E-governance of Agriculture

Submitted in partial fulfillment of the requirements
of the degree of

BACHELOR OF ENGINEERING
(Computer Engineering)

by

1.Mr. Aniket Gode (BE/A/127)
2. Mr. Deepak Singh (BE/A /132)
3. Mr. Vikas Mandal (BE/A/168)

Guide
Prof. Dr. Subhash K. Shinde

Department of Computer Engineering
Lokmanya Tilak College Of Engineering
Sector-4, Koparkhairne, Navi Mumbai
(2020-2021)

Certificate

This is to certify that the project entitled "**E-governance of Agriculture**" is a bonafide work of

1. "**Aniket Gode**" **(127)**
2. "**Deepak Singh**" **(132)**
3. "**Vikas Mandal**" **(168)**

submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Engineering**" in "C**omputer Engineering**".

(Prof. Dr. Subhash Shinde)

Guide

( )

Co-Guide / External guide

(Prof. R.D. Gawali)
Head of Department

(Dr. Vivek Sunnapwar)
Principal

<u>Project Report Approval for B.E</u>

The project report entitled "**E-Governance for Agriculture"** by Aniket Gode (BE/A/127), Deepak Singh(BE/A/132), Vikas Mandal(BE/A/168) is approved for the award of "**Bachelor of Engineering"** degree in "**Computer Engineering".**

Examiners

1. ______________________

2. ______________________

Date:
Place: Koparkhairane, Navi Mumubai

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principals of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data / fact / source in my submission. I 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.

1. Aniket Gode (BE/A/127)

2. Deepak Singh (BE/A/132)

3. Vikas Mandal (BE/A/168)

TABLE OF CONTENTS

Abstract

E-governance in Agriculture is one of practically important issues in Agricultural Development in India. Application of E-governance includes in Deciding the suitable crops to grow in specific region, Recognizing the Disease in plant, tracking farmers in specific reasons for disbursing benefits. The heart of the problem lies in the ability that can a) Decide the crops to be grown in specific reason depending on the Weather, Soil Structure and Water Supply. b) Recognizing the disease of the plant and then coming up with the correct reason for that disease and coming up with all the possible cures. c) Converting all the English text into regional language. d) Fertilizer Recommendation. The main objective of this project is to ensure hassle-free experience to Food raisers of the nation.

Acknowledgement

I remain immensely obliged to Prof. Dr. S.K. Shinde for providing me with the idea of this topic, and for his/her invaluable support in gathering resources for me either by way of information or computer also his/her guidance and supervision which made this project successful.

We would like to thank mini project Coordinators, Dr. S.K. Shinde, Vice Principal and Head, Computer Engineering Department, and Dr. Vivek Sunnapwar, Principal, LTCoE.

I am also thankful to faculty and staff of Computer Engineering Department and Lokmanya Tilak College of Engineering, Navi Mumbai for their invaluable support.

I would like to say that it has indeed been a fulfilling experience for working out this project topic.

(Aniket Gode - 127) (Deepak Singh - 132) (Vikas Mandal- 168)

List of Figures

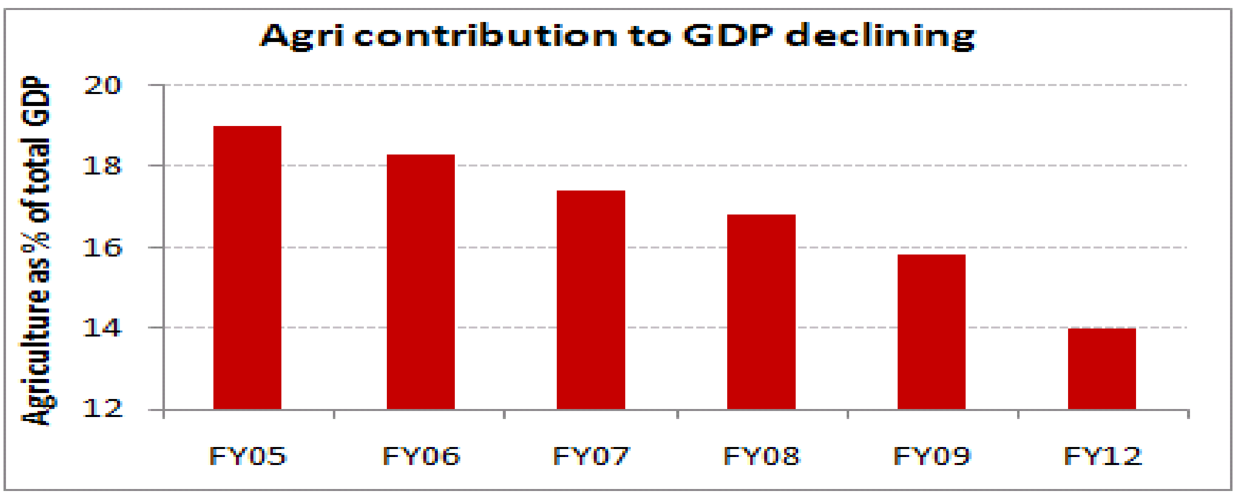
1. Introduction

1.1 Introduction:

According to the latest report published by UNESCO (United Nations Educational, Scientific and Cultural Organisation) India has the highest adult illiteracy rate in the world. Whereas on the other-hand A recent government study estimated that 32 percent of India's rural population is unable to read, compared to 15 percent in urban areas. For farmers, that percentage may be even higher.

Therefore; It's not surprising, then, that dealing with the modern challenges of agriculture proves to be difficult for many Indian farmers. Limited access to information due to low literacy rates directly translates into low-efficiency, low-productivty crops, trapping many farmers in a vicious cycle of poverty.

Agriculture accounts for up to **18 percent of India's GDP** and employs more than half of the country's workers. Yet agricultural communities are among the most deprived communities in the nation.



According to survey, many Indian farmers still primarily pass information from generation to generation verbally, as they have for centuries due to which the upcoming generation also produces the same crops without knowing the that the alternative crops can also be grown on the same field which have similar climatic requirement of previous crop which can fullfil the government requirement and can produce money from the same.

Despite these obstacles, as well as asymmetries among India's urban and rural populations, India lays claim to a surprisingly high rate of mobile phone penetration. There are nearly a billion mobile phones in India today—and it has been predicted that all 1.4 billion residents will have a mobile phone by 2020. A recent port from the United Nations revealed that more Indians have access to mobile phones than toilets.

“By now, everyone has a mobile phone even in the poorest villages,”. So we decided to arm today’s Indian farmers with an app designed to meet them where they’re at—illiterate and in need of a better way to track their crops. “We wanted to keep things as simple as possible, considering language and language barrier issues across India,”

1.2 Motivation:

Food is one of the most important pillars of society. Without food, there is eventually no life. Our main motive to do this project was to help our farmers who are the food providers of our great country. The reason we are able to feed ourselves daily is the laborious tasks that farmers undertake day and night while asking for very little in return. Being Computer Engineering students, our team's main motive was to use our technical knowledge and produce something which can help our farmers in a productive way.

1.3 Statement of the Problem:

In past few decades the population in India has been growing exponentially. Government of India takes due measures to procure and provide the essential food items and provide it too the needed ones. Despite of the efforts of government, there is a huge gap between demand and supply of essential food items to needed one, due to less number of farmers aware of what and how to grow, farmers usually grow the similar crops. Our proposal is to build an Application for all the above problems. The main objective of this project is to ensure hassle-free experience to Food raisers of the nation. The proposed system can be mainly used to Identify the crops to be grown based on the geographical location and government needs, Detect diseased plant with disease and provide the best treatment for the plant and give an option of buying the required pesticides and fertilizers through government vendors at minimal rate.

2. Literature Survey

2.1 Survey Existing System

We have studied various existing systems which have already been implemented at some scale, some of them are listed as;

A) Crop Prediction System:

→ Agricultural Production Output Prediction Using Supervised Machine Learning Techniques

Two supervised classification machine learning formula has been enforced during this study. the choice Tree LearningID3 (Iterative Dichotomiser 3) and KNNR discover the patterns within the knowledge set containing average temperature and precipitation worth obtained throughout the cropping amount of six major crops in 10 major cities of Bangladesh for the past twelve years and provides the prediction. ID3 uses the choice tree table that consists of the ranges of the precipitation, temperature and yield knowledge. The research provides an answer to the current downside that was much required for farmers in People's Republic of Bangladesh. Though the research is restricted to some mounted dataset, the long run ahead promises addition of a lot of knowledge which will be analyzed with more machine learning techniques to come up with crop predictions with higher exactness. Moreover, the analysis will result in profits and invention of advanced farming techniques which will improve our economy and can facilitate United States stand out as a technologically advanced country.

→ Machine learning approach for forecasting crop yield based on climatic parameters

The present study provides the potential use of information mining techniques in predicting the crop yield supported the environmental condition input parameters. The developed webpage is user friendly and therefore the accuracy of predictions square measure higher than seventy-five per cent all told the crops and districts designated within the study indicating higher accuracy of prediction. The user-friendly web content developed for predicting crop yield may be utilized by any user their alternative of crop by providing environmental condition knowledge of that place.

B) Disease Prediction System:

→ Plant Disease Detection using CNN

The present study provides the potential use of image recognition technique in predicting the diseased crop. The developed code base is complex and also the accuracy of predictions measures lower than 80%. It deals with diseases of 28 species of crops.

2.2 <u>Limitation Existing system or research gap</u>

We studied various existing systems and found out various drawbacks in the current existing system. They are as follows:-

- The Existing System does not recommend the crop to be yield according to the Geographical Conditions.
- The language barrier amongst the farmers are not overcome by the existing systems which leads to misunderstanding of information.
- The Existing System doesn't detect the crop disease precisely by them the image is recognized by the agriculture experts.
- The Existing Systems are Less Reliable due to less accuracy in Image Detection.
- The Existing Systems doesn't recommend the pesticides related to crop diseases.
- The Existing System does not recommends the fertilizer depending on Soil Nutrients.
- There is no system with all the features in one application. To avail different kinds of services, farmers have to be dependent on the various kinds of applications.

These are some of the drawbacks of the existing systems, which we will try to overcome.

2.3 Problem Statement and Objective

"In past few decades the population in India has been growing exponentially. Government of India takes due measures to procure and provide the essential food items and provide it too the needed ones. Despite of the efforts of government, there is a huge gap between demand and supply of essential food items to needed one, due to less number of farmers aware of what and how to grow, farmers usually grow the similar crops. Our proposal is to build an Application for all the above problems. The main objective of this project is to ensure hassle-free experience to Food raisers of the nation. The proposed system can be mainly used to Identify the crops to be grown based on the geographical location and soil nutrients, detect diseased plant with disease and provide the best treatment for the plant and give an recommendation about fertilizers depending upon the Soil Nutrients."

Objectives;

- Decide the crops to be grown in specific reason depending on the Weather, Soil Structure, and Water Supply.
- Recognizing the disease of the plant and then coming up with the correct reason for that disease and coming up with all the possible cures.
- Converting all the English text into regional language.
- Recommending Fertilizer depending upon the Soil Nutrients.

2.4 Scope of the Work

The intended purpose of this project is to: -

- To provide farmers with the modern technologies.
- To promote importance of technology in agriculture sector.
- To help farmers grow the most appropriate crops with respect to Weather and Soil Conditions.
- To recognize if the crop is diseased or not and if diseased come up with the suitable remedies.
- To overcome the language barrier for the farmers who are not able to read and write.

3. Proposed System

3.1 Algorithm:

A) Image Recognition:

What is Image Recognition?

The first question you may have is what the difference is between computer vision and image recognition. Indeed, computer vision has been vigorously developed by Google, Amazon and many AI developers, and the two terms "computer vision" and "image recognition" may have used interchangeably. Computer vision (CV) is to let a computer imitate human vision and *take action*. For example, CV can be designed to sense a running child on the road and produces a warning signal to the driver. In contrast, image recognition is about the pixel and pattern analysis of an image to recognize the image as a particular object. Computer vision means it can "do something" with the recognized images.

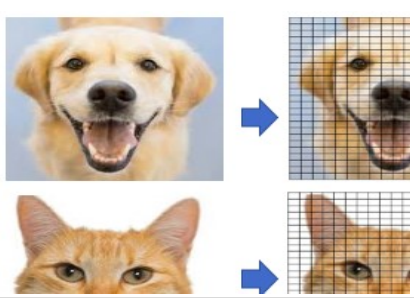
Just like the phrase "What-you-see-is-what-you-get" says, human brains make vision easy. It doesn't take any effort for humans to tell apart a dog, a cat or a flying saucer. But this process is quite hard for a computer to imitate: they only seem easy because God designs our brains incredibly good in recognizing images. A common example of image recognition is optical character recognition (OCR). A scanner can identify the characters in the image to convert the texts in an image to a text file. With the same process, OCR can be applied to recognize the text of a license plate in an image.

How does Image Recognition Works?

How do we train a computer to tell one image apart from another image? The process of an image recognition model is no different from the process of machine learning modeling. I list the modeling process for image recognition in Step 1 through 4.

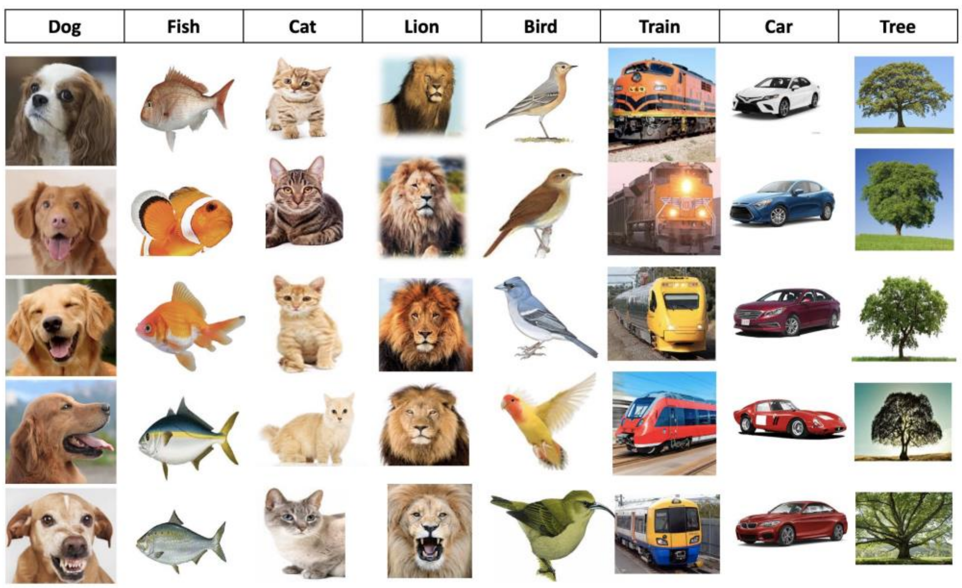
Step 1: Extract Pixel Features from an Image.

(Figire A)



extracted from the image. An image is actually made of “pixels”, as shown in Figure (A). Each pixel is represented by a number or a set of numbers — and the range of these numbers is called the color depth (or bit depth). In other words, the color depth indicates the maximum number of potential colors that can be used in an image. In an (8-bit) greyscale image (black and white) each pixel has one value that ranges from 0 to 255. Most images today use 24-bit color or higher. An RGB color image means the color in a pixel is the combination of red, green and blue. Each of the colors ranges from 0 to 255. This RGB color generator shows how any color can be generated by RGB. So a pixel contains a set of three values RGB (102, 255, 102) refers to color **#66ff66.** An image 800 pixel wide, 600 pixels high has 800 x 600 = 480,000 pixels = 0.48 megapixels (“megapixel” is 1 million pixels). An image with a resolution of 1024×768 is a grid with 1,024 columns and 768 rows, which therefore contains 1,024 × 768 = 0.78 megapixels.

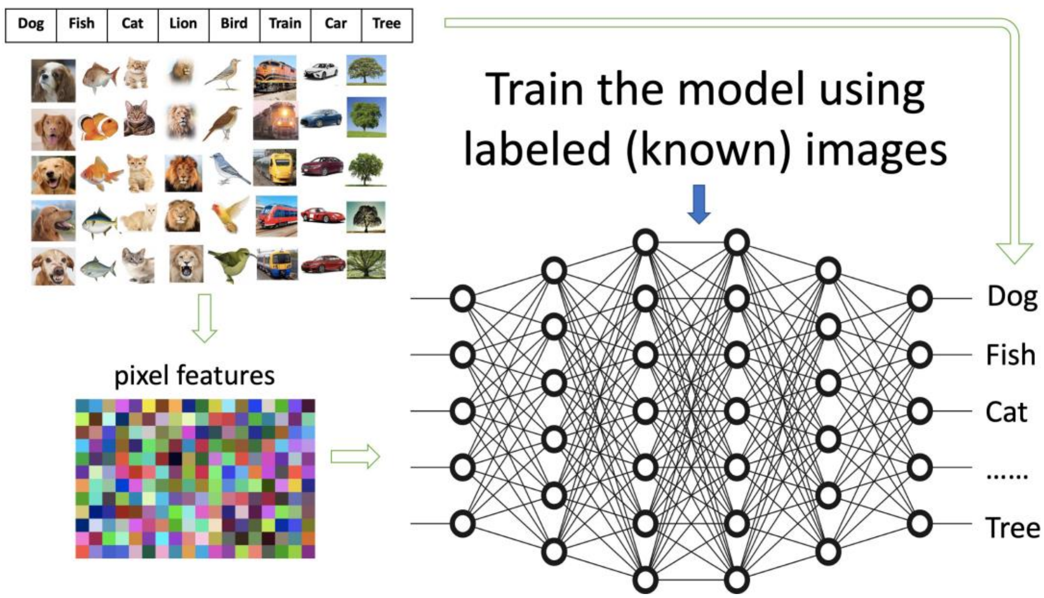
Step 2: Prepare labeled images to train the model.



(Figure B)

Once each image is converted to thousands of features, with the known labels of the images we can use them to train a model. Figure (B) shows many labeled images that belong to different categories such as “dog” or “fish”. The more images we can use for each category, the better a model can be trained to tell an image whether is a dog or a fish image. Here we already know the category that an image belongs to and we use them to train the model. This is called supervised machine learning.

Step 3: Train the model to be able to categorize images.

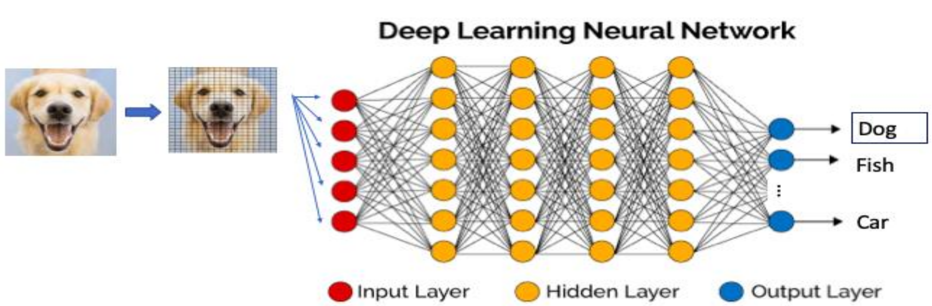


(Figure C)

Figure (C) demonstrates how a model is trained with the pre-labeled images. The huge *networks* in the middle can be considered as a giant filter. The images in their extracted forms enter the input side and the labels are in the output side. The purpose here is to train the networks such that an image with its features coming from the input will match the label in the right.

Step 4: Recognize (or predict) a new image to be one of the categories.

(Figure D)

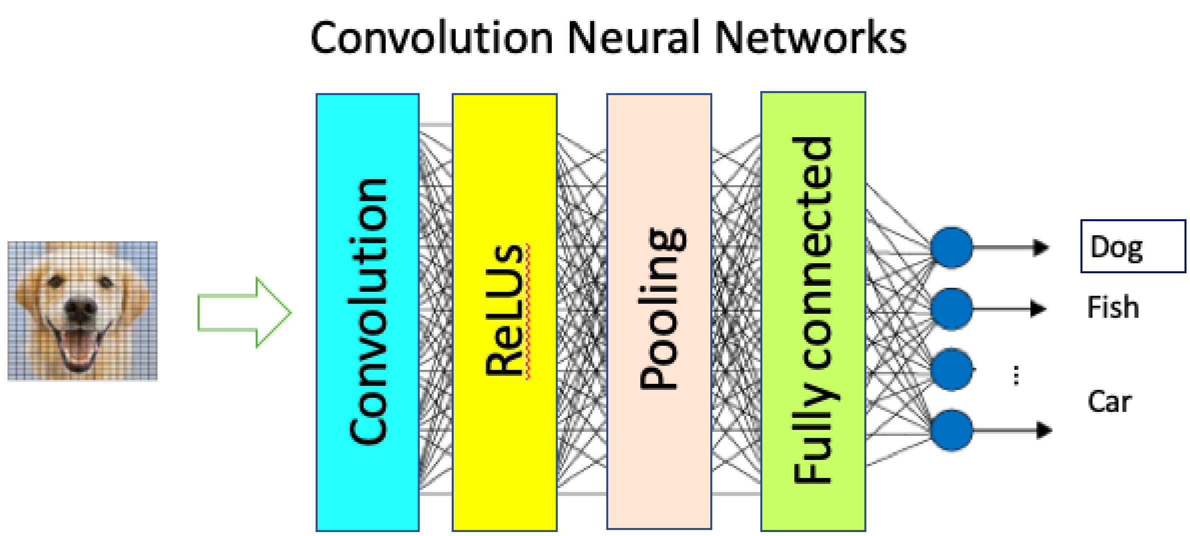


Once a model is trained, it can be used to recognize (or predict) an unknown image. Figure (D) shows a new image is recognized as a dog image. Notice that the new image will also go through the pixel feature extraction process.

Convolutional Neural Networks (the algorithm for Image-Recognition)

The networks in Figure (C) or (D) have implied the popular models are neural networks models. Convolutional Neural Networks (CNNs or ConvNets) have been widely applied in image classification, object detection or image recognition.

Explanation for Convolutional Neural Networks



(Image Representation for CNN)

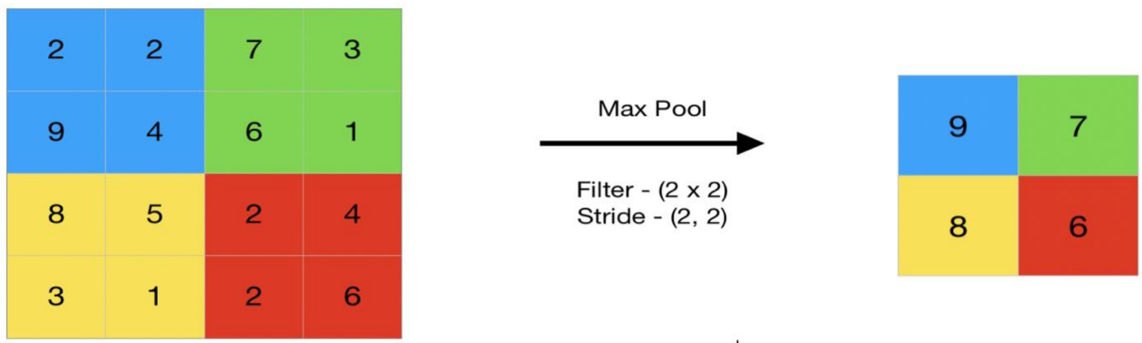
1. **Convolutional Layer**

   The first step that CNNs do is to create many small pieces called *features* like the 2x2 boxes. Each feature characterizes some shape about the original image. Let each feature scan through the original image. If there is a perfect match, there is a high score in that box. If there is a low match or no match, the score is low or zero. This process in producing the scores is called *filtering*. Each feature produces a *filtered* image with high scores and low scores when scanning through the original image. For example, the red box found four areas in the original image that show a perfect match with the feature, so scores are high for those four areas. The pink boxes are the areas that match to some extent. The act of trying every possible match by scanning through the original image is called *convolution*. The filtered images are stacked together to become the *convolution layer*.

2. **ReLUs Layers**

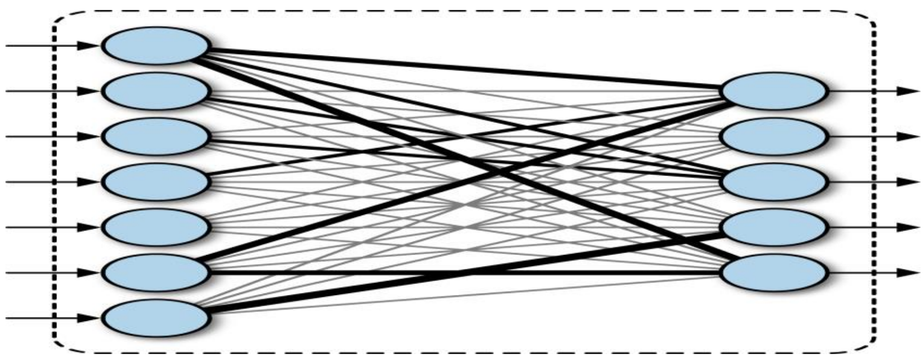
The Rectified Linear Unit (ReLU) is the step that is the same as the step in the typical neural networks. It rectifies any negative value to zero so as to guarantee the math will behave correctly.

3. **Max Pooling Layer**



Pooling shrinks the image size. A 2x2 window scans through each of the filtered images and assigns the max value of that 2x2 window to a 1x1 box in a new image. As the maximum value in the first 2x2 window is a high score, so the high score is assigned to the 1x1 box. The 2x2 box moves to the second window where there is a high score and a low score, so a high score is assigned to the 1x1 box. After pooling, a new stack of smaller filtered images is produced

4. **Fully Connected Layers**



Now we split the smaller filtered images and stack them into a single list. Each value in the single list predicts a probability for each of the final values 1,2,…, 0. This part is the same as the output layer in the typical neural networks.

What is the difference between CNN and typical NN's?

The typical neural networks stack the original image into a list and turn it to be the input layer. The information between neighboring pixels may not be retained. In contrast, CNNs construct the convolution layer that retains the information between neighboring pixels.

Model for Classifying Plant Disease:

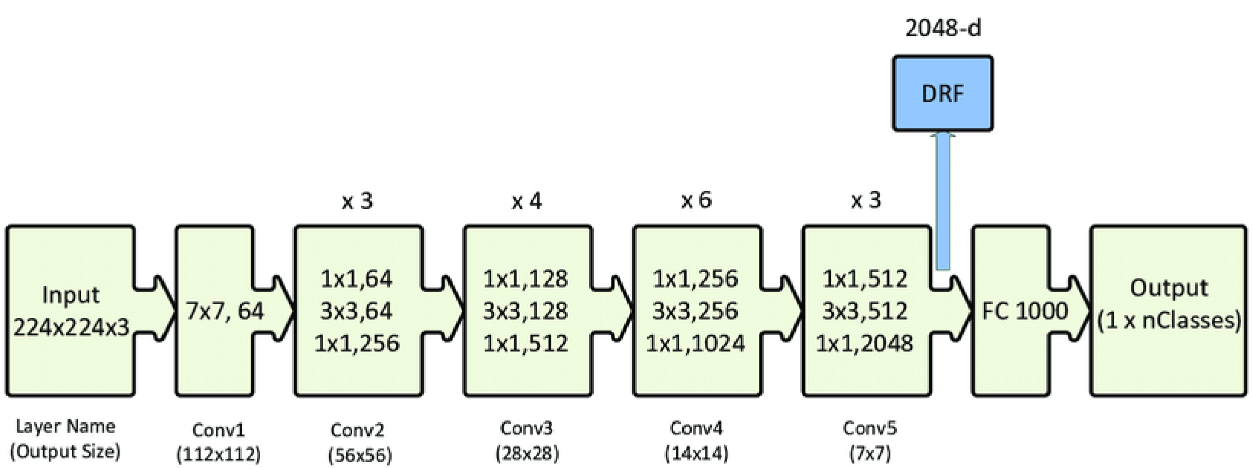
For the purpose of training the model to predict the diseased plant accurately, we have used VGG16 architecture in the form of Transfer-Learning.

What is Transfer Learning?

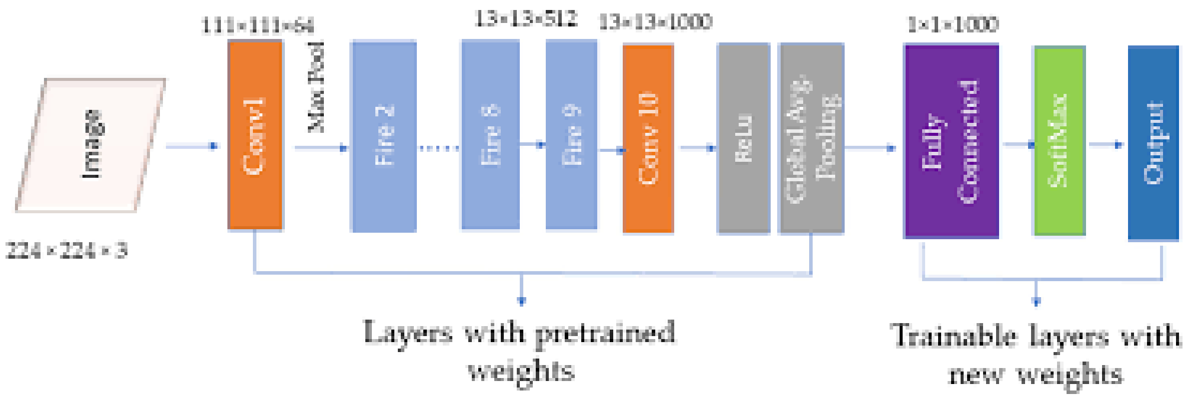
Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

ResNet50 Architecture:



(Block Diagram for ResNet50)



(VGG16 architecture)

The input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Max-pooling is performed over a 2×2 pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with the rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN), such normalization does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time.

Configuration:

The ConvNet configurations are outlined in figure 02. The nets are referred to their names (A-E). All configurations follow the generic design present in architecture and differ only in the depth: from 11 weight layers in the network A (8 conv. and 3 FC layers) to 19 weight layers in the network E (16 conv. and 3 FC layers). The width of conv. layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
conv2_x	56x56	3x3 max pool, stride 2				
		$\begin{bmatrix} 3\text{x}3, 64 \\ 3\text{x}3, 64 \end{bmatrix}$ x2	$\begin{bmatrix} 3\text{x}3, 64 \\ 3\text{x}3, 64 \end{bmatrix}$ x3	$\begin{bmatrix} 1\text{x}1, 64 \\ 3\text{x}3, 64 \\ 1\text{x}1, 256 \end{bmatrix}$ x3	$\begin{bmatrix} 1\text{x}1, 64 \\ 3\text{x}3, 64 \\ 1\text{x}1, 256 \end{bmatrix}$ x3	$\begin{bmatrix} 1\text{x}1, 64 \\ 3\text{x}3, 64 \\ 1\text{x}1, 256 \end{bmatrix}$ x3
conv3_x	28x28	$\begin{bmatrix} 3\text{x}3, 128 \\ 3\text{x}3, 128 \end{bmatrix}$ x2	$\begin{bmatrix} 3\text{x}3, 128 \\ 3\text{x}3, 128 \end{bmatrix}$ x4	$\begin{bmatrix} 1\text{x}1, 128 \\ 3\text{x}3, 128 \\ 1\text{x}1, 512 \end{bmatrix}$ x4	$\begin{bmatrix} 1\text{x}1, 128 \\ 3\text{x}3, 128 \\ 1\text{x}1, 512 \end{bmatrix}$ x4	$\begin{bmatrix} 1\text{x}1, 128 \\ 3\text{x}3, 128 \\ 1\text{x}1, 512 \end{bmatrix}$ x8
conv4_x	14x14	$\begin{bmatrix} 3\text{x}3, 256 \\ 3\text{x}3, 256 \end{bmatrix}$ x2	$\begin{bmatrix} 3\text{x}3, 256 \\ 3\text{x}3, 256 \end{bmatrix}$ x6	$\begin{bmatrix} 1\text{x}1, 256 \\ 3\text{x}3, 256 \\ 1\text{x}1, 1024 \end{bmatrix}$ x6	$\begin{bmatrix} 1\text{x}1, 256 \\ 3\text{x}3, 256 \\ 1\text{x}1, 1024 \end{bmatrix}$ x23	$\begin{bmatrix} 1\text{x}1, 256 \\ 3\text{x}3, 256 \\ 1\text{x}1, 1024 \end{bmatrix}$ x36
conv5_x	7x7	$\begin{bmatrix} 3\text{x}3, 512 \\ 3\text{x}3, 512 \end{bmatrix}$ x2	$\begin{bmatrix} 3\text{x}3, 512 \\ 3\text{x}3, 512 \end{bmatrix}$ x3	$\begin{bmatrix} 1\text{x}1, 512 \\ 3\text{x}3, 512 \\ 1\text{x}1, 2048 \end{bmatrix}$ x3	$\begin{bmatrix} 1\text{x}1, 512 \\ 3\text{x}3, 512 \\ 1\text{x}1, 2048 \end{bmatrix}$ x3	$\begin{bmatrix} 1\text{x}1, 512 \\ 3\text{x}3, 512 \\ 1\text{x}1, 2048 \end{bmatrix}$ x3
	1x1	average pool, 1000-d fc, softmax				
FLOPs		$1.8\text{x}10^9$	$3.6\text{x}10^9$	$3.8\text{x}10^9$	$7.6\text{x}10^9$	$11.3\text{x}10^9$

Due to its depth and number of fully-connected nodes, VGG16 is over 533MB. This makes deploying VGG a tiresome task.VGG16 is used in many deep learning image classification problems; however, smaller network architectures are often more desirable (such as SqueezeNet, GoogLeNet, etc.). But it is a great building block for learning purpose as it is easy to implement.

VGG16 significantly outperforms the previous generation of models in the ILSVRC-2012 and ILSVRC-2013 competitions. The VGG16 result is also competing for the classification task winner (GoogLeNet with 6.7% error) and substantially outperforms the ILSVRC-2013 winning submission Clarifai, which achieved 11.2% with external training data and 11.7% without it. Concerning the single-net performance, VGG16 architecture achieves the best result (7.0% test error), outperforming a single GoogLeNet by 0.9%.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

It was demonstrated that the representation depth is beneficial for the classification accuracy, and that state-of-the-art performance on the ImageNet challenge dataset can be achieved using a conventional ConvNet architecture with substantially increased depth.

Dataset:

For the purpose of this project we have used Open-Source dataset, comprising of more than 20 thousand images. It consists of 15 different plant species. Which are further divided into train and test split.

Implementation Details:

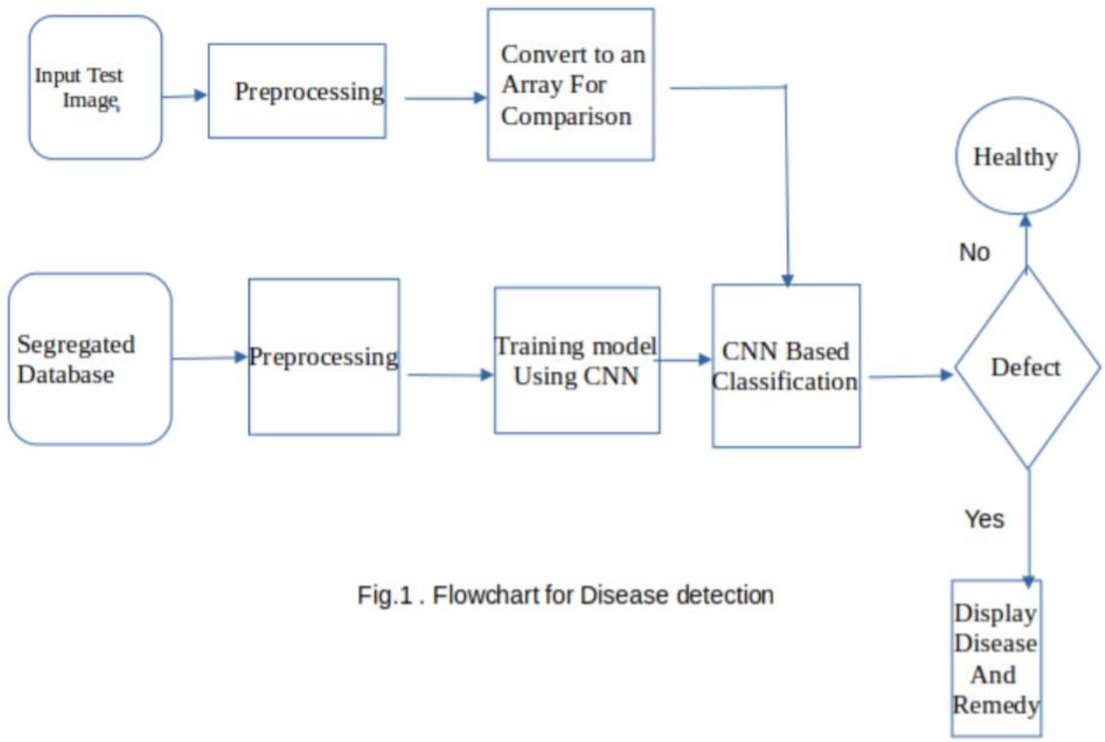


Fig.1 . Flowchart for Disease detection

The project will be implemented in Python3, Tensorflow and Keras.

<u>Algorithm for Disease Detection:</u>

1) Preprocessing the Images from dataset.

2) Training a CNN Model for Image Classification.

3) Saving the model.

4) Scanning diseased crop.

5) Preprocessing.

6) Converting it into Array.

7) Comparing it with the Saved Model

8) Giving the Result as Healthy or Diseased.

B) <u>Crop Prediction:</u>

Crop Prediction further has two important characteristics for accurate predictions. They are

a) Weather Prediction (Rainfall Prediction)

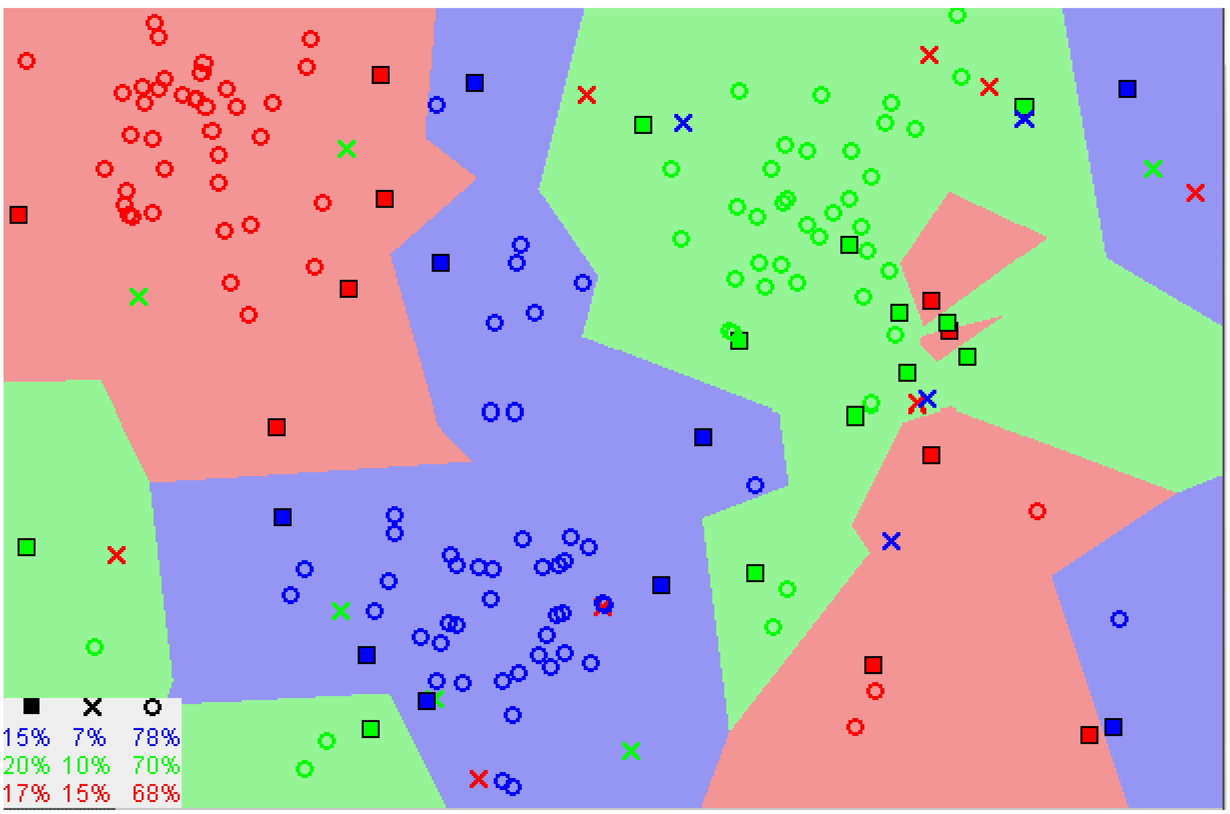
b) Crop Prediction based on Season

<u>Weather (Rainfall) prediction using KNN :</u>

KNN Algorithm:

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.



(<u>Image showing how similar data points typically exist close to each other</u>)

In the image above that most of the time, similar data points are close to each other. The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph.

The KNN Algorithm:

1) Load the data.
2) Initialize K to your chosen number of neighbors.
3) For each example in data.

   3.1) Calculate the distance between the query example and the current example from the data.

   3.2) Add the distance and the index of the example to an ordered collection.

4) Sort the ordered collection of distances and indices from smallest to largest by the distances.
5) Pick the first K entries from the sorted collection.
6) Get the labels of the selected K entries.
7) If regression, return the mean of K labels.
8) If classification, return the mode of K labels.

Choosing the right value of K:

To select the K that's right for your data, we run the KNN algorithm several times with different values of K and choose the K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before.

Here are some things to keep in mind:

1. As we decrease the value of K to 1, our predictions become less stable. Just think for a minute, imagine K=1 and we have a query point surrounded by several reds and one

green. Reasonably, we would think the query point is most likely red, but because K=1, KNN incorrectly predicts that the query point is green.

2. Inversely, as we increase the value of K, our predictions become more stable due to majority voting / averaging, and thus, more likely to make more accurate predictions (up to a certain point). Eventually, we begin to witness an increasing number of errors. It is at this point we know we have pushed the value of K too far.

3. In cases where we are taking a majority vote (e.g. picking the mode in a classification problem) among labels, we usually make K an odd number to have a tiebreaker.

Advantages:

1. The algorithm is simple and easy to implement.
2. There's no need to build a model, tune several parameters, or make additional assumptions.
3. The algorithm is versatile. It can be used for classification, regression, and search (as we will see in the next section).

Disadvantages:

1. The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

Summary:

The k-nearest neighbours (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows.

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

In the case of classification and regression, we saw that choosing the right K for our data is done by trying several Ks and picking the one that works best.

Crop Prediction Using Decision Trees:

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

Decision Tree Algorithm:

1) Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving **regression and classification problems** too.

2) The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data (training data).

3) In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Types of Decision Tree:

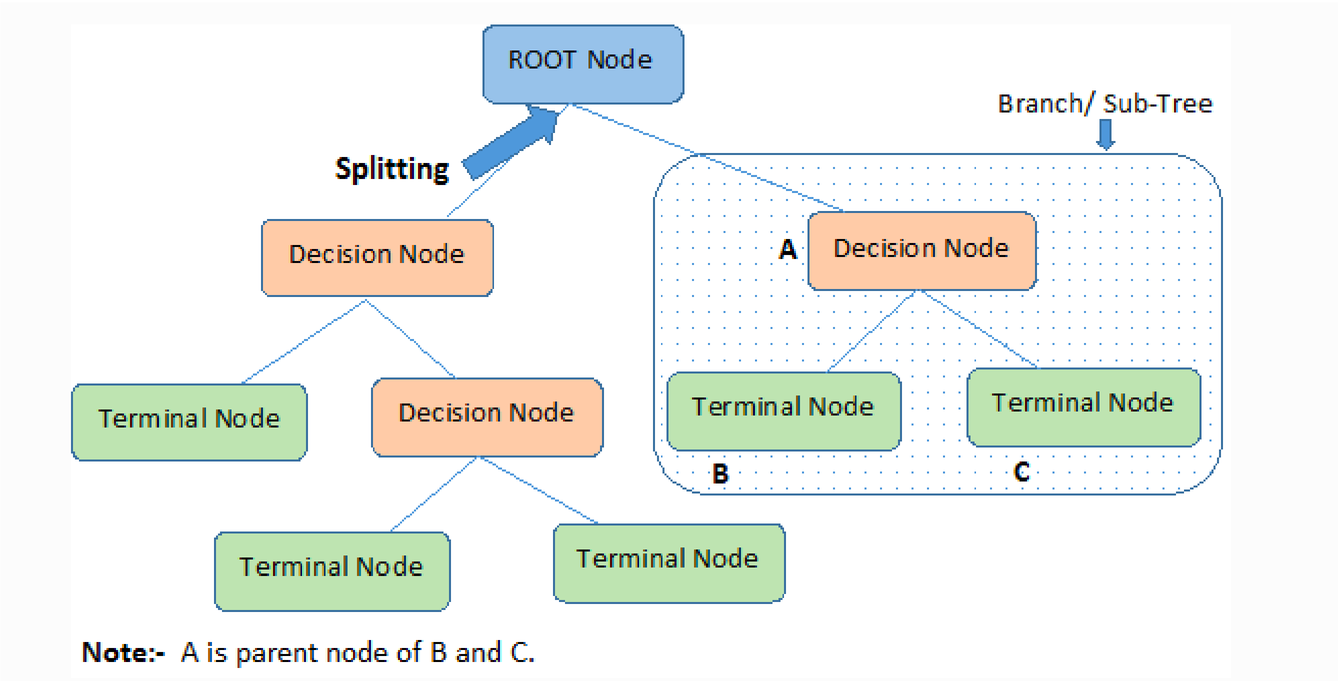
Types of decision trees are based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** Decision Tree which has a categorical target variable then it called a **Categorical variable decision tree.**
2. **Continuous Variable Decision Tree:** Decision Tree has a continuous target variable then it is called **Continuous Variable Decision Tree.**

Important Terminology related to Decision Tress:

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.

4. **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
6. **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.
7. **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.



Assumptions while creating Decision Tree:

- In the beginning, the whole training set is considered as the **root.**
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

How do Decision Trees Work?

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

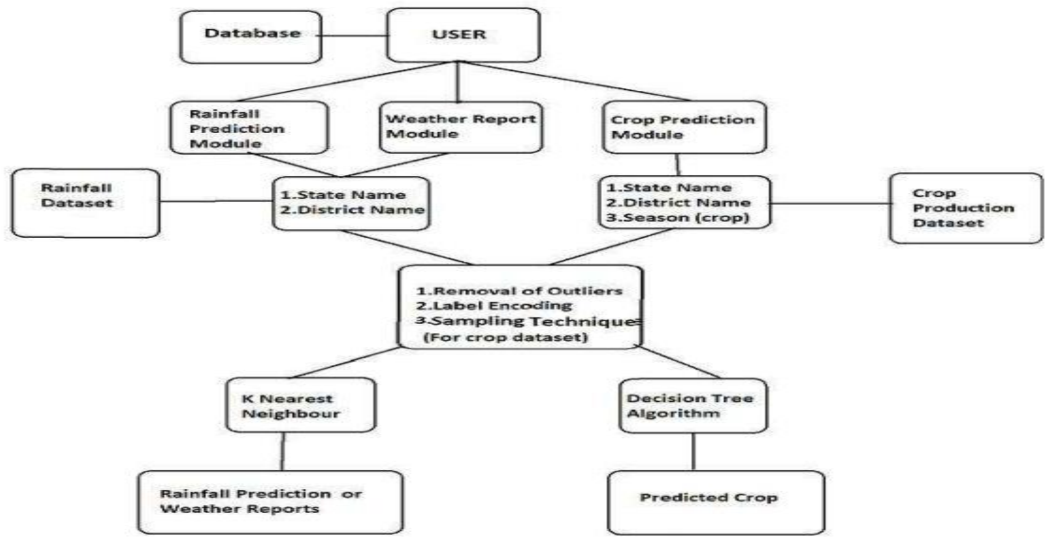
The algorithm selection is also based on the type of target variables. Let us look at some algorithms used in Decision Trees:

ID3 → (extension of D3)
C4.5 → (successor of ID3)
CART → (Classification And Regression Tree)
CHAID → (Chi-square automatic interaction detection Performs multi-level splits when computing classification trees)
MARS → (multivariate adaptive regression splines)

Steps in ID3:

1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates **Entropy(H)** and **Information gain (IG)** of this attribute.
3. It then selects the attribute which has the smallest Entropy or Largest Information gain.
4. The set S is then split by the selected attribute to produce a subset of the data.
5. The algorithm continues to recur on each subset, considering only attributes never selected before.

Implementation Details:



The project will be implemented using Python, Scikit Learn.

Algorithm:

1) Dataset for Rainfall and Crop Production gathering.

2) Dataset is pre-processed for removal of outliers and handling missing values.

3) KNN is applied on Weather dataset for getting prediction for Rainfall.

4) Decision Tree based approach is used on Crop Production Dataset for prediction of Crop.

5) After matching it with government requirements, Crop is predicted.

3.2Hardware & Software:

Software:

1. Android Studio (IDE for development of Apps.)

2. VS Code (Code Editor)

3. Anaconda 3.8 (For creating Virtual-env and Jupyter Notebook)

4. Google Colab (Platform as a Service for developing and building models)

5. Android/IOS OS (For using Application.)

6. Linux OS (To develop algorithm and app.)

Frameworks:

1. Tensorflow (Python library to build Deep-Learning models.)

2. Python 3.8 (For development of Algorithms.)

3. Bootstrap (For building the Web-App.)

4. Keras (For building CNN Models.)

Hardware:

1. Camera Module (For capturing images of plants.)

2. TESLA K80 GPU (For training CNN models.)

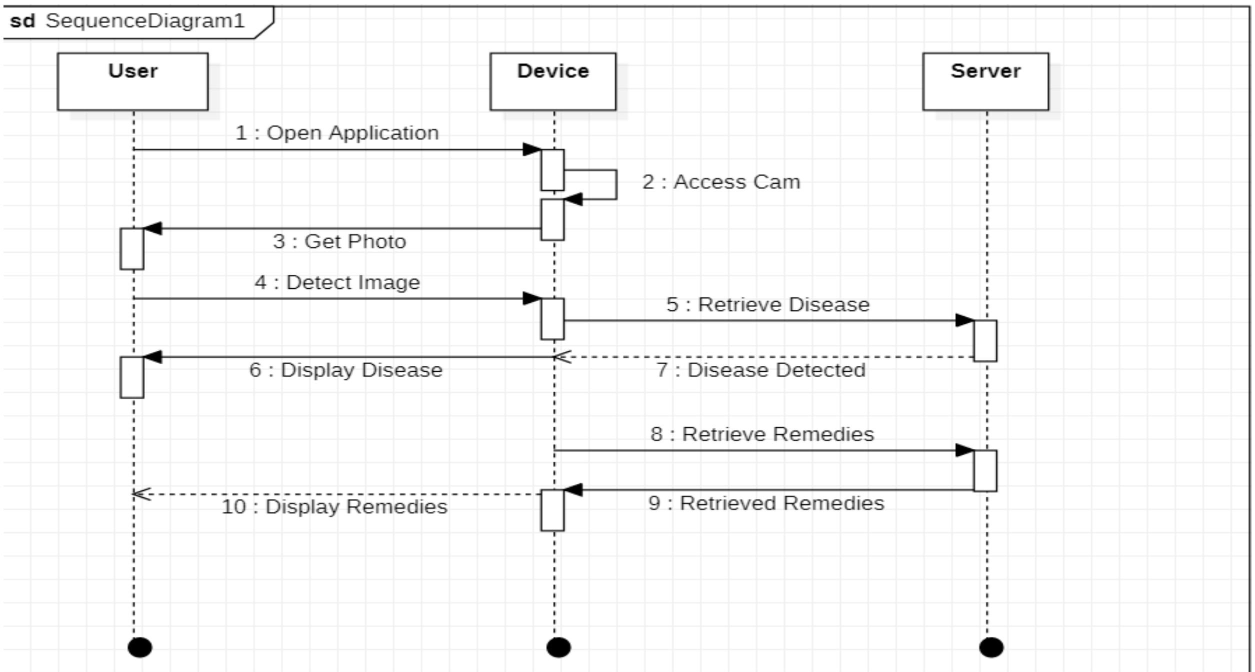
3. Smartphone/Computer (For using Web-App.)

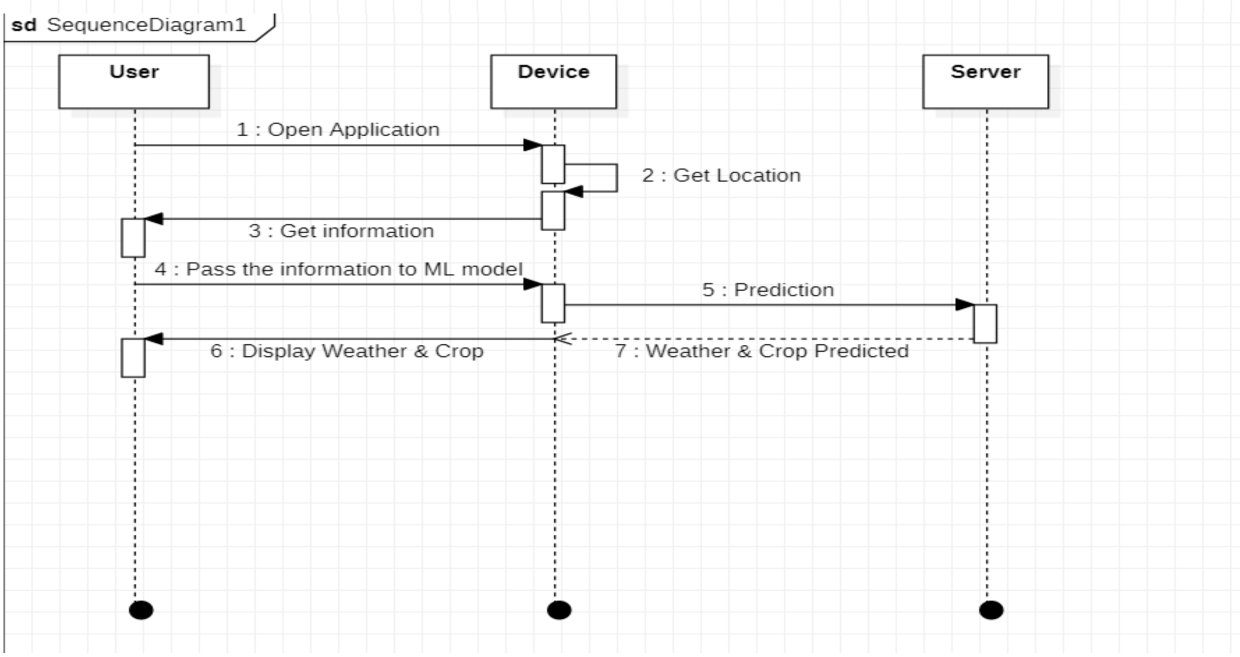
3.3 Design Details:

1) Sequence Diagrams:

a) Sequence diagram for Disease Prediction:

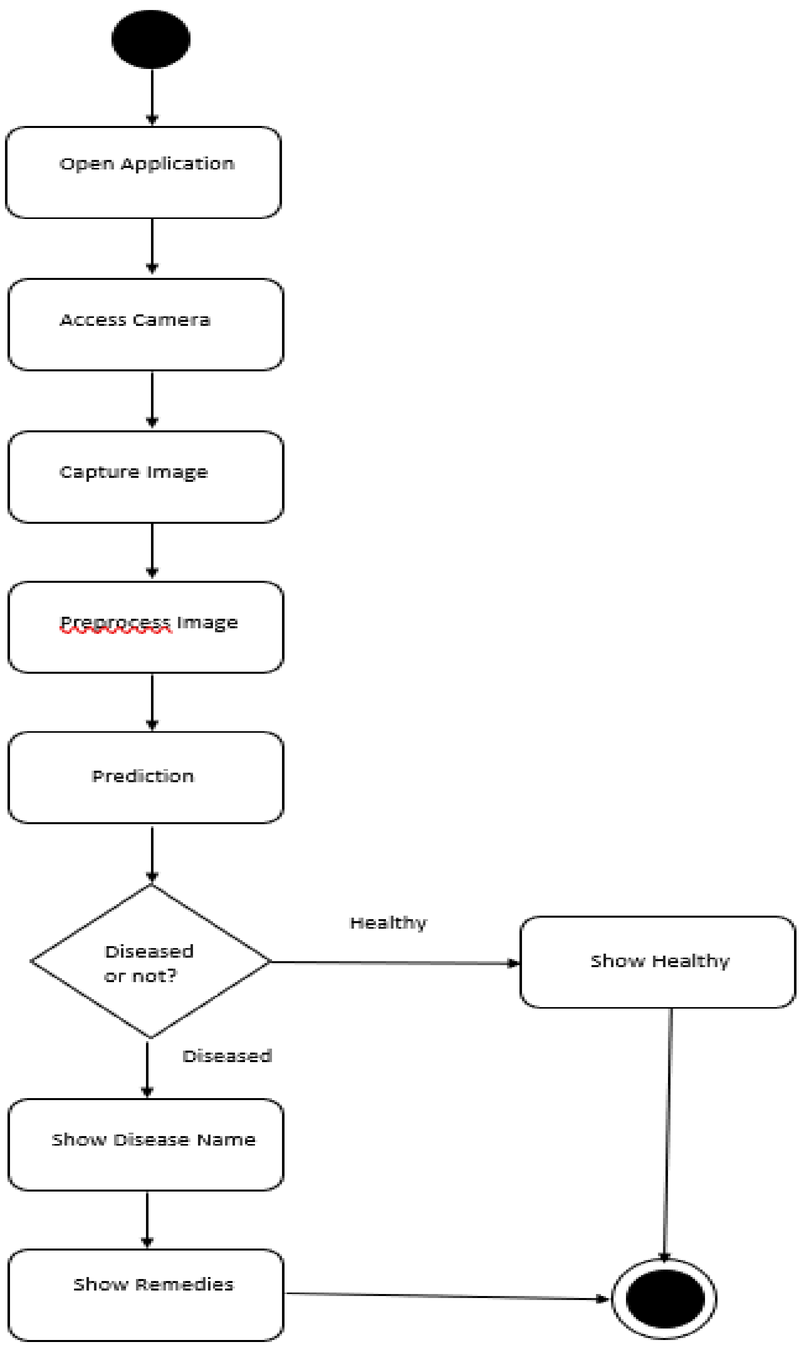
b) Sequence Diagram for Weather and Crop Prediction:



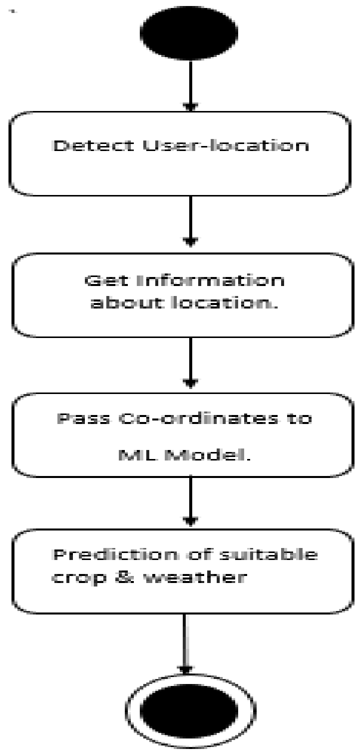


2) Activity Diagrams:

a) Activity Diagram for Disease Prediction:

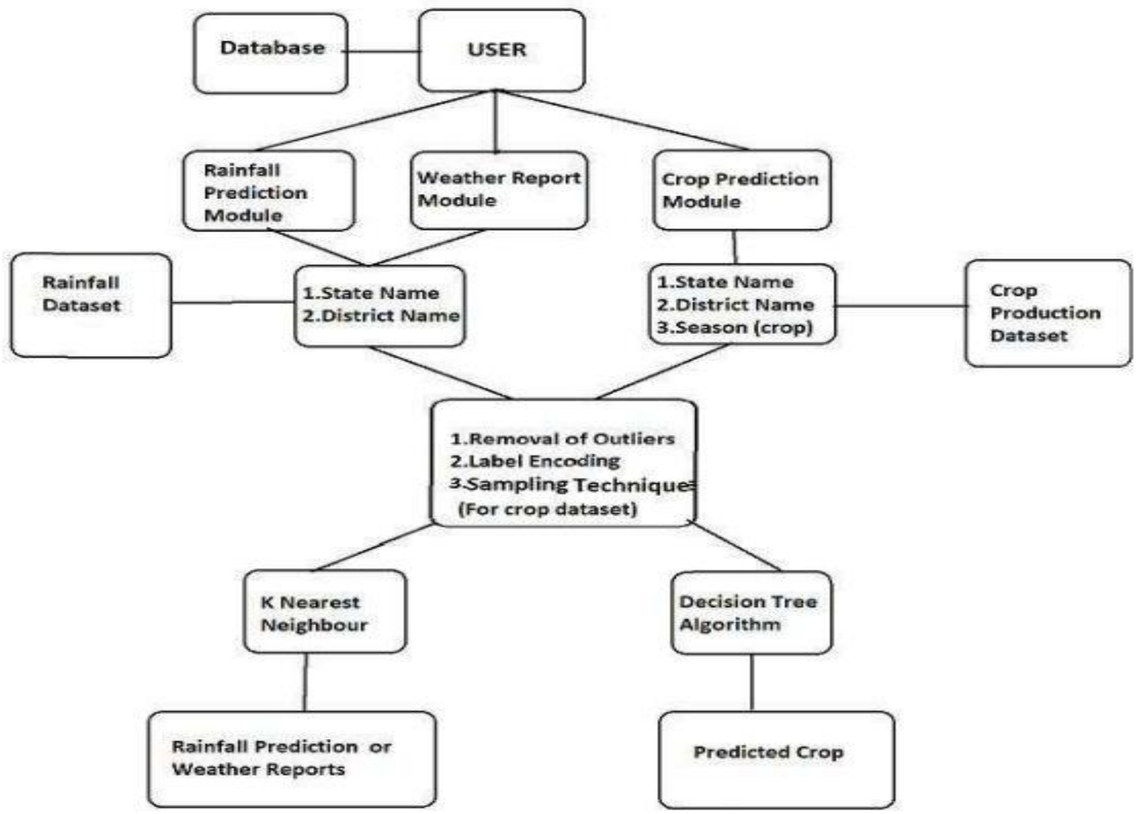


b) Activity Diagram for Weather and Crop Prediction:



3.4 Methodology:

1) Crop Production:



(Flowchart for Crop Prediction)

Algorithm:

1) Dataset for Rainfall and Crop Production gathering.

2) Dataset is pre-processed for removal of outliers and handling missing values.

3) KNN is applied on Weather dataset for getting prediction for Rainfall.

4) Decision Tree based approach is used on Crop Production Dataset for prediction of Crop.

5) After matching it with government requirements, Crop is predicted.

2) Disease Prediction:

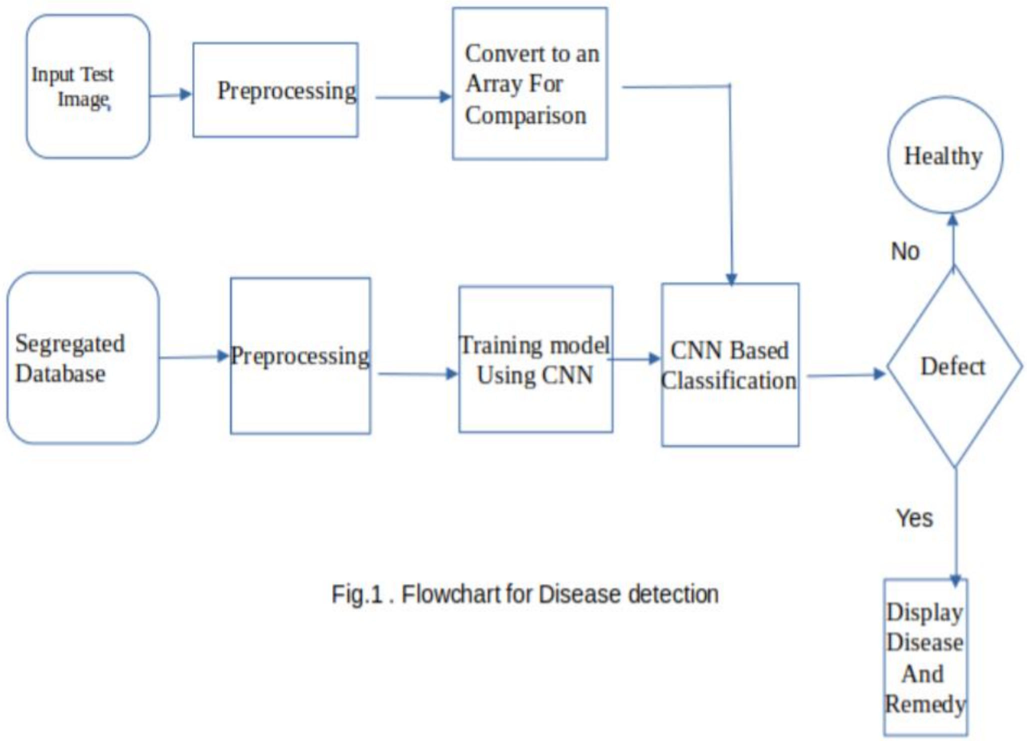


Fig.1 . Flowchart for Disease detection

Algorithm:

1) Preprocessing the images from the dataset.

2) Training the CNN Model for Image Classification.

3) Saving the model.

4) Scanning the diseased crop and capturing the image.

5) Preprocessing the Captured Image.

6) Passing the processed image to the saved Model and thereby classifying it.

7) Getting the result as Healthy or Diseased Plant.

3) Remedy Prediction:

1) Once the result is obtained by Classification.

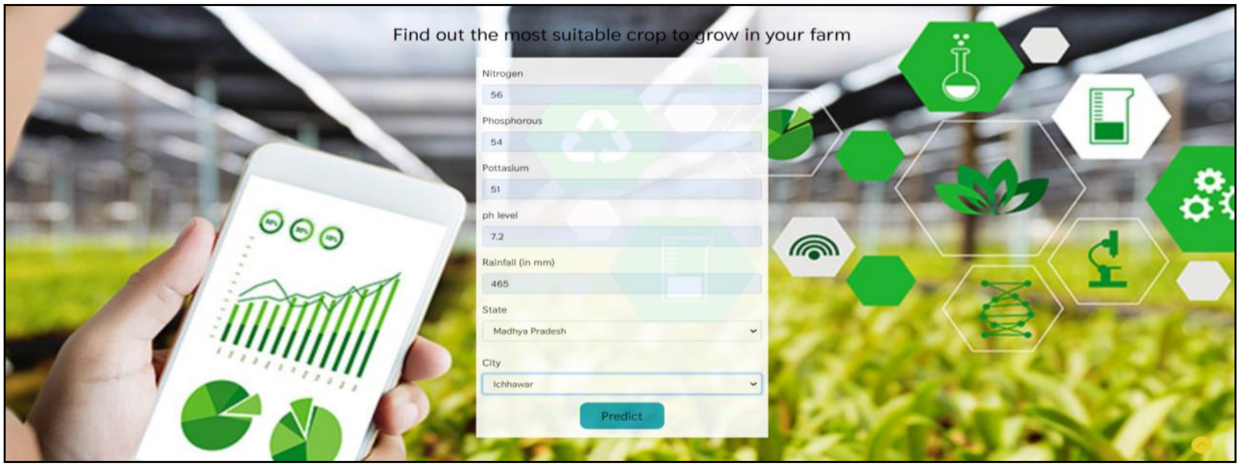
2) If the result is diseased crop, then the Disease as well crop species name is recorded.

3) Further this recorded-values are matched in central database and the associated remedy is displayed.

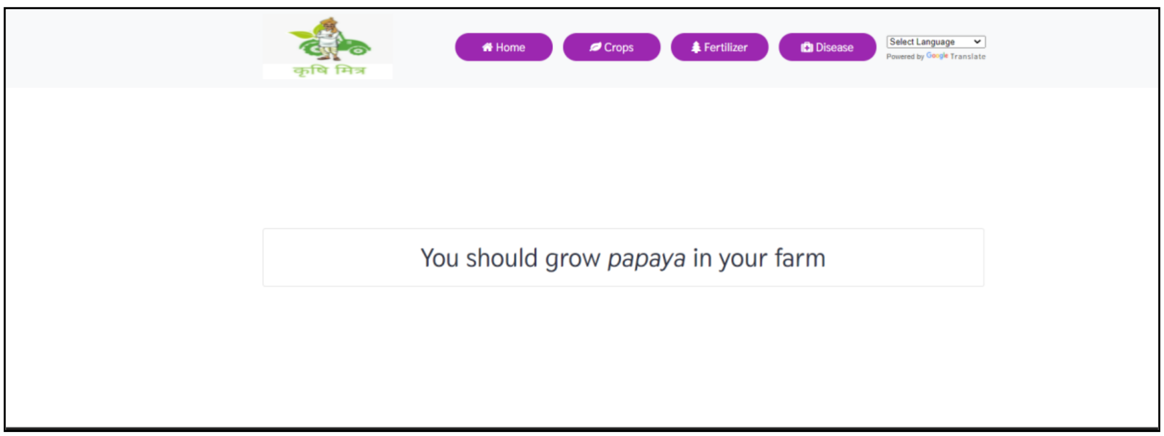
4) Option for buying generic pesticides/fertilizer displayed.

4. Result Analysis

1) Crop Prediction:

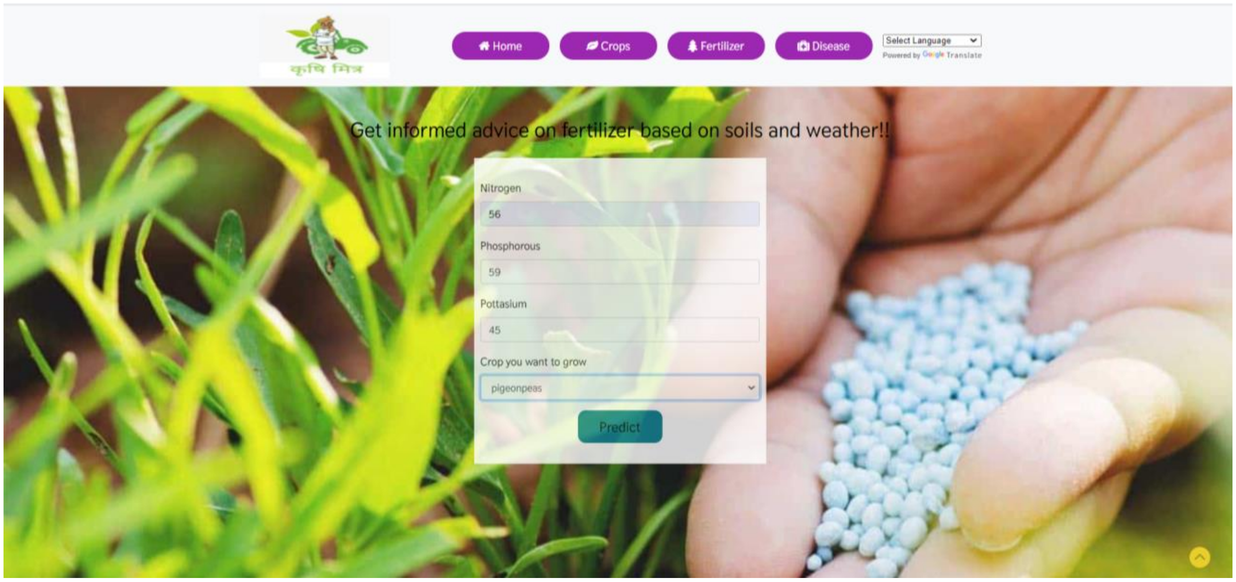


(Crop Prediction Page)

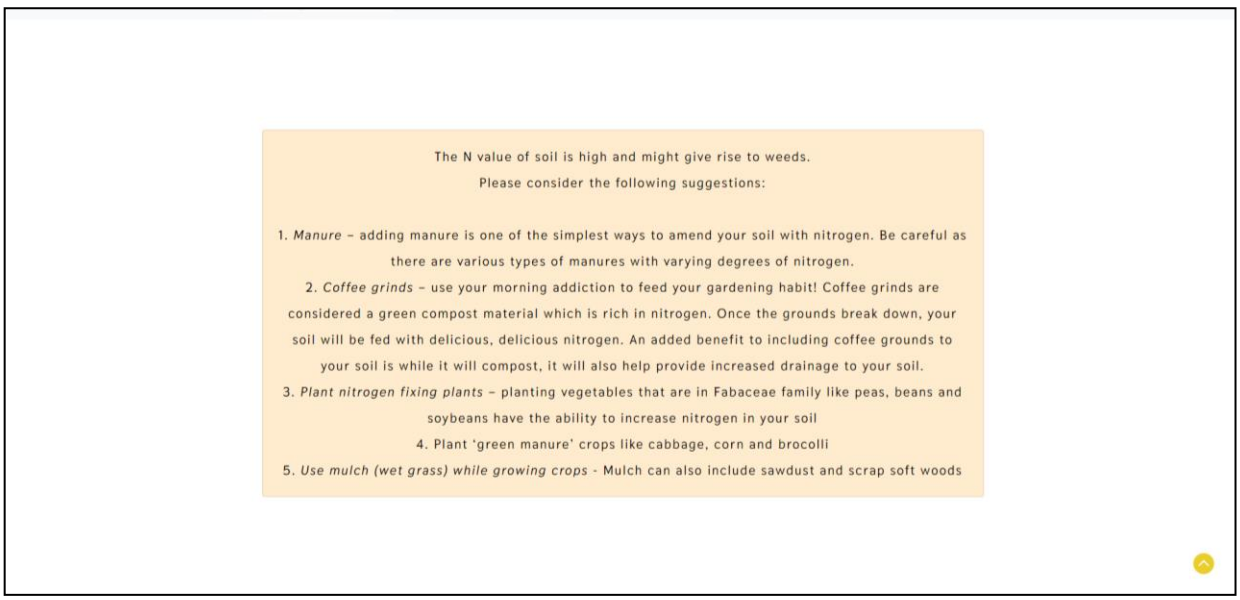


(Crop Prediction Result)

2) Fertilizer Recommendation:

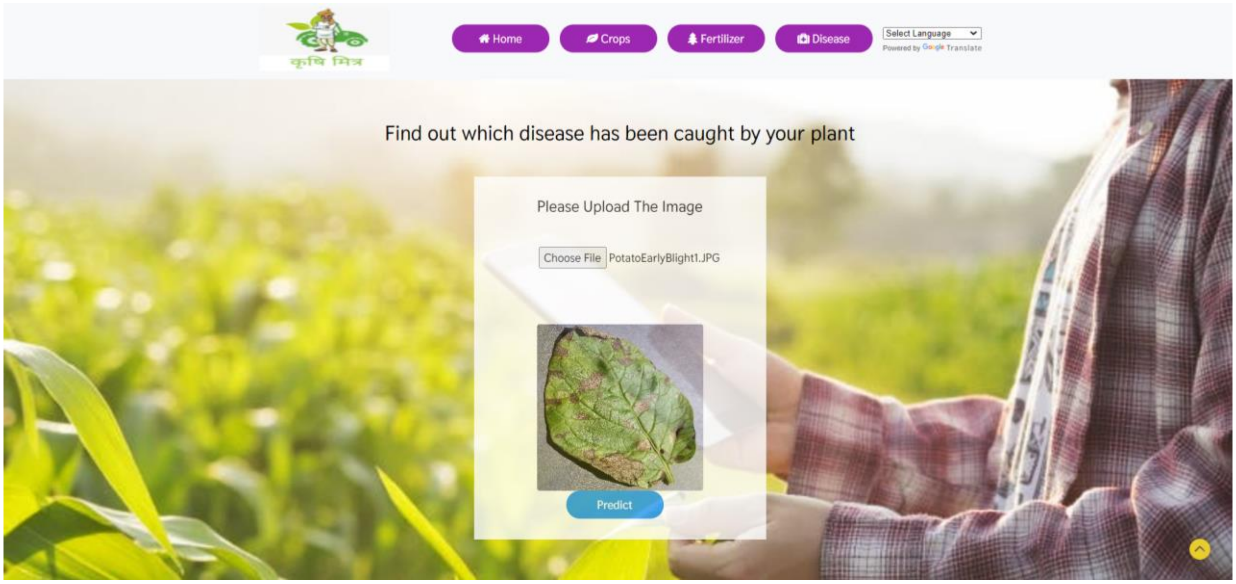


(Fertilizer Prediction Page)

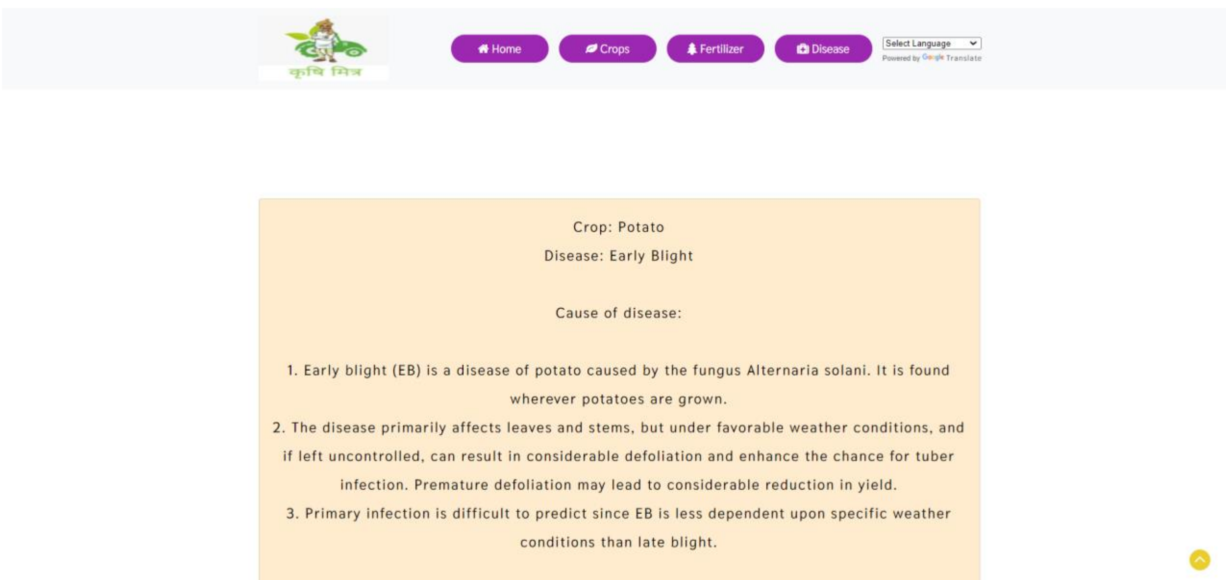


(Fertilizer Prediction Result)

3)Disease Prediction:



(Disease Prediction Page)



(Disease Prediction Page)

4) Remedy Prediction

How to prevent/cure the disease

1. Plant only diseasefree, certified seed.
2. Follow a complete and regular foliar fungicide spray program.
3. Practice good killing techniques to lessen tuber infections.
4. Allow tubers to mature before digging, dig when vines are dry, not wet, and avoid excessive wounding of potatoes during harvesting and handling.

(Remedy Recommendation)

5. Conclusion

E-Governance represents a paradigm shift in the field of agricultural reforms. Bringing it about will be a perilous journey, but it will be a journey for the good, a journey for the best. Our application uses cutting-edge technology in the fields of image recognition and deep learning. Using our application, farmers can learn more about location specific diseases towards crops and the best ways to combat these diseases. This will result in a reduction in terms of crop loss, better growth, optimum soil quality, and increased revenue for the farmers, which, in turn, will obviously result in the revenue growth for the government of our esteemed country.

5. References

1. Hands-On Machine Learning with Scikit-Learn and TensorFlow.

2. Udemy- Projects in Machine Learning: Beginner to Professional.

3. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images.

4. Crop Prediction System using Machine Learning Algorithms by Pavan Patil, Virendra Panpatil, Shrikant Kokate.

5. Plant Disease Detection using CNN & Remedy by Adnan Mushtaq Ali Karol, Drushti Gulhane, Tejal Chandiwade

6. Coursera- Convolutional Neural Network in Tensorflow.

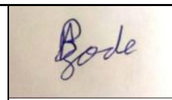
7. Striving for Simplicity : The All Convolution Net by Department of Computer Science, University of Freidburg

SCHEDULE FOR MAJOR PROJECT – I

Date	Week	Sign of the Student			Remark	Guide Sign
24/7/2020	1					
31/7/2020	2					
07/8/2020	3					
14/8/2020	4					
21/8/2020	5					
28/8/2020	6					
4/9/2020	7					
11/9/2020	8					
18/9/2020	9					
03/10/2020	10					
09/10/2020	11					
30/10/2020	12					
6/11/2020	13					
13/11/2020	14					

SCHEDULE FOR MAJOR PROJECT – II

Date	Week	Sign of the Student			Remark	Guide Sign
29/1/2021	1	Bode				
5/2/2021	2	Bode				
12/2/2021	3	Bode				
19/2/2021	4	Bode				
26/2/2021	5	Bode				
5/3/2021	6	Bode				
12/3/2021	7	Bode				
19/3/2021	8	Bode				
26/3/2021	9	Bode				
9/4/2021	10	Bode				
16/4/2021	11	Bode				
23/4/2021	12	Bode				

30/4/2021	13					
8/5/2021	14