for each pixel in the layer; this both reduces memory footprint and improves performance. The convolutional neural network is also known as shift invariant or space invariant artificial neural network (SIANN), which is named based on its shared weights architecture and translation invariance characteristics. The convolutional layer is the essential building block of the convolutional neural network. The layer's parameters are comprised of a set of learnable kernels which possess a small receptive field but extend through the full depth of the input volume. Rectified Linear Units (ReLU) are used as substitute for saturating nonlinearities.

Deep CNN with ReLUs trains several times faster. This method is applied to the output of every convolutional and fully connected layer. Despite the output, the input normalization is not required; it is applied after ReLU nonlinearity after the first and second convolutional layer because it reduces top-1 and top-5 error rates. In CNN, neurons within a hidden layer are segmented into "feature maps." The neurons within a feature map share the same weight and bias. The neurons within the feature map search for the same feature. These neurons are unique since they are connected to different neurons in the lower layer. So for the first hidden layer, neurons within a feature map will be connected to different regions of the input image. The hidden layer is segmented into feature maps where each neuron in a feature map looks for the same feature but at different positions of the input image. Basically, the feature map is the result of applying convolution across an image. The convolutional layer is the core build ing block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input and producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input. Stacking the activation maps for all filters along the depth dimension forms the full output volume of the convolution layer. Every entry in the output volume can thus also be interpreted as an output of a neuron that looks at a s mall region in the input and shares parameters with neurons in the same activation map. When dealing with high-dimensional inputs such as images, it is impractical to connect neurons to all neurons in the previous volume because such network architecture does not take the spatial structure of the data into account. Convolutional networks exploit spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers: each neuron is connected to only a small region of the input volume. The extent of this connectivity is a hyper parameter called the receptive field of the neuron. The connections are local in space (along width and height), but always extend along the entire depth of the input volume. Such architecture ensures that the learnt filters produce the strongest response to a spatially local input pattern.

Three hyper parameters control the size of the output volume of the convolutional layer: The depth, stride and zero-padding.

1. Depth of the output volume controls the number of neurons in the layer that connect to the same region of the input volume. A ll of these neurons will learn to activate for different features in the input. For example, if the first Convolutional Layer takes the raw image as input, then different neurons along the depth dimension may activate in the presence of various oriented edges, or blobs of color.

2. Stride controls how depth columns around the spatial dimensions (width and height) are allocated. When the stride is 1, a new depth column of neurons is allocated to spatial positions only 1 spatial unit apart. This leads to heavily overlapping receptive fields between the columns, and also to large output volumes. Conversely, if higher strides are used then the receptive fields will overlap less and the resulting output volume will have smaller dimensions spatially.

3. Stride controls how depth columns around the spatial dimensions (width and height) are allocated. When the stride is 1, a new depth column of neurons is allocated to spatial positions only 1 spatial unit apart. This leads to heavily overlapping receptive fields between the columns, and also to large output volumes. Conversely, if higher strides are used then the receptive fields will overlap less and the resulting output volume will have smaller dimensions spatially. Parameter sharing scheme is used in convolutional layers to control the number of free parameters. It relies on one reasonable assumption: That if one patch feature is useful to compute at some spatial position, then it should also be useful to compute at a different position. In other words, denoting a single 2-dimensional slice of depth as a depth slice, we constrain the neurons in each depth slice to use the same weights and bias.

Since all neurons in a single depth slice are sharing the same parameterization, then the forward pass in each depth slice of the CONV layer can be computed as a convolution of the neuron's weights with the input volume (hence the name: convolutional layer). Therefore, it is common to refer to the sets of weights as a filter (or a kernel), which is convolved with the input. The result of this convolution is an activation map, and the set of activation maps for each different filter are stacked together along the depth dimension to produce the output volume. Parameter Sharing contributes to the translation invariance of the CNN architecture.

Another important layer of CNNs is the pooling layer, which is a form of nonlinear down sampling. Pooling operation gives the form of translation invariance; it operates independently on every depth slice of the input and resizes it spatially. Overlapping pooling is beneficially applied to lessen over fitting. Also in favour of reducing over fitting, a dropout layer is used in the first two fully connected layers. But the shortcoming of dropout is that it increases training time 2-3 times comparing to a standard neural network of the exact architecture. Bayesian optimization experiments also proved that ReLUs and dropout have synergy effects, which means that it is advantageous when they are used together. The advance of CNNs refers to their ability to learn rich mid-level image representations as opposed to hand-designed low-level features used in other image classification methods.

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 Methodology

In India, automating crop disease diagnosis is an important task, particularly for regions with few experts. Most current methods detect disease by analysing leaf images, particularly for diseases that manifest on the aerial part of the plant. To train a good classifier one requires a huge image dataset and the appropriate methods to extract relevant features from the images that represent the disease unambiguously. Image data also tends to be prone to effects of occlusion that make consistent analysis of the data hard.

We collected a dataset of 4 crops- Maize, Millet, Cotton and Wheat. Total 57 categories of various diseases have been covered.

Convolutional Neural Networks

A Convolutional Neural Network (ConvNet/CNN) 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 have the ability to learn these filters/characteristics.

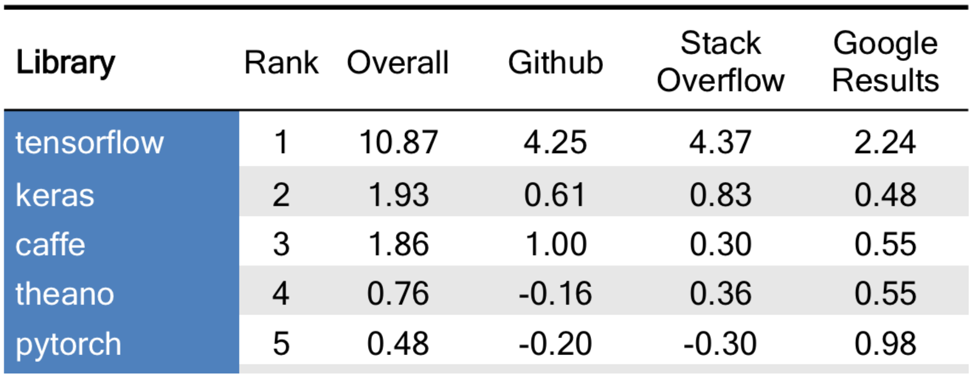
The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

TensorFlow

TensorFlow is an open source library for numerical computation and large-scale machine learning, created by the Google Brain team. TensorFlow bundles together a slew of machine learning and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

Why TensorFlow ?

With more and more research and real-life use cases going mainstream, we can see a big trend among programmers, and developers flocking towards the tool called TensorFlow. The popularity for TensorFlow is quite evident, with big names adopting TensorFlow for carrying out artificial intelligence tasks. Many popular companies such as NVIDIA, Twitter, Snapchat, Uber and more are using TensorFlow for all their major operations and research areas.



Library	Rank	Overall	Github	Stack Overflow	Google Results
tensorflow	1	10.87	4.25	4.37	2.24
keras	2	1.93	0.61	0.83	0.48
caffe	3	1.86	1.00	0.30	0.55
theano	4	0.76	-0.16	0.36	0.55
pytorch	5	0.48	-0.20	-0.30	0.98

Fig 3.1 Comparison of libraries

TensorFlow lite

TensorFlow Lite is TensorFlow's lightweight solution for mobile and embedded devices. It lets us run machine-learned models on mobile devices with low latency, so you can take advantage of them to do classification, regression or anything else you might want without necessarily incurring a round trip to a server.

It’s presently supported on Android and iOS via a C++ API, as well as having a Java Wrapper for Android Developers. Additionally, on Android Devices that support it, the interpreter can also use the Android Neural Networks API for hardware acceleration, otherwise it will default to the CPU for execution. In this article I’ll focus on how you use it in an Android app.

TensorFlow Lite is comprised of a *runtime* on which you can run *pre-existing* models, and a suite of tools that you can use to prepare your models for use on mobile and embedded devices.

It’s not yet designed for *training* models. Instead, you train a model on a higher powered machine, and then *convert* that model to the .TFLITE format, from which it is loaded into a mobile interpreter.

3.2 System Diagram

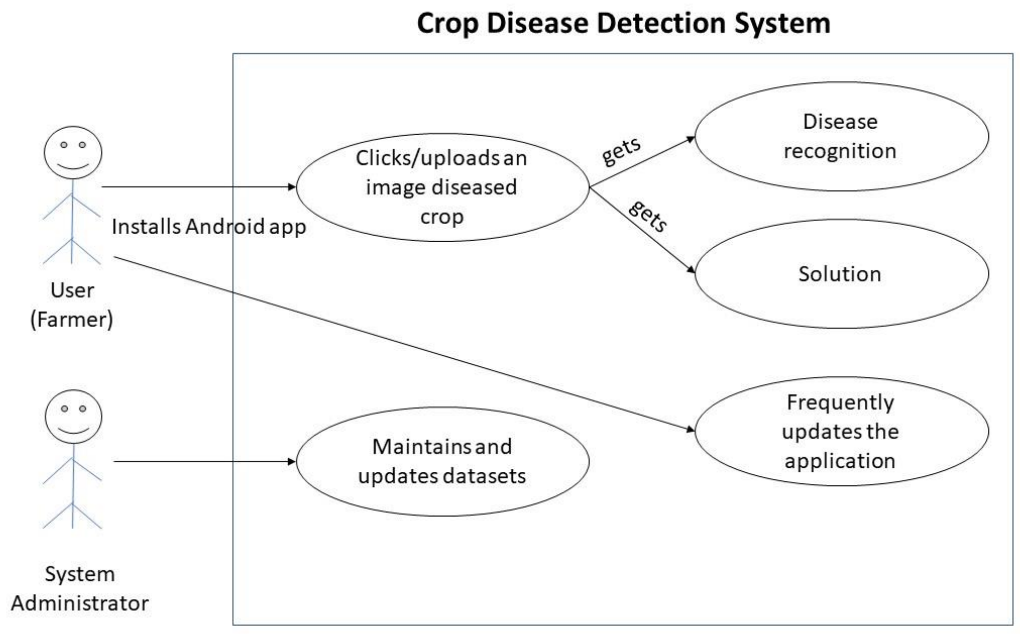

Crop Disease Detection System
User (Farmer)
Installs Android app
Clicks/uploads an image diseased crop
gets
gets
Disease recognition
Solution
Frequently updates the application
System Administrator
Maintains and updates datasets

Fig 3.2 Use case diagram

Actors:

1. Farmer:

The farmer first has to install the android application. Farmer has two options – Either start camera and take an image or upload an image of the infected / diseased crop. Then he has to select the good portion of the uploaded/clicked image. After this, top three predictions of crop diseases will be displayed in app. After clicking solution button, app will be redirected to a website showing most accurate disease, its symptoms, preventive measures, biological and chemical controls. The farmer will get to know about the disease as well as its solution. Farmer should frequently update application in order to avoid any new disease.

2. System administrator:

System administrator will have to play the role of maintaining and updating the database. He/she will also have to update the application frequently as per the requirements.

3.3 Algorithm Used

Convolutional Neural Network

In neural networks, Convolutional neural network is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

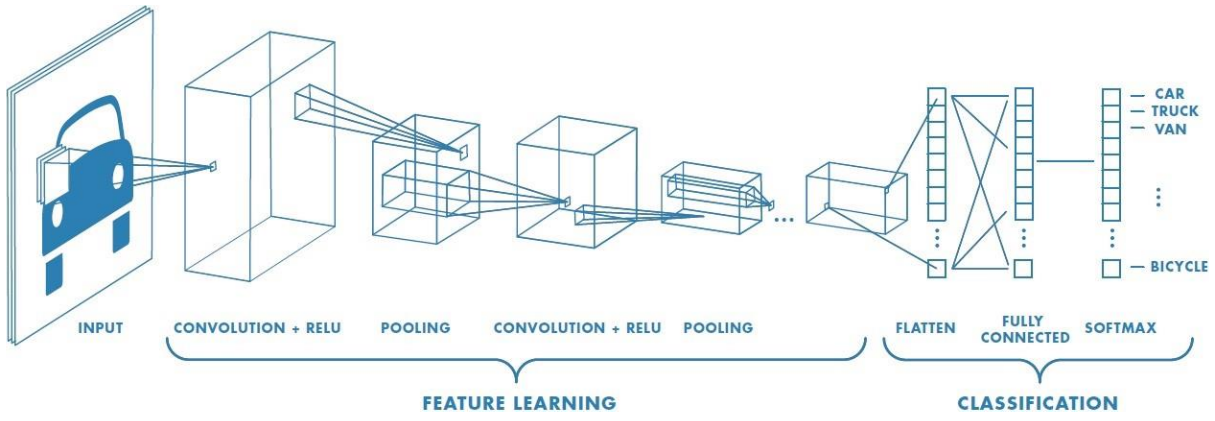


Fig 3.3 Architecture of CNN

Step 1: Convolution operation

Convolution is the first step in the process that convolutional neural networks undergo. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Here are the three elements that enter into the convolution operation:

1. Input image

2. Feature detector

3. Feature map

How exactly does the convolution operation work?

You can think of the feature detector as a window consisting of 9 (3x3) cells. Here is what you do with it:

1.You place it over the input image beginning from the top-left corner within the borders you see demarcated above, and then you count the number of cells in which the feature detector matches the input image.

2. The number of matching cells is then inserted in the top-left cell of the feature map.

3. You then move the feature detector one cell to the right and do the same thing. This movement is called and since we are moving the feature detector one cell at time, that would be called a stride of one pixel.

4. You will find in this example is that the feature detector's middle-left cell with the number 1 inside it matches the cell that it is standing over inside the input image. That's the only matching cell, and so you write "1" in the next cell in the feature map, and so on and so forth.

5. After you have gone through the whole first row, you can then move it over to the next row and go through the same process.

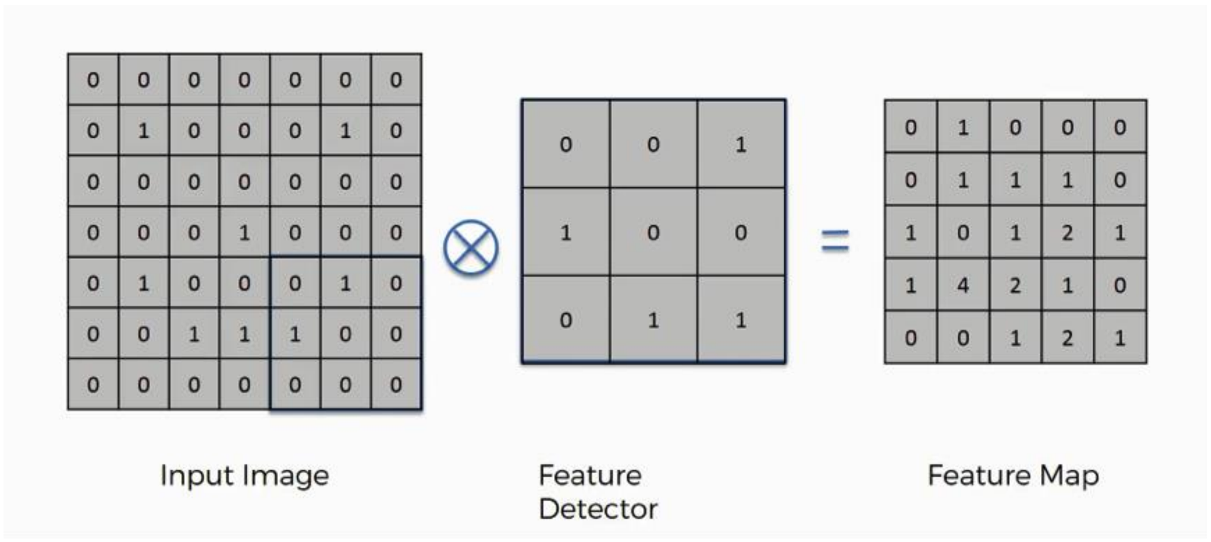


Fig 3.4 Convolution Operation

Step 1(b) : Rectified linear unit(ReLU)

The Rectified Linear Unit, or ReLU, is not a separate component of the convolutional neural networks' process. It's a supplementary step to the convolution operation. The purpose of applying the rectifier function is to increase the non-linearity in our images. The reason we want to do that is that images are naturally non-linear. When you look at any image, you'll find it contains a lot of non-linear features (e.g. the transition between pixels, the borders, the colours, etc.).

The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose an image when we put it through the convolution operation.

Rectification

What the rectifier function does to an image like this is remove all the black elements from it, keeping only those carrying a positive value (the grey and white colours).
The essential difference between the non-rectified version of the image and the rectified one is the progression of colours.

Step 2. Pooling

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or down sampling which reduces the dimensionality of each map but retains the important information. Spatial pooling can be of different types:

1.Max Pooling
2.Average Pooling
3.Sum Pooling

Max pooling take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

Step 3. Flattening

In this step we flatten our pooled feature map into one vector of input data that you then pass through the artificial neural network to have it processed further.

Step 4. Full connection

Here's where artificial neural networks and convolutional neural networks collide as we add the former to our latter. It's here that the process of creating a convolutional neural network begins to take a more complex and sophisticated turn.

We have three layers in the full connection step:

1. Input layer
2. Fully connected layer
3. Output layer

The input layer contains the vector of data that was created in the flattening step. The features that we distilled throughout the previous steps are encoded in this vector. The Fully Connected layer is configured exactly the way its name implies: it is fully connected with the output of the previous layer. Fully-connected layers are typically used in the last stages of the CNN to connect to the output layer and construct the desired number of outputs.

The full connection process practically works as follows:

- The neuron in the fully-connected layer detects a certain feature; say, a nose.
- It preserves its value.
- It communicates this value to available classes.
- All classes check out the feature and decide whether it's relevant to them.

3.4 Specifications

1. Camera Permission : User needs to allow camera usege.
2. Storage read and write permission : User needs to allow storage permission

3.5 Hardware Requirement

- Android phone with good quality camera
- 105 MB free space
- RAM required 4 GB

3.6 Software Requirement

- Mobile phones having android operating system
- Minimum API level 15 (Ice-cream Sandwich)

CHAPTER 4

PERFORMANCE ANALYSIS

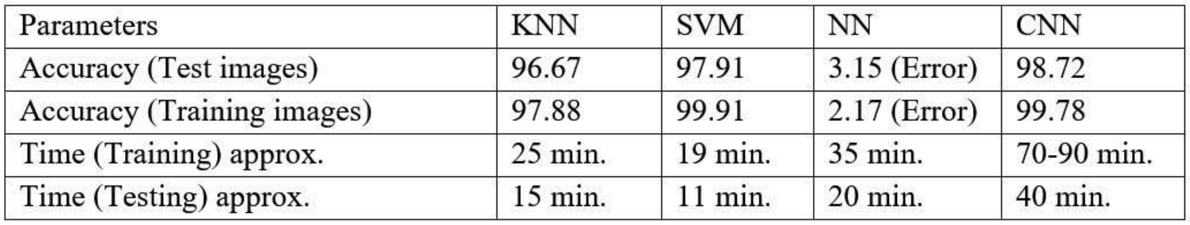
4.1 Comparison between Algorithms

Neural Network

Artificial Neural Network is a kind of artificial intelligence that controls human mind's function. It is a nonparametric approach. Non- parametric approach has no assumption about the data and where correctness depends on the no. of inputs and network. Ann learns from the environment, and stores the experiential knowledge. ANN is a collection of layer basically it has 2 layers i.e. input and output, but some system has hidden layers. Every layer has no. of neurons. They connected with each other by a weighted link.

Support Vector Machines

SVM is a binary Non- parametric classifier. Some SVM supports multiclass classifiers also. SVMs are learning systems that use a hypothesis space of linear functions in a hyper space. SVM is trained with a Learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. The aim of Classification via SVM is to find a 'computationally efficient' way of learning good separating hyper planes in a hyperspace, where 'good' hyper planes mean ones optimizing the generalizing bounds and by 'computationally efficient' we mean algorithms able to deal with sample sizes of very high .



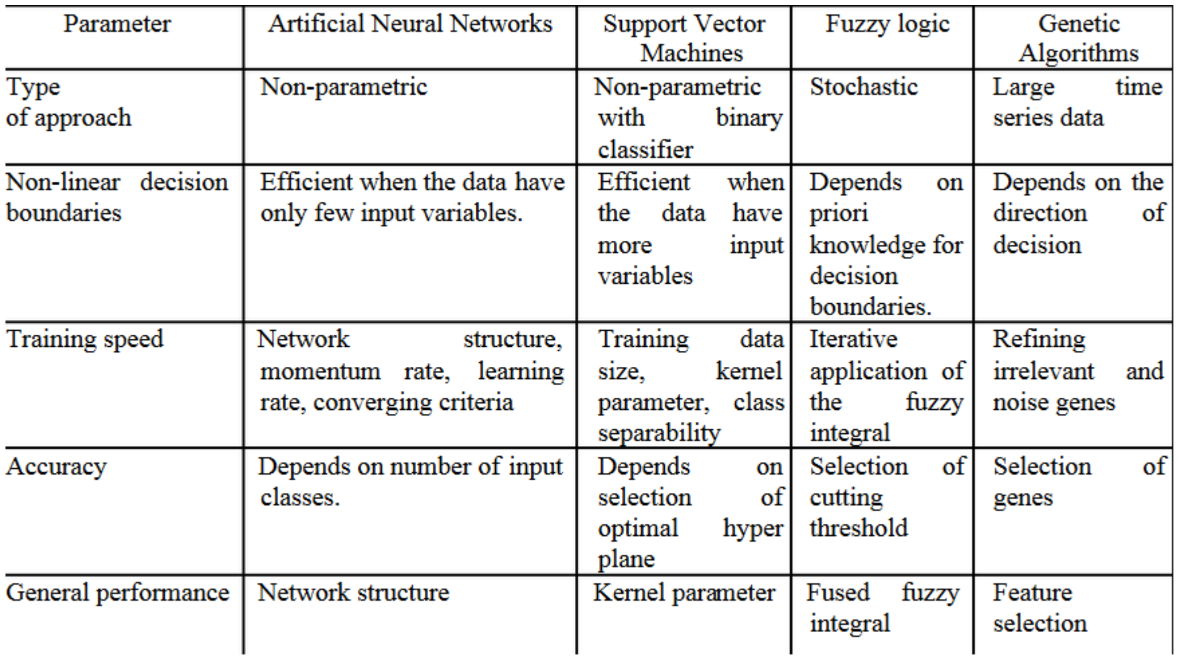
Parameters	KNN	SVM	NN	CNN
Accuracy (Test images)	96.67	97.91	3.15 (Error)	98.72
Accuracy (Training images)	97.88	99.91	2.17 (Error)	99.78
Time (Training) approx.	25 min.	19 min.	35 min.	70-90 min.
Time (Testing) approx.	15 min.	11 min.	20 min.	40 min.

k-Nearest neighbour

One example for an instance-based learning algorithm is the k-Nearest Neighbour (kNN) algorithm. It uses the k nearest neighbours to make the decision of class attribution directly from the training instances themselves. Usually, Euclidean distance is used as distance metric. The decision for attaching the sample in question to one of the several classes is based on the majority vote of its k nearest neighbours. An odd number should be chosen for k to allow for a definite majority vote.

Convolutional Neural Network

ConvNet/CNN 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 have the ability to learn these filters/characteristics.



Parameter	Artificial Neural Networks	Support Vector Machines	Fuzzy logic	Genetic Algorithms
Type of approach	Non-parametric	Non-parametric with binary classifier	Stochastic	Large time series data
Non-linear decision boundaries	Efficient when the data have only few input variables.	Efficient when the data have more input variables	Depends on priori knowledge for decision boundaries.	Depends on the direction of decision
Training speed	Network structure, momentum rate, learning rate, converging criteria	Training data size, kernel parameter, class separability	Iterative application of the fuzzy integral	Refining irrelevant and noise genes
Accuracy	Depends on number of input classes.	Depends on selection of optimal hyper plane	Selection of cutting threshold	Selection of genes
General performance	Network structure	Kernel parameter	Fused fuzzy integral	Feature selection

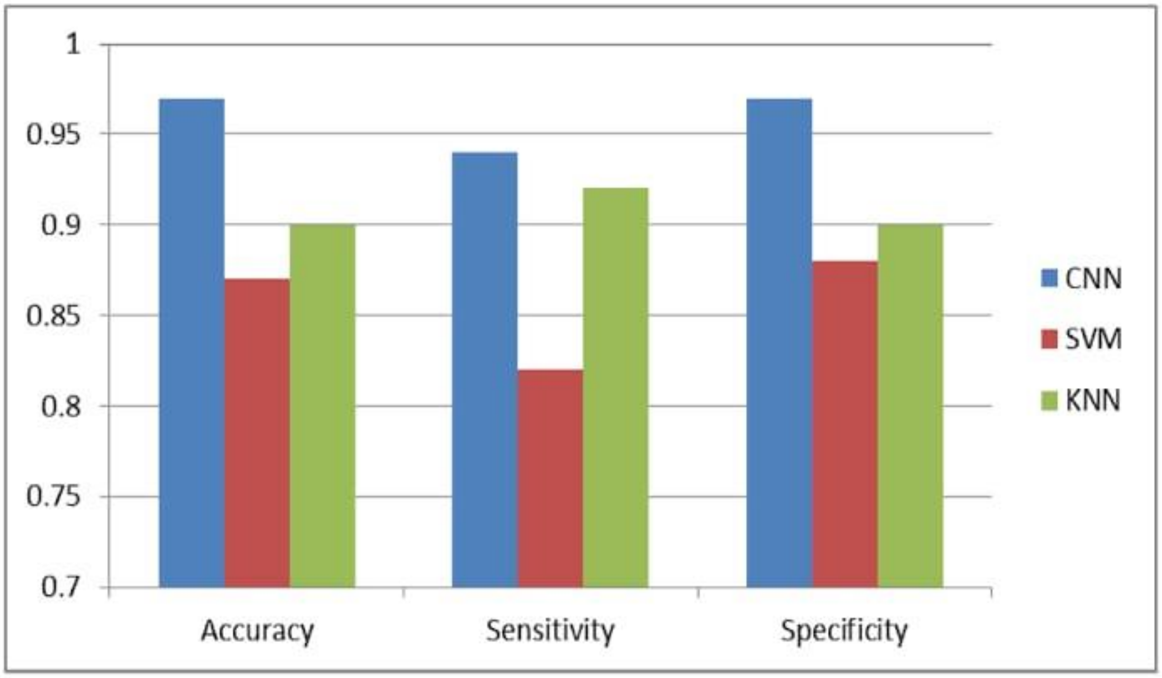


Fig 4.1 Graphical Comparison between algorithms

4.2 Graphical User Interface

When a user opens Crop Disease Detection application from an Android phone, a splash screen appears for 5 seconds as shown in fig 4.2.1. UI is very user-friendly and simple. User is given two options- START CAMERA and UPLOAD AN IMAGE (Fig 4.2.2). Android phone should have a good quality camera having at least 5 MP camera. User should click a clear image in order to get more accurate results.

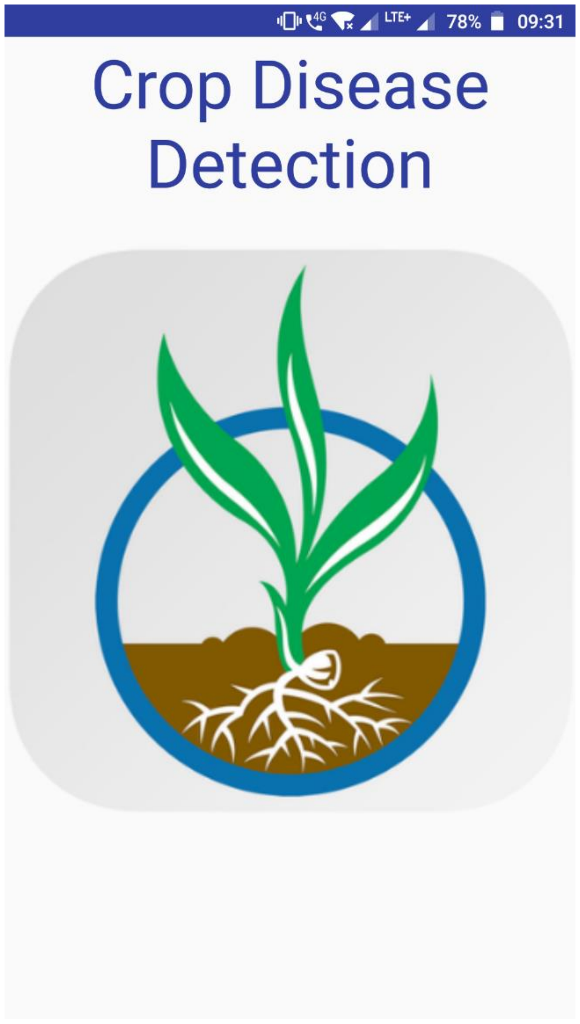


Fig. 4.2.1 Splash Screen

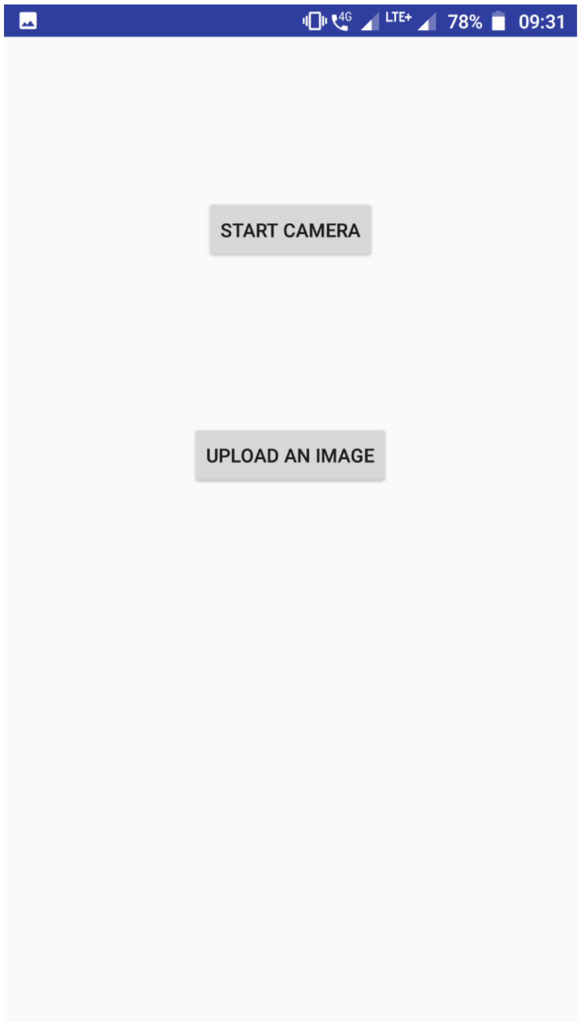
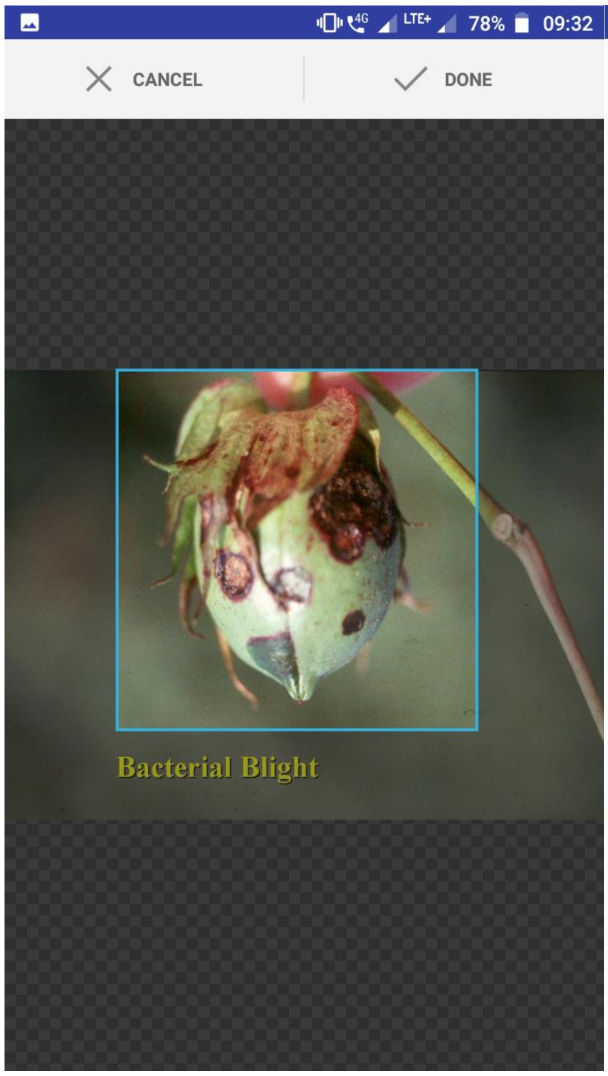
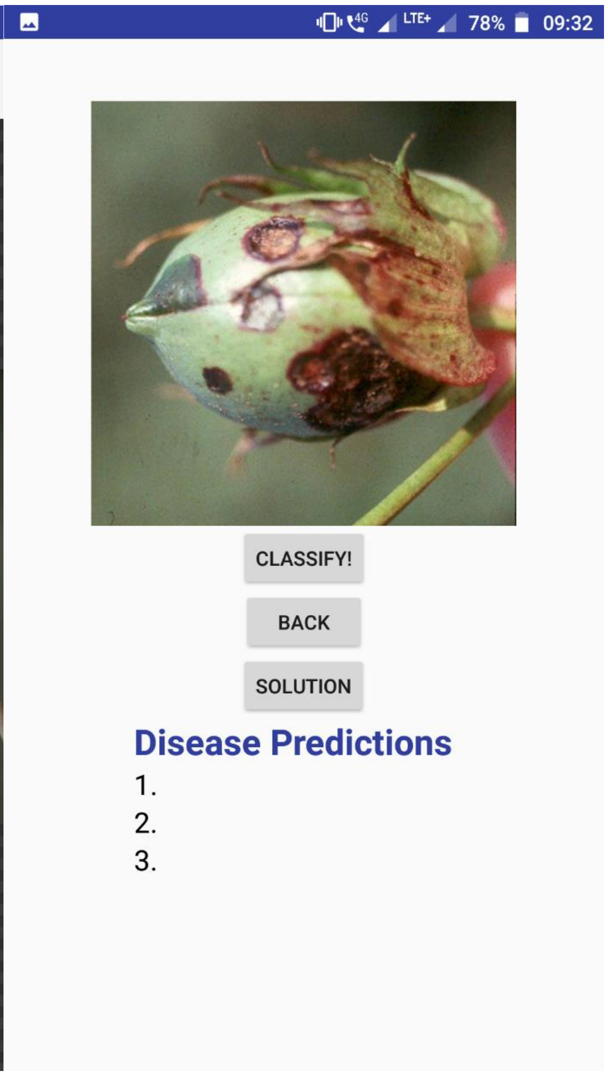


Fig. 4.2.2 Choose one option

Here, we have uploaded an internet downloaded image. User needs to specify important diseased section of the plant by cropping and resizing the image (Fig. 4.2.3).

After clicking Done button, a screen shown in Fig. 4.2.4 appears. Click Classify button.





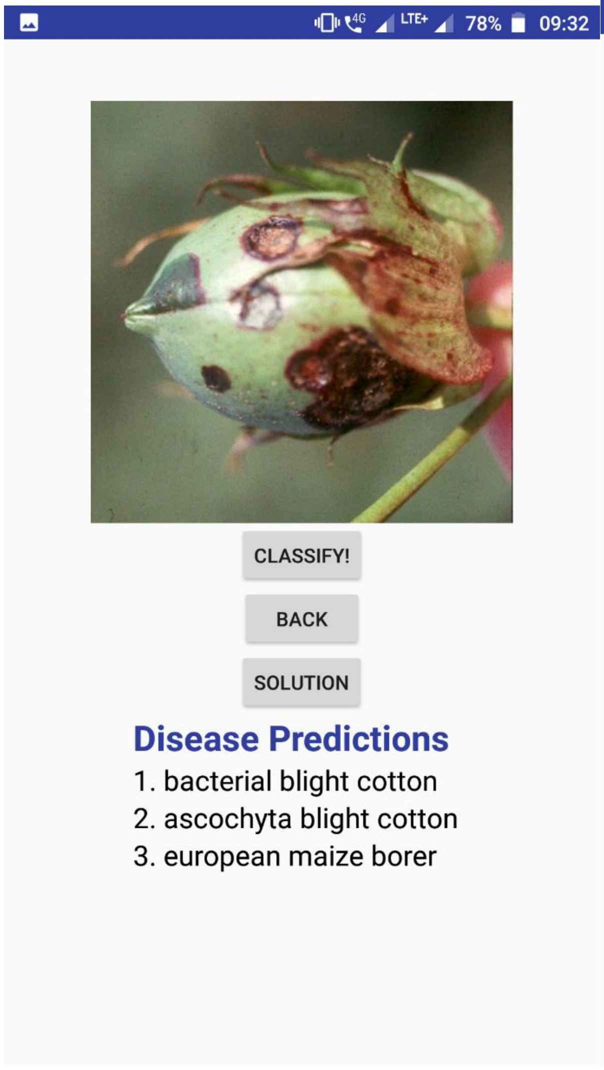
Disease Predictions
1.
2.
3.

Fig. 4.2.3 Select diseased region

Fig. 4.2.4 Press Classify!

After clicking classify, three most accurate predictions are displayed (Fig. 4.2.5). In our example, Bacterial blight cotton is most accurate.

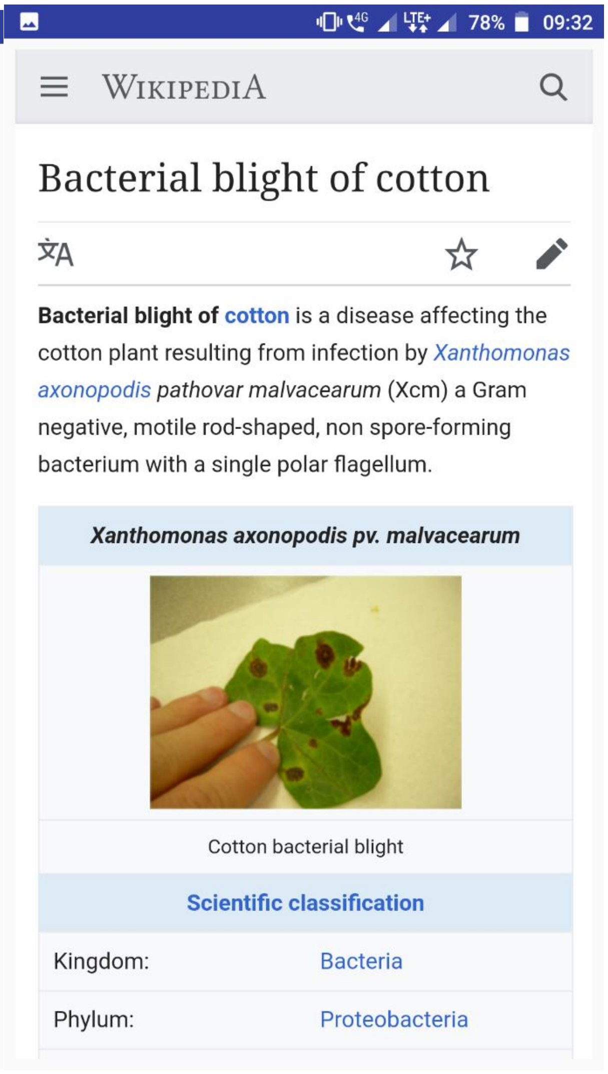
To get the symptoms, preventive measures, biological and chemical controls of disease, user needs to press Solution button. Internet connection with good speed is required to get information of disease. After clicking solution button, app will be redirected to a website showing most accurate disease. (Fig. 4.2.6)



Disease Predictions

1. bacterial blight cotton
2. ascochyta blight cotton
3. european maize borer

Fig. 4.2.5 Predictions displayed



Bacterial blight of cotton

Bacterial blight of cotton is a disease affecting the cotton plant resulting from infection by *Xanthomonas axonopodis pathovar malvacearum* (Xcm) a Gram negative, motile rod-shaped, non spore-forming bacterium with a single polar flagellum.

Xanthomonas axonopodis pv. malvacearum

Cotton bacterial blight

Scientific classification

Kingdom:	Bacteria
Phylum:	Proteobacteria

Fig. 4.2.6 After pressing Solution button

4.3 Testing

4.3.1 App installation

Test name: App installation

Description: To check functionality of installation process

Expected result: Installation should be done without any bug

Actual result: Installation done.

Status: Pass

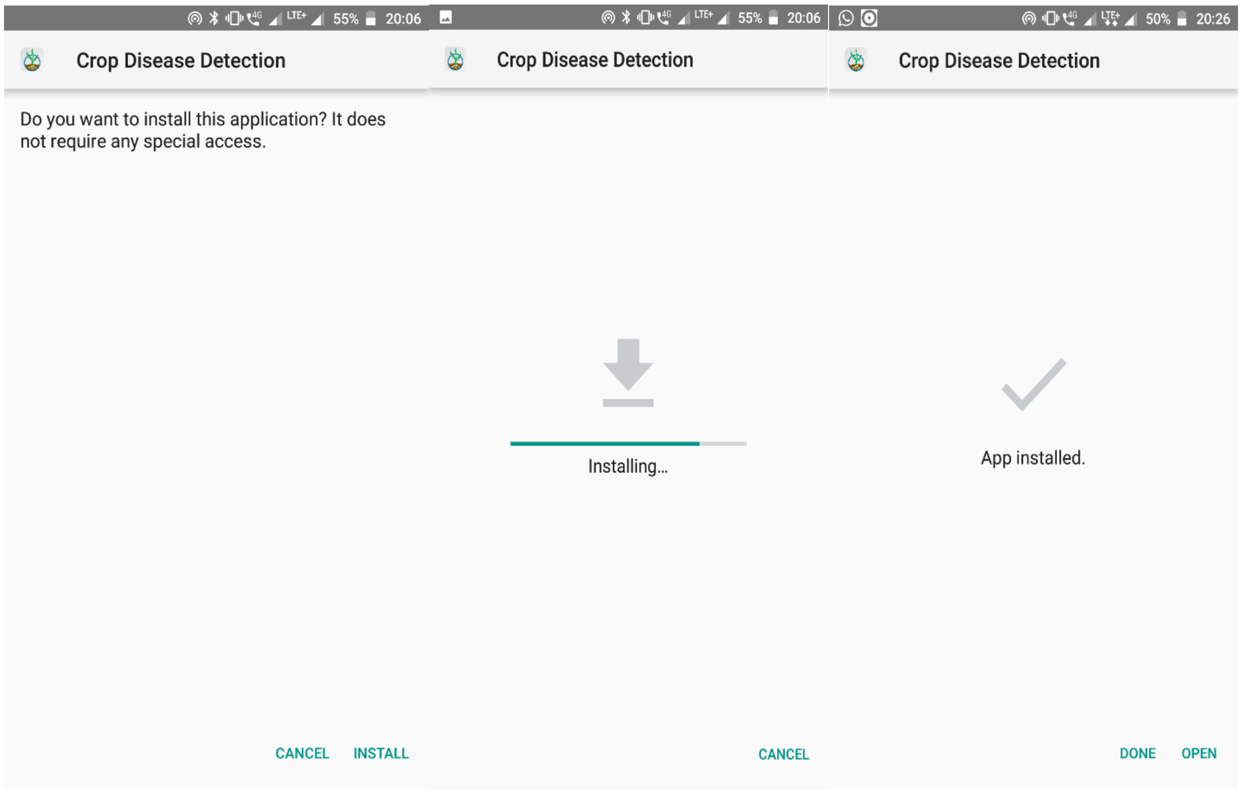


Fig. 4.3.1 Installation Process of the app

4.3.2 Maize Testing

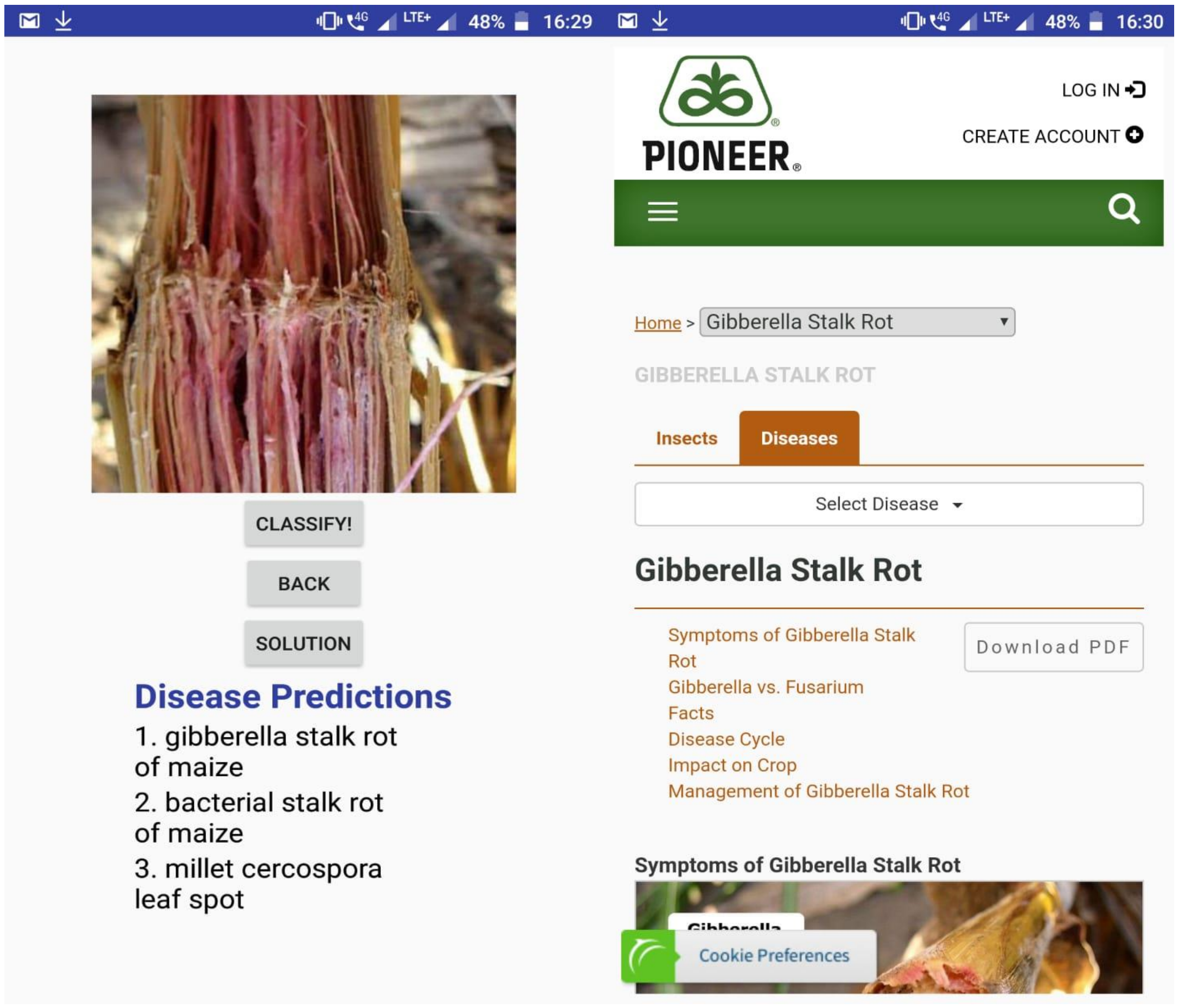
Test name: Maize Testing

Description: To check whether app gives correct result for Maize crop

Expected result: Gibberella stalk rot

Actual result: Gibberella stalk rot

Status: Pass



Disease Predictions

1. gibberella stalk rot of maize
2. bacterial stalk rot of maize
3. millet cercospora leaf spot

Home > Gibberella Stalk Rot

GIBBERELLA STALK ROT

Insects | Diseases

Select Disease

Gibberella Stalk Rot

Symptoms of Gibberella Stalk Rot
Gibberella vs. Fusarium
Facts
Disease Cycle
Impact on Crop
Management of Gibberella Stalk Rot

Download PDF

Symptoms of Gibberella Stalk Rot

Fig 4.3.2 Maize Testing

4.3.3 Wheat Testing

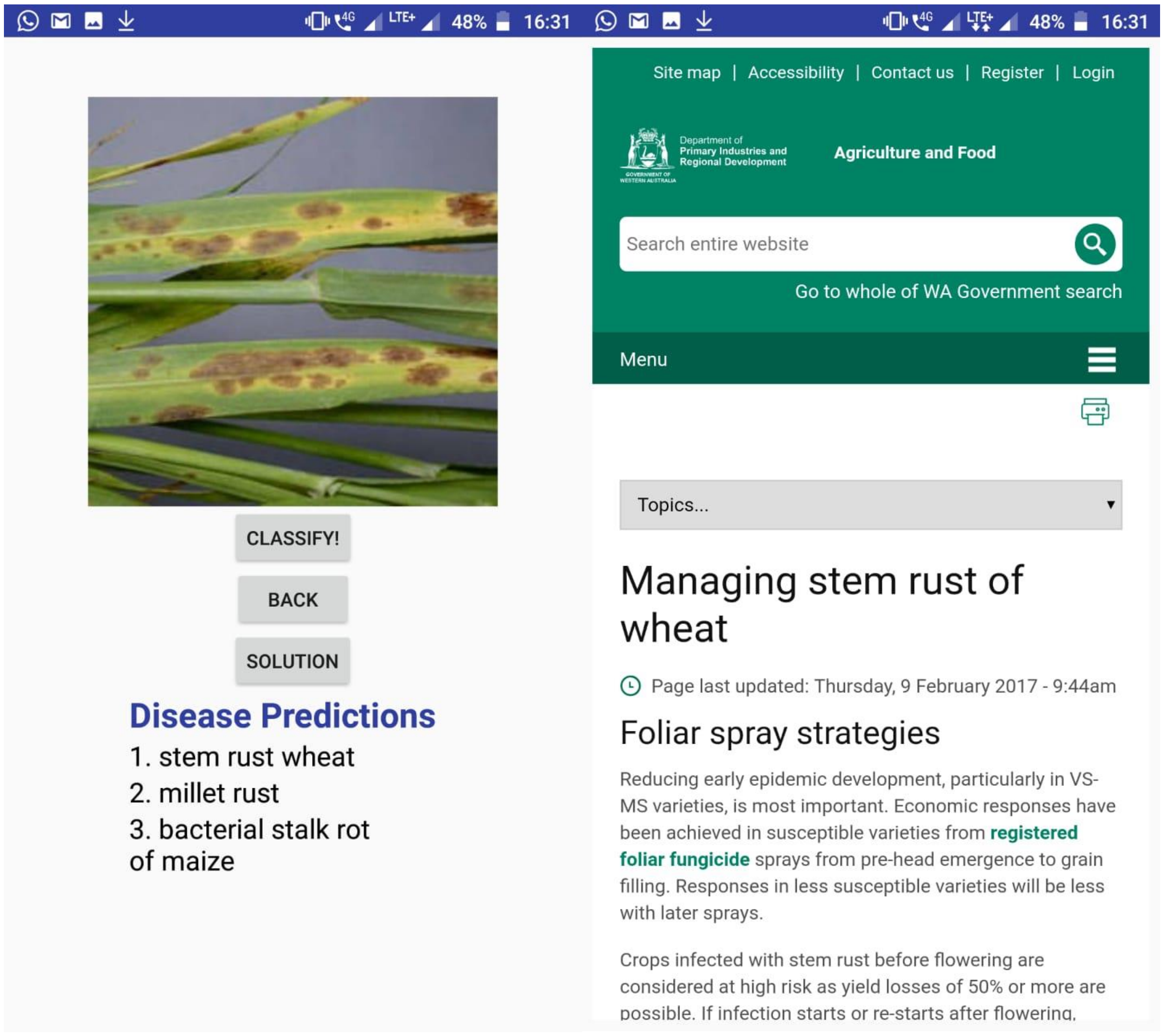
Test name: Wheat Testing

Description: To check whether app gives correct result for Wheat crop

Expected result: Stem rust wheat

Actual result: Stem rust wheat

Status: Pass


48%
16:31
CLASSIFY!
BACK
SOLUTION
Site map | Accessibility | Contact us | Register | Login
Department of Primary Industries and Regional Development
Agriculture and Food
Search entire website
Go to whole of WA Government search
Menu
Topics...

Disease Predictions

1. stem rust wheat
2. millet rust
3. bacterial stalk rot of maize

Managing stem rust of wheat

Page last updated: Thursday, 9 February 2017 - 9:44am

Foliar spray strategies

Reducing early epidemic development, particularly in VS-MS varieties, is most important. Economic responses have been achieved in susceptible varieties from **registered foliar fungicide** sprays from pre-head emergence to grain filling. Responses in less susceptible varieties will be less with later sprays.

Crops infected with stem rust before flowering are considered at high risk as yield losses of 50% or more are possible. If infection starts or re-starts after flowering,

Fig 4.3.3 Wheat Testing

4.3.4 Millet Testing

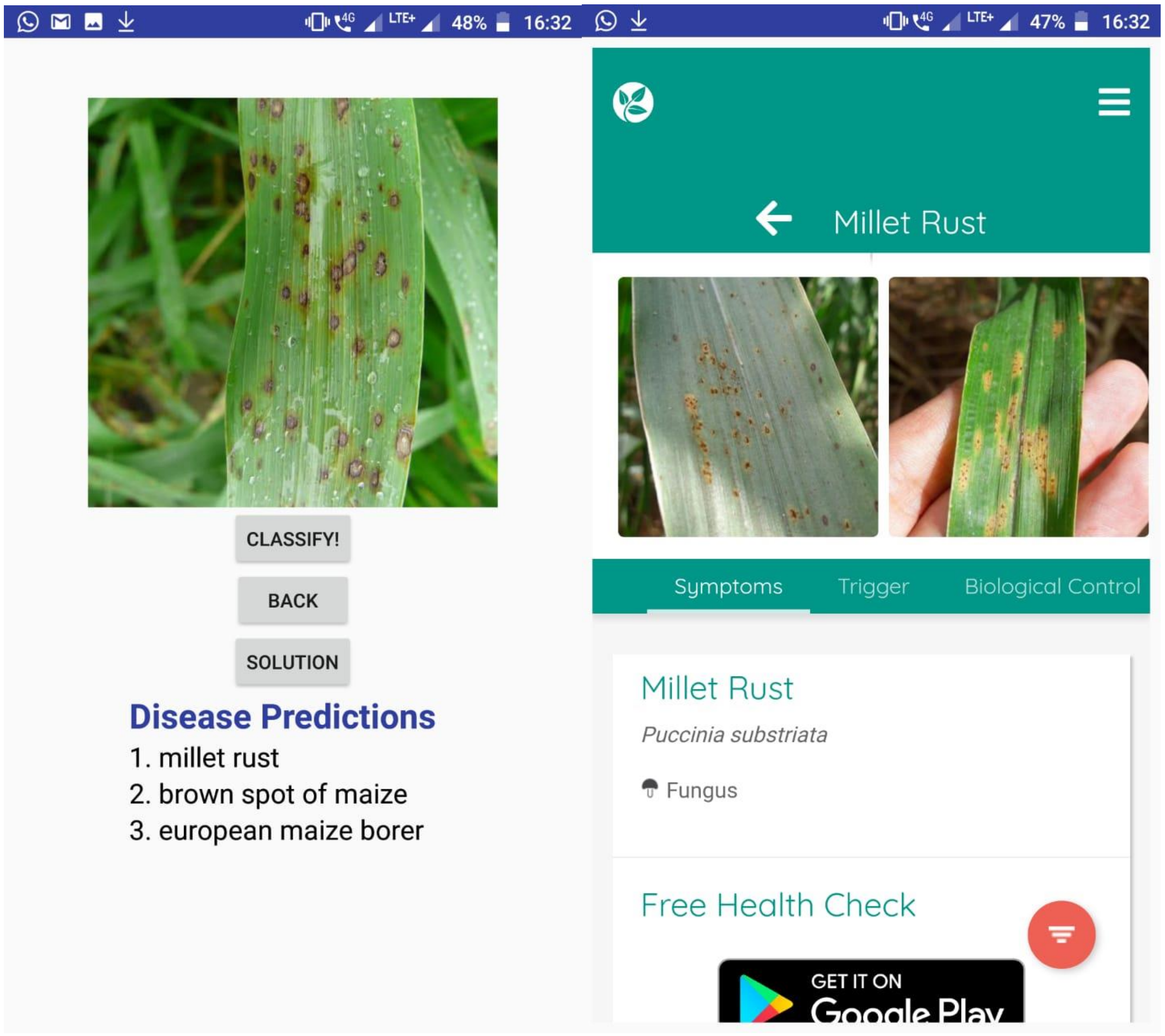
Test name: Millet Testing

Description: To check whether app gives correct result for Millet crop

Expected result: Millet rust

Actual result: Millet rust

Status: Pass



Disease Predictions
1. millet rust
2. brown spot of maize
3. european maize borer

Millet Rust
Puccinia substriata
Fungus

Free Health Check

Fig 4.3.4 Millet Testing

CHAPTER 5

CONCLUSION

5.1 Conclusion

This application tries to solve the main problem of plants by detecting their diseases. It helps users to understand the crop disease and take preventive measures to protect the crop. It also provides solution of the detected disease so that the crop is prevented from further damage. Farmers can be benefited with the help of this app as they only need to take a picture of the leaf of the plant and upload it on the app and the app will detect its disease and provide the preventive measures in fraction of seconds as this app works on the internet.

5.2 Future Scope

An extension of this study will be on gathering images for enriching the database and improving accuracy of the model using different techniques of fine-tuning and augmentation. Furthermore, future work will involve spreading the usage of the model by training it for plant disease recognition on wider land areas, combining aerial photos captured by drones and convolution neural networks for object detection**.**

5.3 Applications/ Utility

• Used for detecting crop diseases

• Provides solution for the diseases

• Saves money

• Helps to prevent further damage of the crops

REFERENCES

[1] https://en.wikipedia.org/wiki/Convolutional_neural_network

[2]https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn

[3]https://www.researchgate.net/publication/304308800_Deep_Neural_Networks_Based_Recognition_of_Plant_Diseases_by_Leaf_Image_Classification

[4]https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks

[5]https://www.researchgate.net/publication/225594092_Convolutional_Neural_Networks_for_Image_Processing_with_Applications_in_Mobile_Robotics

[6]https://www.researchgate.net/publication/325116934_Image_classification_using_Deep_learning

[7] http://cs231n.stanford.edu/

[8] https://www.researchgate.net/publication/237609215_Comparison_of_Advanced_Techniques_of_Image_Classification

[9] https://machinelearningmastery.com/introduction-python-deep-learning-library-tensorflow/

[10] https://www.tensorflow.org/

[11] https://medium.com/tensorflow/using-tensorflow-lite-on-android-9bbc9cb7d69d

[12] Review of Image Classification Methods and Techniques , Maneela Jain , Pushpendra Singh Tomar , International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 8, August - 2013

[13] Survey On Image Classification Methods In Image Processing Chaitali Dhaware[1], Mrs. K. H. Wanjale[2] , International Journal of Computer Science Trends and Technology (IJCST) – Volume 4 Issue 3, May - Jun 2016

ACKNOWLEDGEMENT

We are profoundly grateful to **Prof. Komal S. Gandle** for her expert guidance and continuous encouragement throughout to see that this seminar completes and achieves its target since its commencement to its completion. We would like to express deepest appreciation towards **Prof. Chitra M. Gaikwad**, Head of The Department of Information Technology whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Information Technology Department who helped us directly or indirectly during this course of work.

Submitted by:

Sneha Kadam (BE15F03F029)

Manali Shah (BE15F03F051)

Prachi Naik (BE15F06F039)

Vrushali Potdar (BE15F06F042)