Name of the project

# Mini Agriculturist

Submitted by:

Group -12, Level-3 Semester-1, Dept. of CSE, HSTU

Farah Sultana Fera, id - 1702078

Md Rakibul Hasan, id - 1702008

Md Touhid Islam, id - 1702040

Supervisor:

Md Rashedul Islam

Assistant professor, Dept. of CSE, HSTU

## Hajee Mohammad Danesh Science and Technology University, Dinajpur

## Motivations for the projects

What make us interested to build this project? Or what was our motivation?

Many of us in the city do gardening, on the patio, on the roof or in the balcony. There are many types of trees in our garden. Since plants are organisms, sometimes they suffer from many diseases. For the reason we don't pay attention or don't know or can't diagnose the disease properly and don't take the necessary steps, the trees die of the disease. As a result we may lose many favourite, expensive, exclusive plants.

Basically the role of our project is as a plant doctor. Where we can easily diagnose the disease from the leaves of the tree, learn more about the disease and what steps to take in case of any plant disease.

## Objectives of the project

The identification and detection of diseases of plants is one of the main points which determine the loss of the yield of crop production and agriculture. The studies of plant disease are the study of any visible points in any part of the plant which helps us differentiate between two plants, technically any spots or colour shades. Hence, image processing is used for the detection of plant diseases. The Detection of diseases follows the methods of image acquisition, image extraction, image segmentation, and image pre-processing. In this project we will show the detection of diseases of plants by getting their images of leaves, stems and fruits, and we will also show what steps can be taken to cure the disease.

## Some observation terminologies

### Machine learning:

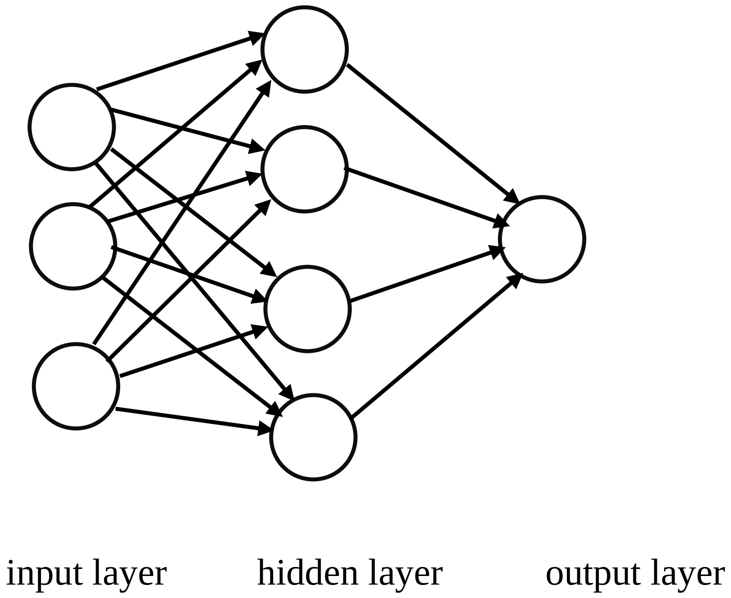
Quite simply, machine learning allows computers to ‘learn’. Traditionally, we always got computers to do things by providing a strict set of instructions. Machine Learning uses a very different approach. Instead of giving the computer a set of instructions on how to do something, we give it instructions on how to learn to do something. For example: think of a system that can classify pictures of animals as ‘cat’, ‘dog’, or ‘mouse’. Instead of manually finding unique characteristics from images of those animals and then coding it up, machine learning takes in images of those animals and finds characteristics and differences by itself. This process of teaching the computer is referred to as training.

## Deep learning:

Deep learning is a Technique for implementing Machine Learning or we can say Deep Learning is a category of machine learning models that use multi-layer neural networks. It uses neural networks to learn, sometimes, using decision trees may also be referred to as deep learning, but for the most part deep learning involves the use of neural networks.

## Neural Network:

Neural networks are computing systems with interconnected nodes that work much like neurons in the human brain. Using algorithms, they can recognize hidden patterns and correlations in raw data, cluster and classify it. There are three parts in a neural network.



i) Input layer
   - Actual values from dataset.

ii) Hidden layer
   - These are layers in between inputs and outputs.
   - If we have three or more layers then it will be Deep Neural Network.

iii) Output layer
   - It contains the final output

Neural Network pass an input, like an image, through multiple layers of digital neurons. Each layer reveals additional features of the input.

# Requirements for this project

## Non-Functional requirement

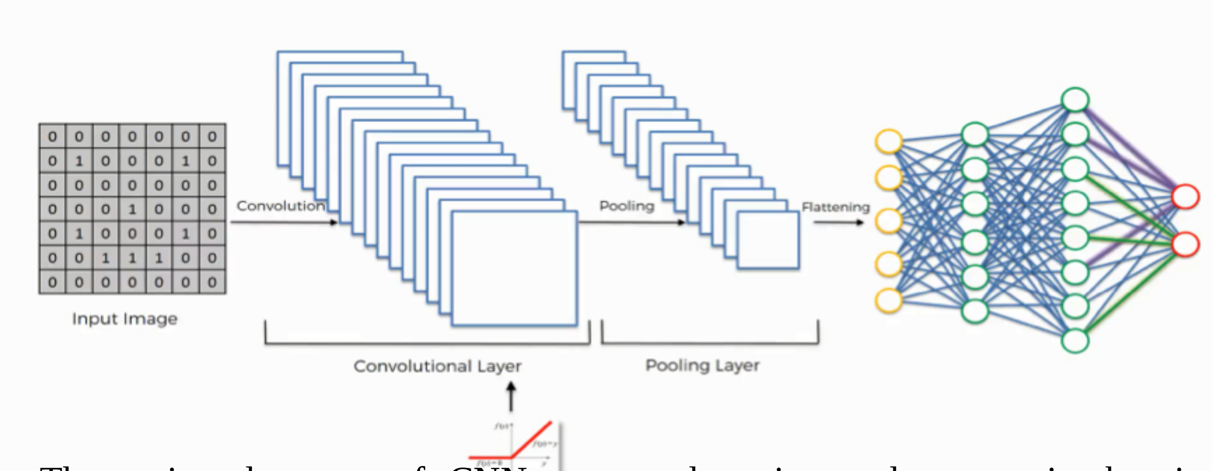
1. **Image Dataset:** We analyse 67,905 images of plant leaves, which have a spread of 30 class labels assigned to them. Each class label is a crop-disease pair, and we make an attempt to predict the crop-disease pair given just the image of the plant leaf. Figure 1 shows one example each from every crop-disease pair from the Plant-Village dataset. In all the approaches described in this paper, we resize the images to 256 × 256 pixels, and we perform both the model optimization and predictions on these downscaled images.

2. **Hardware Components:** 1) a computer with windows 10 or Linux os, python IDE Pycharm, VS code or any other text editor. 2) Internal or External camera.

3. **Keras:** Keras is a high-level Python neural networks library that runs on top of either TensorFlow or Theano. There are other high level Python neural networks libraries that can be used on top of TensorFlow, such as TF-Slim, although these are less developed simplifies the codes used in tensorflow by making use of a smaller code base so that the code length will reduce and make sure that the processing will run smoothly. Keras is used for a graphical representation of the models which helps to understand the structure of the model. Auto Keras, a library based on keras, has also gained popularity and can be used to make it quicker to get results.

4. **Tensorflow:** TensorFlow is a framework created by Google for creating Deep Learning models. Deep Learning is a category of machine learning models (=algorithms) that use multi-layer neural networks. Machine Learning has enabled us to build complex applications with great accuracy. Whether it has to do with images, videos, text or even audio, Machine Learning can solve problems from a wide range. Tensorflow can be used to achieve all of these applications.

5. **CNN:** CNN is one of the most popular deep learning algorithms**.** Convolutional Neural Networks are a complex neural network chain which work to get the features of an image from a dataset which is trained and classify them to get the required output. It trains the neural networks by using the dataset images and changing them to numerical values.

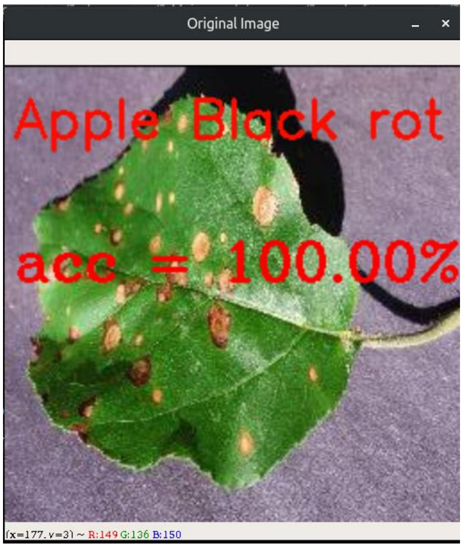


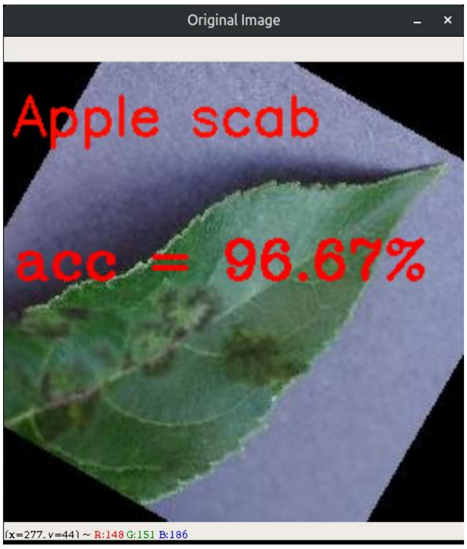
| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. ConvNets are more powerful than machine learning algorithms and are also computationally efficient. These numerical values are then put into numerical arrays based on their categorized characteristics. These arrays are then put into different nodes in the network and passed through multiple iterations based on the input given. The CNN models are used for geographical classification in multiple companies which require data to be classified in a quick and secure way it almost acts like a filter removing dust and separates the features of the images.

## Functional requirement

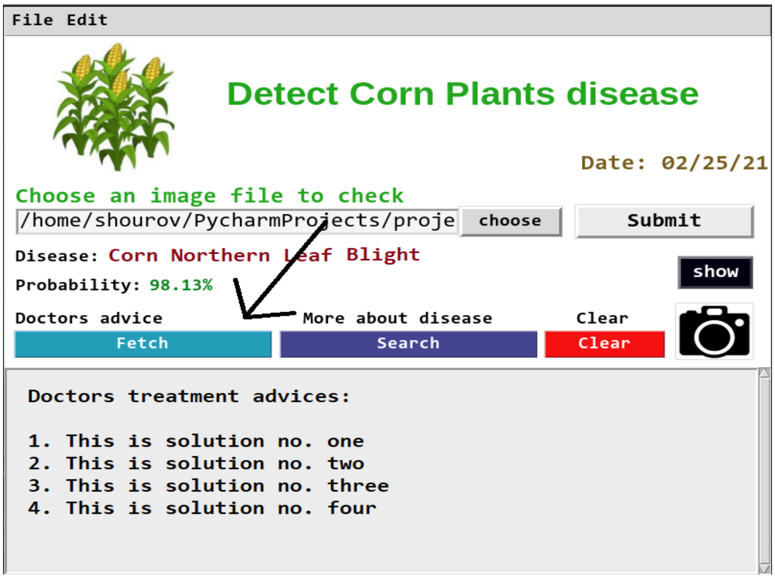
1. **Classification:** Feature classification is used for plant disease detection. The diseased features are removed from the plant and classified with the healthy leaf image. When the leaf is healthy and there is no classification the results are shown as healthy and when there is a disease which when grey scaled shows black spots, it classifies them so they are shown as which disease they are and the confidence of the classification. Classification takes place between two numerical arrays. If the numerical arrays match, then it is a healthy or a diseased leaf, depending upon the dataset provided.





Classification is a simple but relevant procedure which gives a proper result and is used in plant disease detection.

2. **Solution for any disease:** Plant disease, an impairment of the normal state of a plant that interrupts or modifies its vital functions. All species of plants, wild and cultivated alike, are subject to disease. Although each species is susceptible to characteristic diseases, these are, in each case, relatively few in number. The occurrence and prevalence of plant diseases vary from season to season, depending on the presence of the pathogen, environmental conditions, and the crops and varieties grown.



Detect Corn Plants disease

Date: 02/25/21

Choose an image file to check

/home/shourov/PycharmProjects/proje

Disease: Corn Northern Leaf Blight

Probability: 98.13%

```
Doctors treatment advices:

1. This is solution no. one
2. This is solution no. two
3. This is solution no. three
4. This is solution no. four
```

In this case the getting solution for any disease is also an import part. Because after identifying a disease we need solution to prevent damages of plants. The user will get doctors solution.

# Description of the project

□ **What is this for? What it does?**

Our project is basically for all the people living in the city who like to do gardening. It will be able to diagnose the disease by scanning the images of the leaves of any plant and will be able to tell the way to cure the disease. As well as we will able to know about some more symptoms of a disease.

□ **How will the user be benefited from this?**

End users Will be able to diagnose any disease in a subtle way without any specialist. Here you can learn about the diagnosis as well as the advice of a doctor and more symptoms of the disease. You can easily treat plants at home, which will make gardening more enjoyable.

## Data Storage:

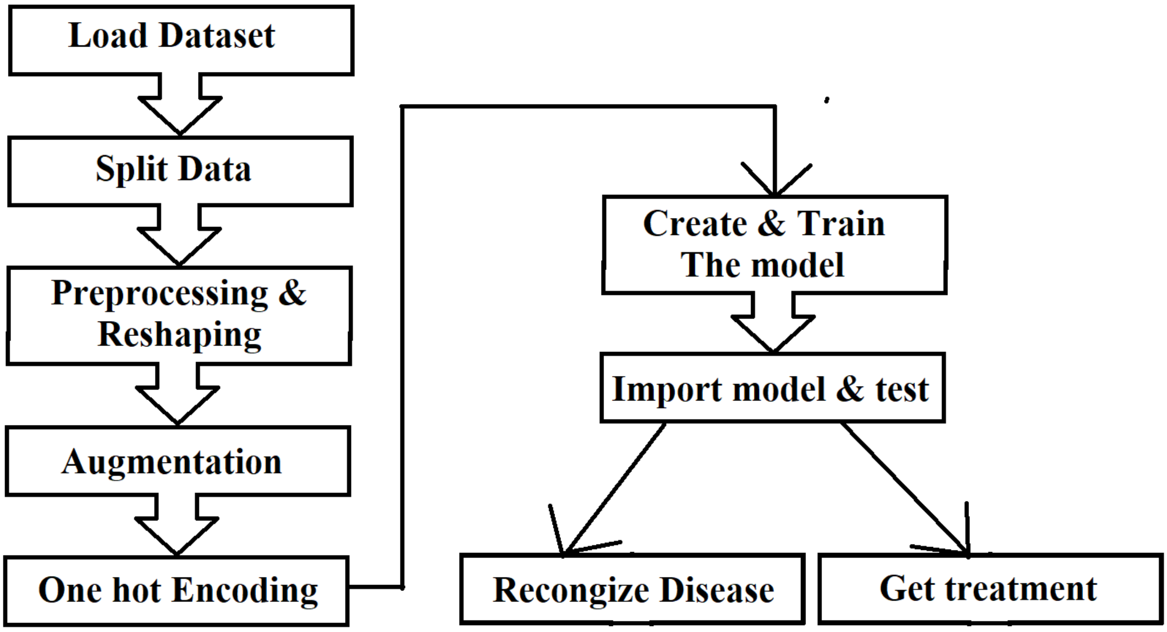
In case of our project, some data are stored in variable and some are as file. For example, pickle, csv files etc.

# Functionalities:

1. **System description**
   - We are going to start by importing the data, which is our images of plant leaves 3 to 6 types of disease classes.
   - We will have folders for each class which contains about one thousand to two thousand images each.
   - We will put all these images in a single list and create a corresponding list which contain all labels of the images.
   - Then we will split the data into i) training, ii) testing, iii) validation.
   - Later we will plot the training images distribution so that we can see if the data of each class is distributed equally.
   - The images will be processed next to optimize for the training process.
   - Then we augment the data more general, we will use function rotation, reshape, zoom etc.
   - Next, we will one hot encode the matrix.
   - Later we will create the model and start training process. once the process is done, we will plot the results.
   - After training process completed we will save the trained model and import it on a test script. Now this will test with image files.

- Finally, we’ll load the trained model and predict classes from our test images.

2. **Flow diagram of the system**

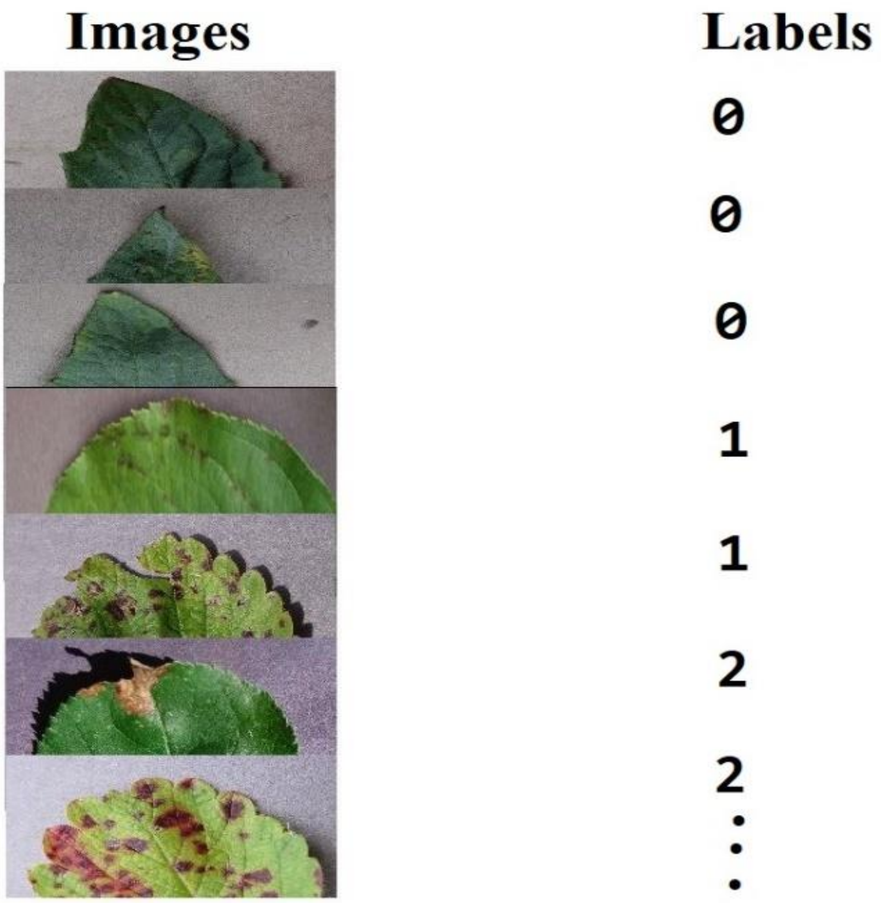


- ## Load Image Dataset:

Plant Village dataset consists of 67,905 images of different plant leaves which are divided into 30 classes. The dataset consists of 11 types of plant species and 30 types of plant diseases. The dataset contains both healthy and diseased crop images.

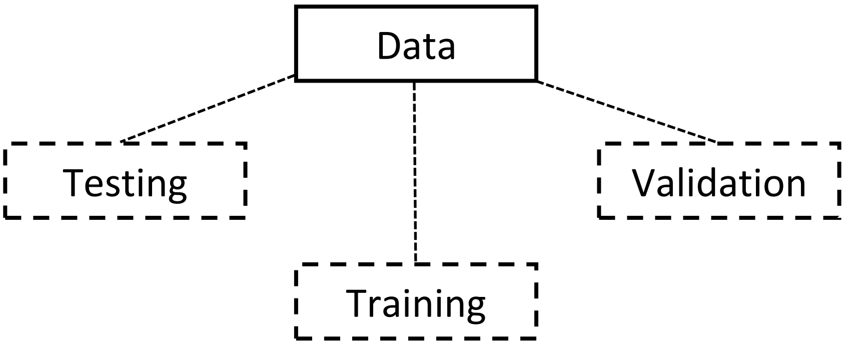
Import Dataset from different disease classes

We will put all these images in a single list and create a corresponding list which contain all labels of the images.



- **Split the data**

  Then we will split the data into i) training, ii) testing, iii) validation. Where the training and testing ratio will be 80% : 20% of total samples. And the validation will be 20% of training data. We should separate our data into train, validation, and test splits to prevent our model from overfitting and to evaluate our model accurately.



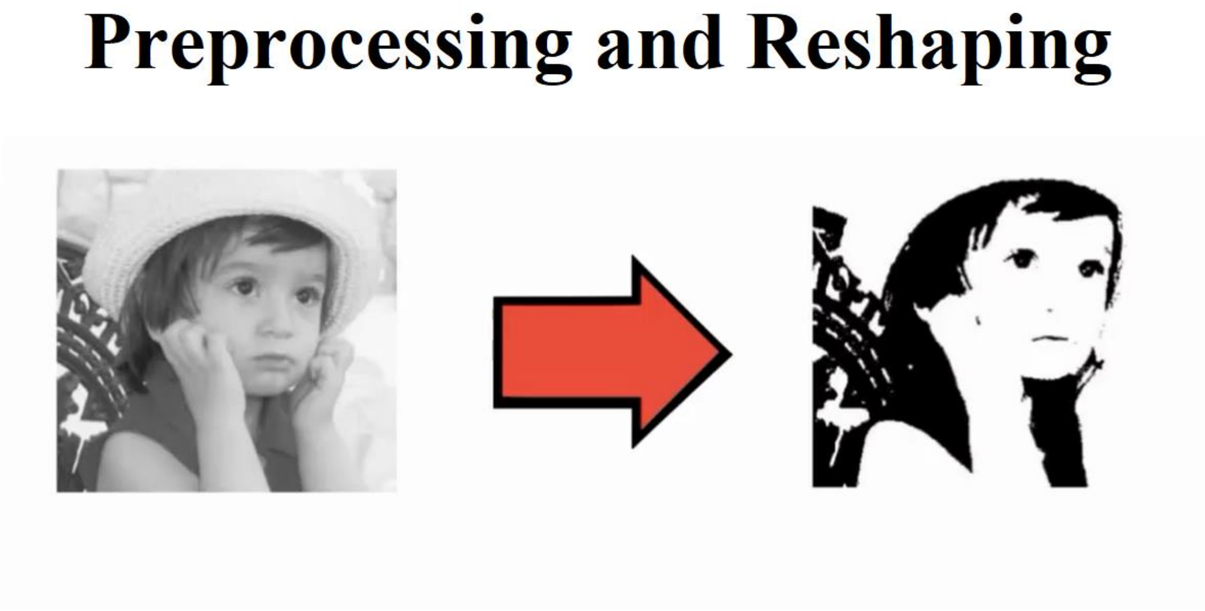
**The training set** the largest corpus of your dataset that you reserve for training your model. After training, inference on these images will be taken with a grain of salt, since the model has already had a chance to look at and memorize the correct output.

**The validation** set is a separate section of our dataset that you will use during training to get a sense of how well our model is doing on images that are not being used in training.

During the run evaluation metrics on **the Test set** at the very end of our project, to get a sense of how well our model will do in production.

- **Pre-processing and Reshaping**

Pre-processing is a very important step in CNN as the images in the dataset may have some inconsistency which may affect the accuracy of the system. The images in the dataset have noise and non-uniform lighting which needs to be rectified in this step. We do so by applying segmentation on the images to get rid of uneven backgrounds. Through segmentation we extract the relevant part of the images which in this case are the image of leaves. Hence, after segmentation we have the images of leaves with black background.



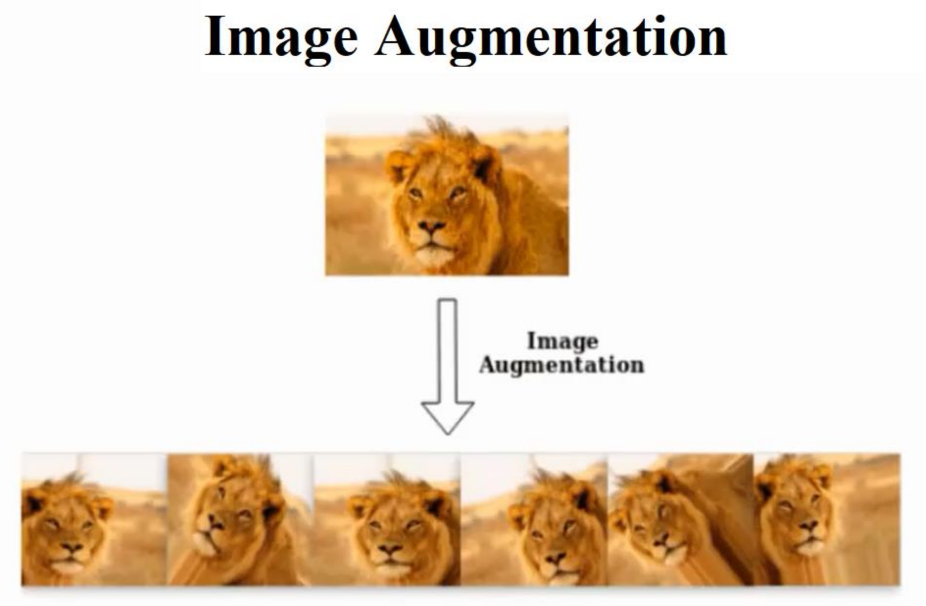
Now to rectify the non-uniform lighting we convert the images to grayscale images and send it for further processing.

## Augmentation

### What is Data Augmentation?

Data augmentation is a technique to artificially create new training data from existing training data. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples. In a word we augment the data to be more general. We will use functions such as

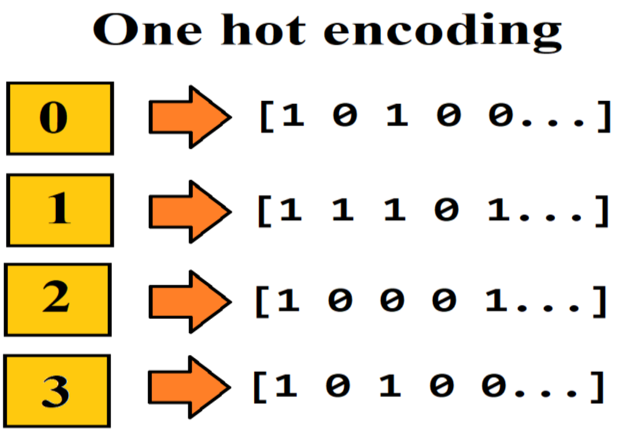
❖ Rotation<br>
❖ Translations,<br>
❖ Zoom<br>
❖ etc.



## One-Hot Encoding

### What is One hot encoding?

One Hot Encoding is a common way of pre-processing categorical features for machine learning models. This type of encoding creates a new binary feature for each possible category and assigns a value of 1 to the feature of each sample that corresponds to its original category. We will onehot encode the matrix.



- **Create and Training the model**

  Later we will create the model and start training process. Train the model using the keras fit() function, providing the training data, target data, and the number of epochs the experiment should run. There are some parameters in fit function. Batch_size, epoch, steps_per_epoch.

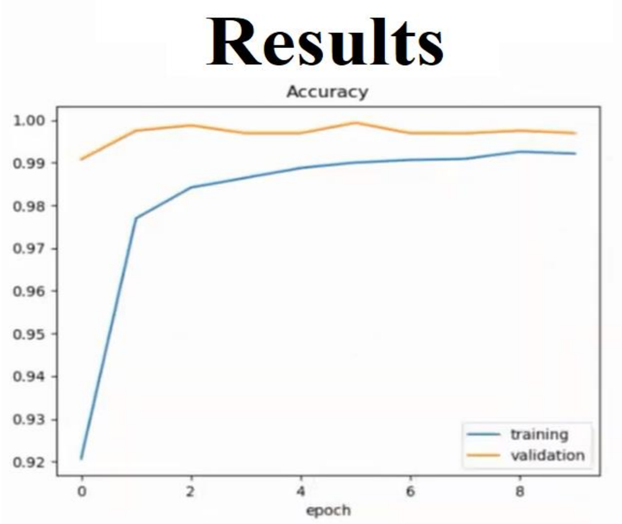
  Assume we have a dataset with **8000 samples** and you choose a batch_size = 32 and epochs = 25. This means that the dataset will be divided into (8000/32) = **250 batches**
  One epoch will train 250 batches or 250 updations to the model.
  here steps_per_epoch = no.of batches (minimum)
  With 25 epochs, the model will pass through the whole dataset 25 times.

  # In our project we've used these values to train the model.
  Image Dimensions= (32 x 32) px [ after resizing ]
  batch_size = 50
  epochs= 10
  steps_per_epoch = 2000

  Here Total number of layers used is 11, consisting 4 convolutional layers, 2 max-pooling layers, 1 flatten layer, 2 Dense layers and 2 DropOut layers.

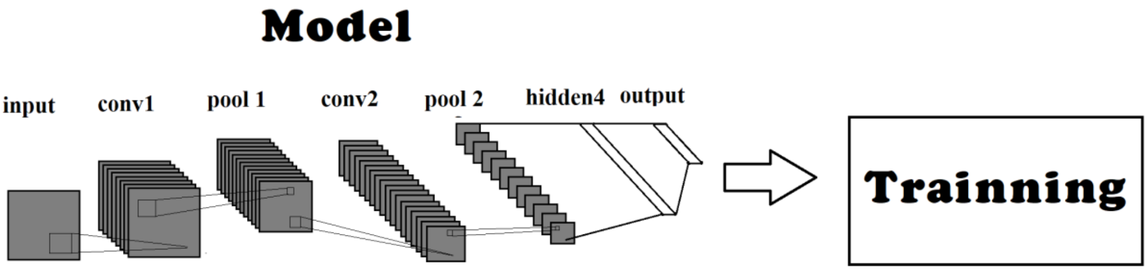
| Layer | Type | Activation Function |
|---|---|---|
| conv2d_1 | Conv2D | ReLU |
| conv2d_2 | Conv2D | ReLU |
| max_pooling2d_1 | MaxPooling2 | ReLU |
| conv2d_3 | Conv2D | ReLU |
| conv2d_4 | Conv2D | ReLU |
| max_pooling2d_2 | MaxPooling2 | ReLU |
| dropout_1 | Dropout | ReLU |
| flatten_1 | Flatten | ReLU |
| dense_1 | Dense | ReLU |
| dropout_2 | Dropout | ReLU |
| dense_2 | Dense | SoftMax |

Once the process is done, we will plot the results.



- **Disease detection and classification:**

  Detection of disease is performed in two steps i.e. detection of the type of crop and detection of type of disease. This takes place with the help of Convolutional Neural Network. We will be using Transfer Learning for building the Model. It is a technique where the pertained models are used to create the current models. Classification also acts as fully connected classifiers which are formed using various learnings done by the model. We do the following by flattening of images which convert the pooled images to single dimension vectors. Once the images are converted to the vectors it gets quite easy to classify the images. Through the trained model we get certain numerical values with respect to various classes. When the leaf is healthy and there is no classification the results are shown as healthy and when there is a disease which when grey scaled shows black spots, it classifies them so they

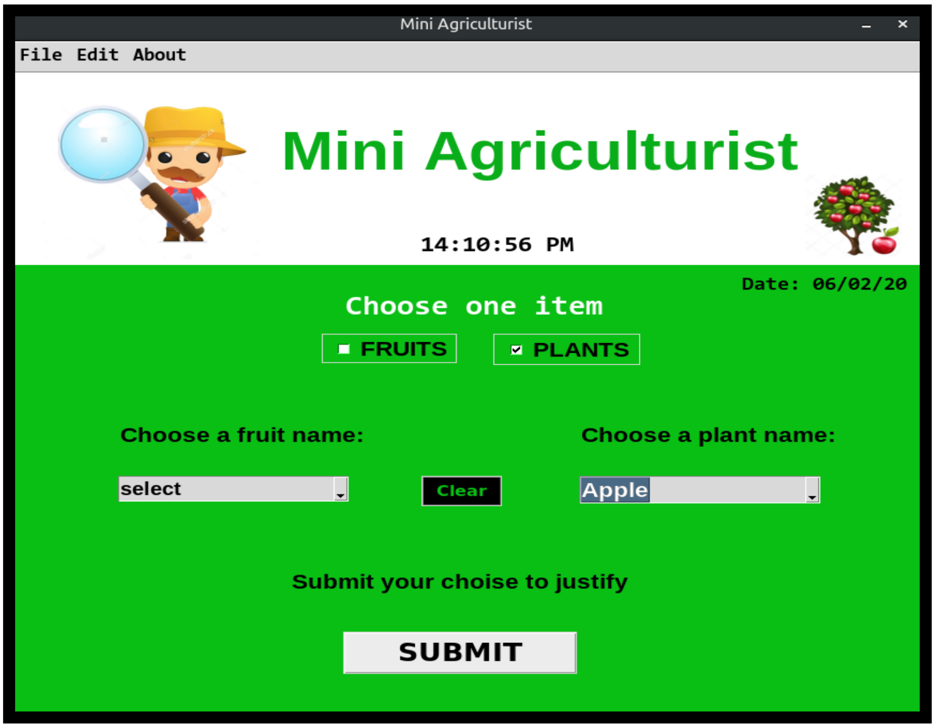


are shown as which disease they are and the confidence of the classification. Classification takes place between two numerical arrays. If the numerical arrays match, then it is a healthy or a diseased leaf, depending upon the dataset provided. Classification is a simple but relevant procedure which gives a proper result and is used in plant disease detection.

- **Making predictions**

  Finally, we will save the trained model and import it on a test script. The predict() function returns an array with 10 numbers, these are the probabilities that an image contains each possible digit from 0 to 9. Run a prediction for the first four images in the test set, and display the first four values in y_test to compare to the actual results.

## UI of this project



14:10:56 PM

Date: 06/02/20

Choose one item

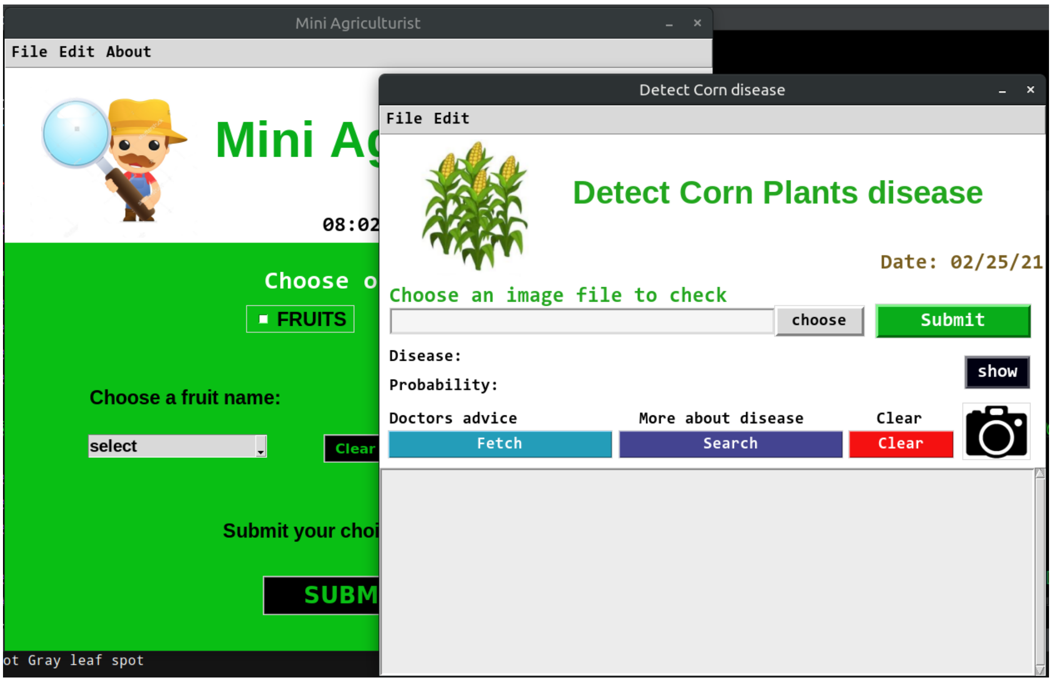
□ FRUITS ☑ PLANTS

Choose a fruit name: select

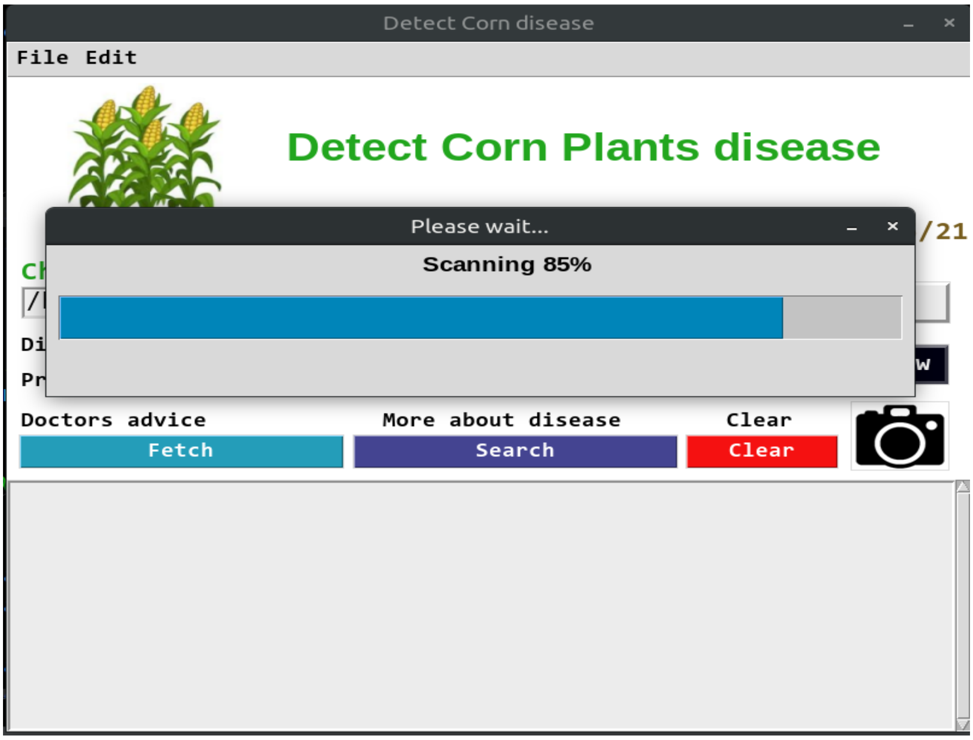
Choose a plant name: Apple

Submit your choise to justify

1. This is home screen of this project. Here you just have to select one item then chose the specific item name. After selecting go to new window by clicking submit button.

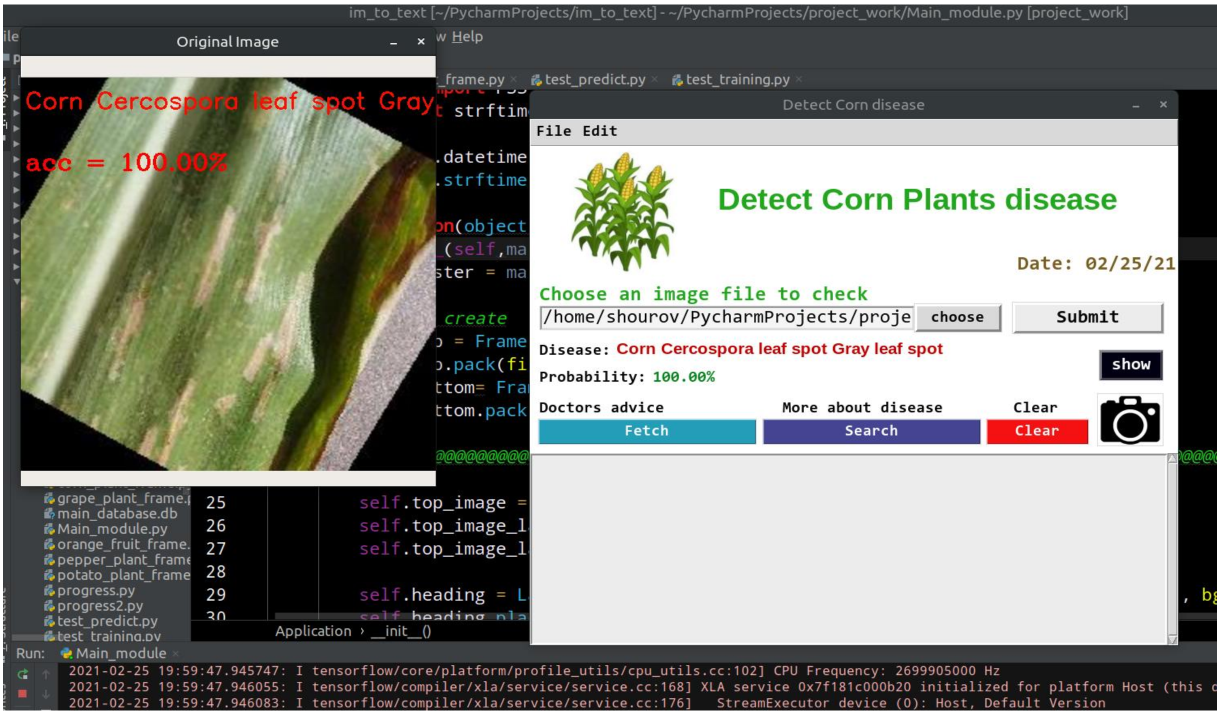


2. This window will be separate for each item. Here the user has to choose the image of the leaves of the diseased tree. Then click on the submit button to diagnose.



3. After submitting an image it will take a moment to load the prediction. Then we can choose different images to classify specific disease one by one.

i)	Disease name: Corn Corcospora leaf spot gray, Accuracy: 100% (1.00)


im_to_text [~/PycharmProjects/im_to_text] - ~/PycharmProjects/project_work/Main_module.py [project_work]

Corn Cercospora leaf spot Gray

acc = 100.00%

Date: 02/25/21

Choose an image file to check

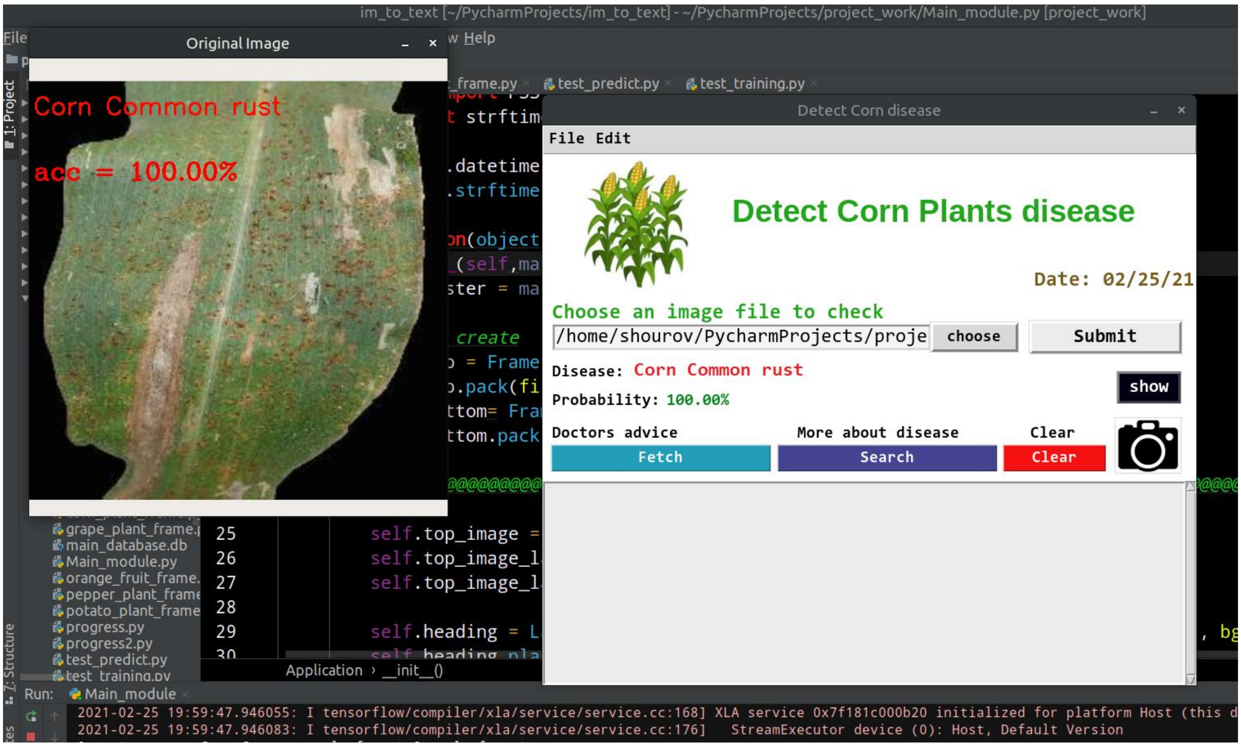
/home/shourov/PycharmProjects/proje

Disease: Corn Cercospora leaf spot Gray leaf spot

Probability: 100.00%

```
2021-02-25 19:59:47.945747: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2699905000 Hz
2021-02-25 19:59:47.946055: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f181c000b20 initialized for platform Host (this c
2021-02-25 19:59:47.946083: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
```

ii)	Disease name: Corn Common Rust, Accuracy: 100% (1.00)


im_to_text [~/PycharmProjects/im_to_text] - ~/PycharmProjects/project_work/Main_module.py [project_work]

Corn Common rust

acc = 100.00%

Date: 02/25/21

Choose an image file to check

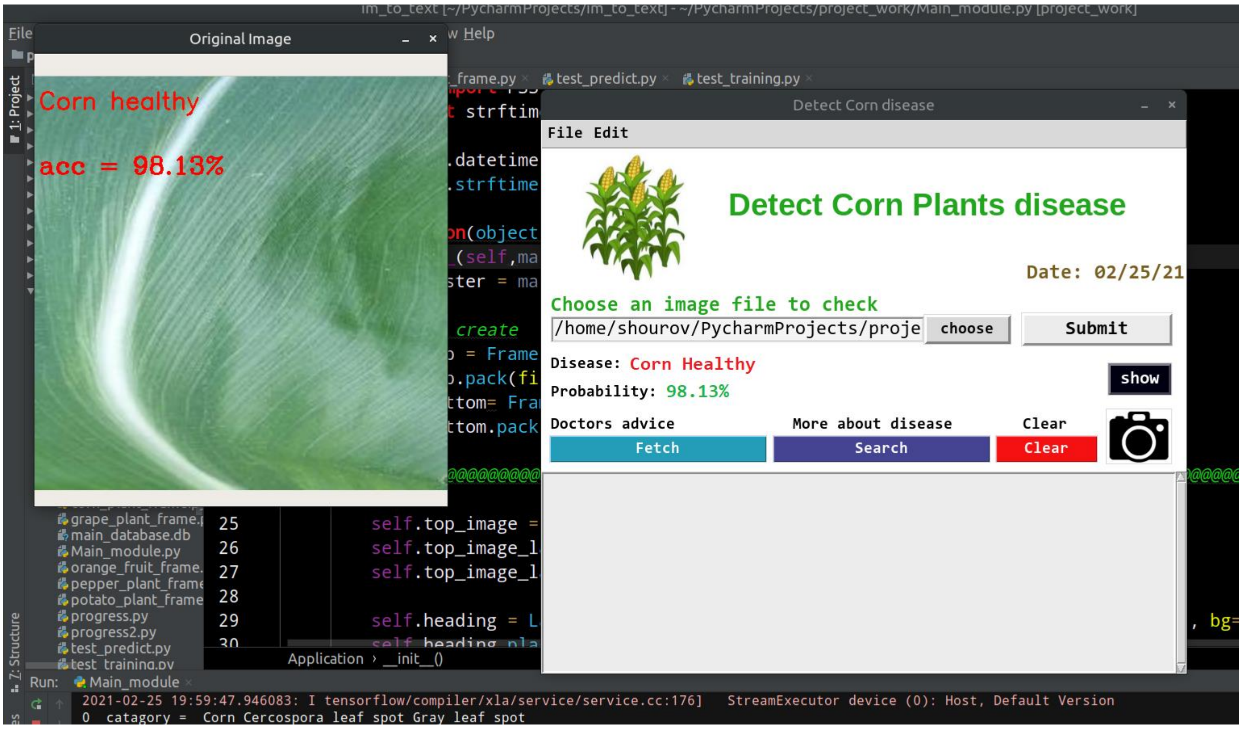
/home/shourov/PycharmProjects/proje

Disease: Corn Common rust

Probability: 100.00%

```
2021-02-25 19:59:47.946055: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f181c000b20 initialized for platform Host (this d
2021-02-25 19:59:47.946083: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
```

iii) Disease name: Corn Healthy, Accuracy: 98.13% (0.9813)



Corn healthy

acc = 98.13%

Detect Corn Plants disease

Date: 02/25/21

Choose an image file to check

/home/shourov/PycharmProjects/proje

Disease: Corn Healthy

Probability: 98.13%

Doctors advice More about disease Clear

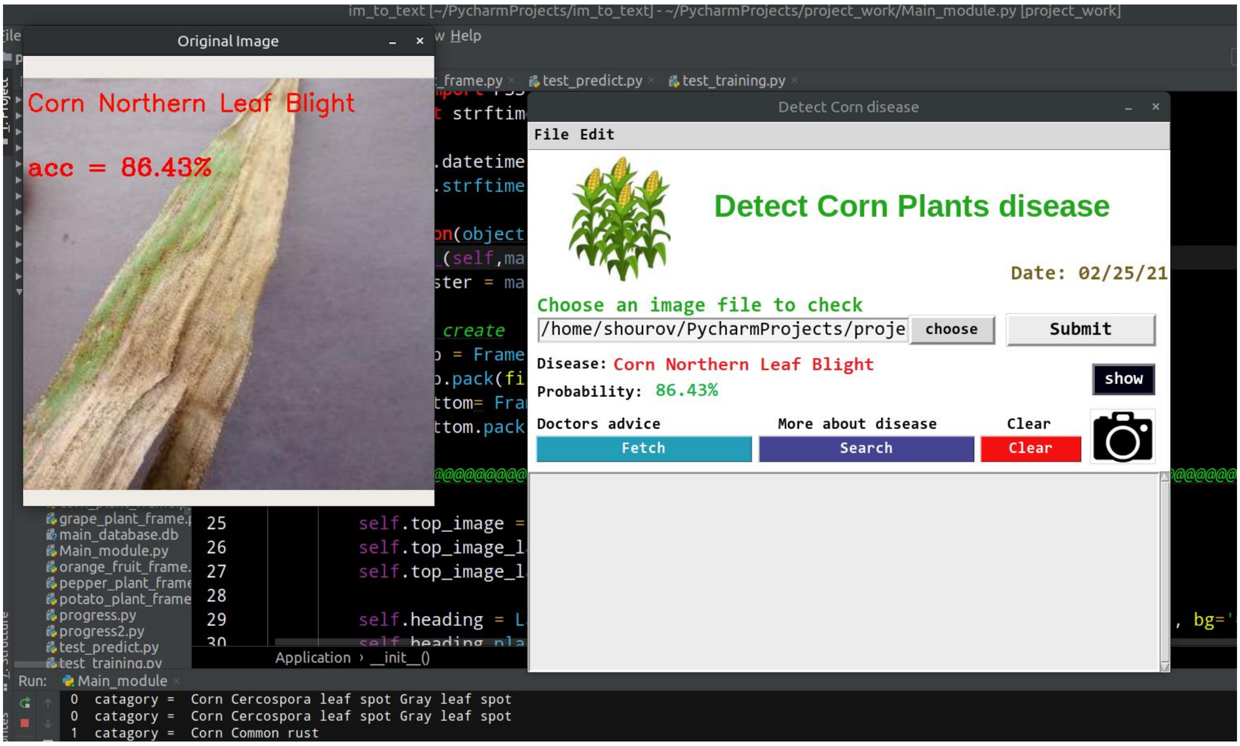
```
grape_plant_frame.p
main_database.db
Main_module.py
orange_fruit_frame.
pepper_plant_frame
potato_plant_frame
progress.py
progress2.py
test_predict.py
test_training.py
```

```
    self.top_image =
    self.top_image_l
    self.top_image_l

    self.heading = L
```

```
2021-02-25 19:59:47.946083: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
0  catagory =  Corn Cercospora leaf spot Gray leaf spot
```

iv) Disease name: Corn Northern Leaf Blight, Accuracy: 86.43% (0.8643)



Corn Northern Leaf Blight

acc = 86.43%

Detect Corn Plants disease

Date: 02/25/21

Choose an image file to check

/home/shourov/PycharmProjects/proje

Disease: Corn Northern Leaf Blight

Probability: 86.43%

Doctors advice More about disease Clear

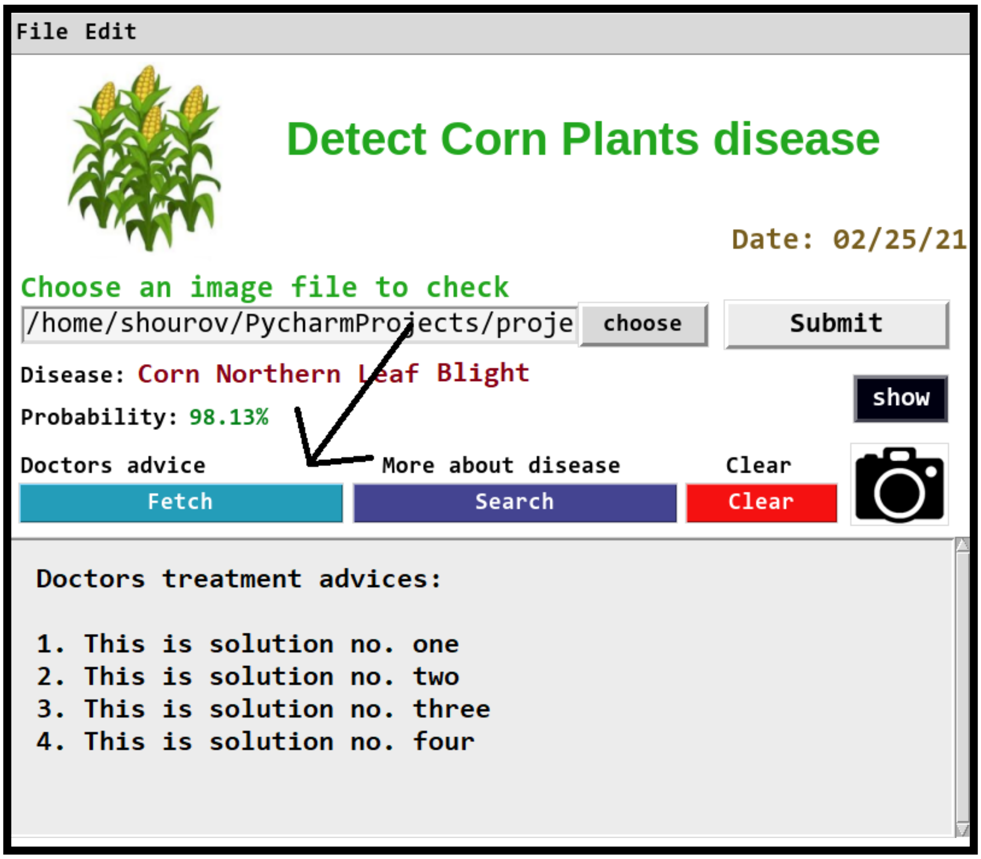
```
grape_plant_frame.p
main_database.db
Main_module.py
orange_fruit_frame.
pepper_plant_frame
potato_plant_frame
progress.py
progress2.py
test_predict.py
test_training.py
```

```
    self.top_image =
    self.top_image_l
    self.top_image_l

    self.heading = L
```

```
0  catagory =  Corn Cercospora leaf spot Gray leaf spot
0  catagory =  Corn Cercospora leaf spot Gray leaf spot
1  catagory =  Corn Common rust
```

4. User can find primary treatment about any disease as well as more symptom about it from our software. After identifying just click the buttons bellow the probability, it will fetch required information.



Detect Corn Plants disease

Date: 02/25/21

Choose an image file to check

/home/shourov/PycharmProjects/proje

Disease: Corn Northern Leaf Blight

Probability: 98.13%

```
Doctors treatment advices:

1. This is solution no. one
2. This is solution no. two
3. This is solution no. three
4. This is solution no. four
```

**Conclusion:** In this paper the Deep Learning algorithm i.e. Convolutional Neural Network is used to with a goal to detect the diseases in the plants. The model is basically tested on some types of plant species with some types of plant diseases. The model was made using Tensor flow and Keras frameworks and the system is implemented on Linux. As an extension to the project the number of classes of plants and its diseases will be increased. Also the model will be further improved by increasing the parameters for training and test.

## Limitation & Future plan:

i) In case of treatment of the plants diseases we need to discuss with an agriculture specialist, those we are providing in our project and their solutions. The accurate treatments are still unavailable in our project.

ii) The realtime prediction accuracy is very poor. Next time we will analyse reasons and will try to improve it.

iii) Here we've created only desktop based software (exe file). Later we would like to build the android version for this project.

## Reference:

1. [ Plant Disease Detection using Machine Learning, Volume: 07 Issue: 07 | July 2020, International Research Journal of Engineering and Technology (IRJET), 2020 - www.irjet.net/ ]