Chapter 1 Introduction

1.1 Introduction to Project

Potato is one of the major harvests in our country. The production of potatoes is hampered by Many kinds of Pests and diseases. So we can’t export potatoes to our expectations in the other countries. Among them early blight, leaf roll virus, scab, Hollow heart etc are the most terrible disease of potatoes at previous present and times, the major area's farmers faced many hampers on this disease every year [5]. The farmers and businessmen of our country are facing many problems with those diseases particularly in the case of export to other countries.

There are various occupations in the world but majorly agriculture is the primary occupation [3]. India economy not exception to it, which depends on agriculture a lot Potato is the one of the major crop which contributes to about 28.9% of total agricultural crop production in India. Potato is a fourth largest agricultural food crop in the world after maize, wheat and rice. India is a 2nd largest country in the production of potatoes which produces 48.5 million every year [1].

Early detection of these disease can allow to take preventive measures and mitigate economic and production losses [1]. Over the last decades, the most practiced approach for detection and identification of plant diseases is naked eye observation by expert. But in many cases, this approach proves unfeasible due to the excessive processing time and unavailability of experts at farms located in remote areas.

Agriculture is an essential sector in countries like India as those countries’ economy directly or indirectly dependent on agriculture [1]. It indicates the necessity of taking care of plants from seedling until the expected crop obtains. Through this process, the crop needs to cross a lot of phases to obtain the expected crop such as weather conditions, the survival of the crop from various diseases, and the survival of the crop from various animals. Of these major phases, the crops can be protected from the various animals by providing proper protection for the field and this issue can be solvable. The next major issue is weather conditions which will not be in the control of humans, humans can only pray for better weather conditions to obtain a better crop. Finally, the major issue which is very crucial to protect the crop from various diseases as these diseases can impact the complete growth and yield of the crop. If one can able to identify these diseases in time, then the crop can be protected using appropriate fertilizers. If this process of identification and classification of diseases able to digitalize which would be helpful for the agriculturists [2]. It will decrease the time for the identification of disease and precision in classifying the diseases. There are a lot of significant crops exist in India, one among them is Potato. More than three-fourths of the population of India consumes potato daily at the same time it is one of the popular yielding crops in India. Yet, the yield of the potato crop can be diminished due to various diseases such as late blight and early blight.

These diseases are also known as Phytophthora Infestans and Alternaria Solani respectively in scientific terms. Timely identification and classification of these diseases will lead to avoid the yield as well as financial losses [6]. The popular way of identification of these diseases through the utilization of the human eye for decades. But this methodology arises with certain infeasibilities such as overtime will be taken for processing and shortage of experts at fields in remote locations. Therefore, the image analysis turned out to be an efficient methodology that will play a vital role in monitoring as well as the identification of the plant disease conditions effectively. Because the visible patterns are available on the plant leaves and patterns will be identified using various image processing methodologies for obtaining a particular pattern corresponding to a disease which will create an impact in the identification of various diseases. Thus, the obtained features or patterns will be compared with the historical data and able to classify the disease which can be done by using various machine learning methodologies [7].

So, the combination of image processing and machine learning is very effective in the identification and classification of diseases. The present document was structured into various sections such as section-1 introduces the necessity of plant leaves diseases detection using image processing and machine learning methodologies, section -2 deals with the study of previous research based on identification and classification of diseases, precisely, a literature review, section-3 discusses the various methodologies necessary for the detection and classification of plant leaves, section-4 discusses the generated results based on the proposed framework and evaluation metrics, lastly, section-5 generates the conclusion of the presented work and future work based on this framework.

1.2 Project Category

Research

1.3 Problem Formulation

The production of potatoes decreases Due to some diseases. We believe that, if we detect the disease of potato properly and provide the proper treatment to increase the production growth. Farmers Identification of leaf diseases it is the important and one of the major problem in early stages. Disease is caused by pathogen which is any agent causing disease. In most of the cases pests or diseases are seen on the leaves or stems of the plant. Therefore identification of plants, leaves, symptoms and finding out the pest or diseases, percentage of the pest or disease incidence, symptoms of the pest or disease attack, plays a key role in successful cultivation of crops. What is wrong with my plant; followed by, what can I do to get rid of the problem? It may be too late to help the specific plant when the question is asked, but proper diagnosis may be extremely important in preventing the problem on other plants or in preventing the problem in the future. Control measures depend on proper identification of diseases and of the causal agents. Therefore, diagnosis is one of the most important aspects of a plant pathologist's training. Without proper identification of the disease and the disease causing agent, disease control measures can be a waste of time and money and can lead to further plant losses. consider various environmental and cultural factors. Be able to identify a disease and disease-causing agent, Be able to narrow the problem down to several possibilities which will require further study in the laboratory before he can make a final diagnosis, or Identify characteristic symptoms. Describing the characteristic symptoms exhibited by a specimen can be very difficult to do accurately. Because of this, it is often difficult, if not impossible, to determine what is wrong with a plant when a person is describing symptoms over the phone. As a test of this you may want to take a plant exhibiting symptoms and have three different individuals describe the symptoms that they observe on a sheet of paper.

1.5 Existing System

Image segmentation and multiclass support vector machine which may be used for early detection of potato leaf Mold once it's troublesome to gather pictures of thousands of latest pages. The experiments enclosed pictures 100 healthy leaves, 100 blight leaves and 100 early Blight. Total 300 potato leaf images are taken as dataset. The program predicts with 95% accuracy.

1.5.1 Limitations

- Accuracy is less.
- It consumes more training time.
- The existing system detects potato disease of only 3 kinds.
- The existing system has been trained with less number of potato leaf images.

1.6 Objectives

1. To gather and preprocess a diverse dataset of potato disease classification.
2. To develop a deep learning model to identify and classify different potato disease.
3. To compare the model's accuracy with other method.

1.7 Proposed System

All through the proposed model, the CNN-VGG16 algorithm is utilized to recognize various kinds of potato infections, sicknesses. It avoids the use of Image segmentation and multiclass support vector machine. VGG16 was identified to be the best performing model on the ImageNet dataset.

Advantages

- Accuracy is high.
- It consumes less training time when compared to existing system.
- Ensures the potato quality.
- Helps to consumers to buy the healthy potato.
- It classifies four types of diseases
- It also suggests the type of pesticide to be used for a particular disease.

Chapter 2. Requirement Analysis and System Specification

2.1 Feasibility study

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. FEASIBILTY STUDY is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. The steps involved in the feasibility analysis are:

- Form a project team and appoint a project leader.
- Enumerate potential proposed system.
- Define and identify characteristics of proposed system.
- Determine and evaluate performance and cost effective of each proposed system. ¬ Weight system performance and cost data.
- Select the best proposed system. ¬ Prepare and report final project directive to management.
- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.1.1 ECONOMICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.1.2 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system

2.1.3 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.2 SYSTEM REQUIREMENTS AND SPECIFICATION

System Requirement Specification (SRS) is a central report, which frames the establishment of the product advancement process. It records the necessities of a framework as well as has a depiction of its significant highlight. An SRS is essentially an association's seeing (in composing) of a client or potential customer's frame work necessities and conditions at a specific point in time (generally) before any genuine configuration or improvement work. It's a two-way protection approach that guarantees that both the customer and the association comprehend alternate's necessities from that viewpoint at a given point in time.

The SRS talks about the item however not the venture that created it, consequently the SRS serves as a premise for later improvement of the completed item. The SRS may need to be changed; however, it does give an establishment to proceed with creation assessment. In straightforward words, programming necessity determination is the beginning stage of the product improvement action.

The SRS means deciphering the thoughts in the brains of the customers – the information, into a formal archive – the yield of the prerequisite stage. Subsequently the yield of the stage is a situated of formally determined necessities, which ideally are finished and steady, while the data has none of these properties.

2.2.1 Functional Requirements

This section describes the functional requirements of the system for those requirements which are expressed in the natural language style.

- User should load potato dataset to our application.
- System will preprocess and extract the features using the CNN.
- Create a desktop application using python tkinter.
- System will split the dataset into trainset and testset.
- System will train the model using CNN.
- User should load input image
- System will apply CNN pre-trained model to classify the potato disease.
- Application should efficiently classify the potato disease.

2.2.2 Non-Functional Requirements

These are requirements that are not functional in nature, that is, these are constraints within which the system must work. The program must be self-contained so that it can easily be moved from one Computer to another. It is assumed that network connection will be available on the computer on which the program resides. The system shall achieve 100 per cent availability at all times. The system shall be scalable to support additional clients and volunteers. The system should be optimized for supportability, or ease of maintenance as far as possible. This may be achieved through the use documentation of coding standards, naming conventions, class libraries and abstraction. The system should be optimized for supportability, or ease of maintenance as far as possible. This may be achieved through the use documentation of coding standards, naming conventions, class libraries and abstraction. It should have randomness to check the nodes and should be load balanced.

HARDWARE REQUIREMENTS:

Processor:

11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.42 GHz

RAM:

4GB & Above

Hard disk:

512 GB SSD

Software Requirements:

Machine learning and deep learning libraries (e.g., TensorFlow, PyTorch, OpenCV).

Data preprocessing tools like pandas, NumPy, and scikit-learn.

Tools for building a mobile app (e.g., Flutter, React Native) to make the classifier user friendly.

Operating System

: Windows 10 home and above version

Language:

Python

Source Code Editor:

Google Collab, Visual studio code, Jupiter Notebook Graphics card 1080

Chapter 3. System Design

3.1 Design Approach

- Identify **key diseases** affecting potatoes (e.g., Late Blight, Early Blight, Black Scurf).
- Define the **goal**: Classify potato leaf images into healthy or diseased categories.
- Determine the **output format**: Binary (Healthy/Diseased) or Multiclass (specific diseases).

2. Data Collection & Preprocessing

2.1 Dataset Collection

- **Sources**: Kaggle, PlantVillage, own dataset via field collection.
- **Data Type**: Images of potato leaves (RGB images).
- **Annotations**: Label images with correct disease categories.

2.2 Data Preprocessing

- **Resizing**: Standardize image dimensions (e.g., 224×224 pixels).
- **Normalization**: Scale pixel values (e.g., [0,1] or [-1,1]).
- **Augmentation**: Improve model robustness (rotation, flipping, noise).
- **Splitting**: Train (70%), Validation (20%), Test (10%).

3. Model Selection & Training

3.1 Choose a Model Architecture

- **Traditional ML**: SVM, Random Forest (if using handcrafted features).
- **Deep Learning**: CNN-based architectures like:
    - ResNet-50

- MobileNetV2 (for lightweight applications)
- EfficientNet (high accuracy)
- Custom CNN

3.2 Training Strategy

- **Loss Function**: Categorical Cross-Entropy (multiclass), Binary Cross-Entropy (binary classification).
- **Optimizer**: Adam, SGD with momentum.
- **Batch Size & Epochs**: Tune based on dataset size.
- **Regularization**: Dropout, L2 weight decay to prevent overfitting.

4. Model Evaluation & Optimization

4.1 Evaluation Metrics

- Accuracy, Precision, Recall, F1-score.
- Confusion Matrix for class-wise performance.

4.2 Hyperparameter Tuning

- Learning rate optimization.
- Number of layers and filters in CNN.
- Fine-tuning dropout rates.

3.2 Detail Design

3.2.1: Flow charts or Block Diagrams

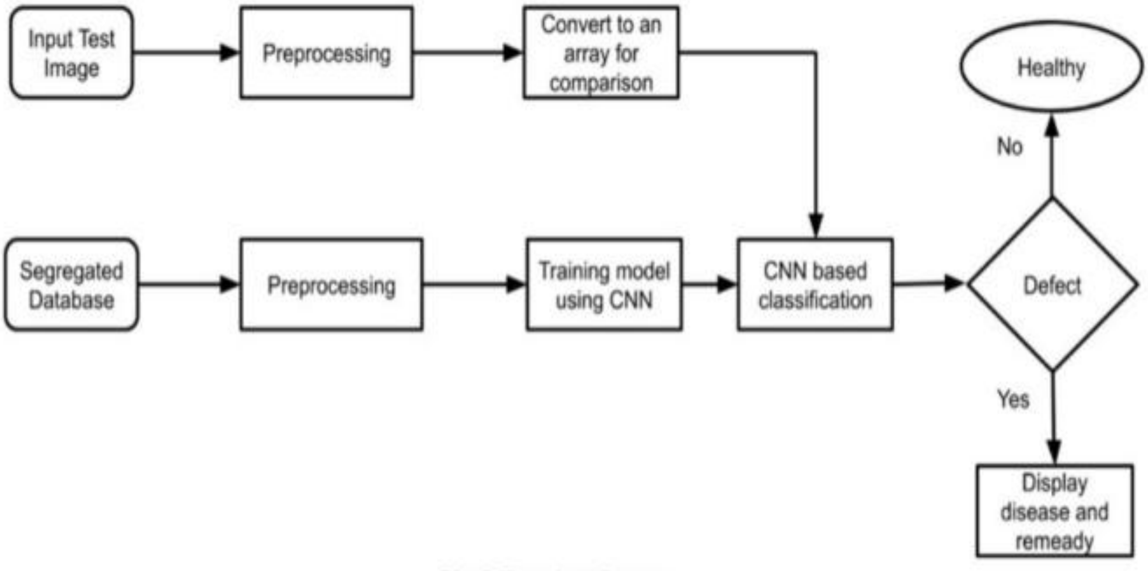


Figure 3.1: Flow chats

3.2.2: UML DAIGRAM

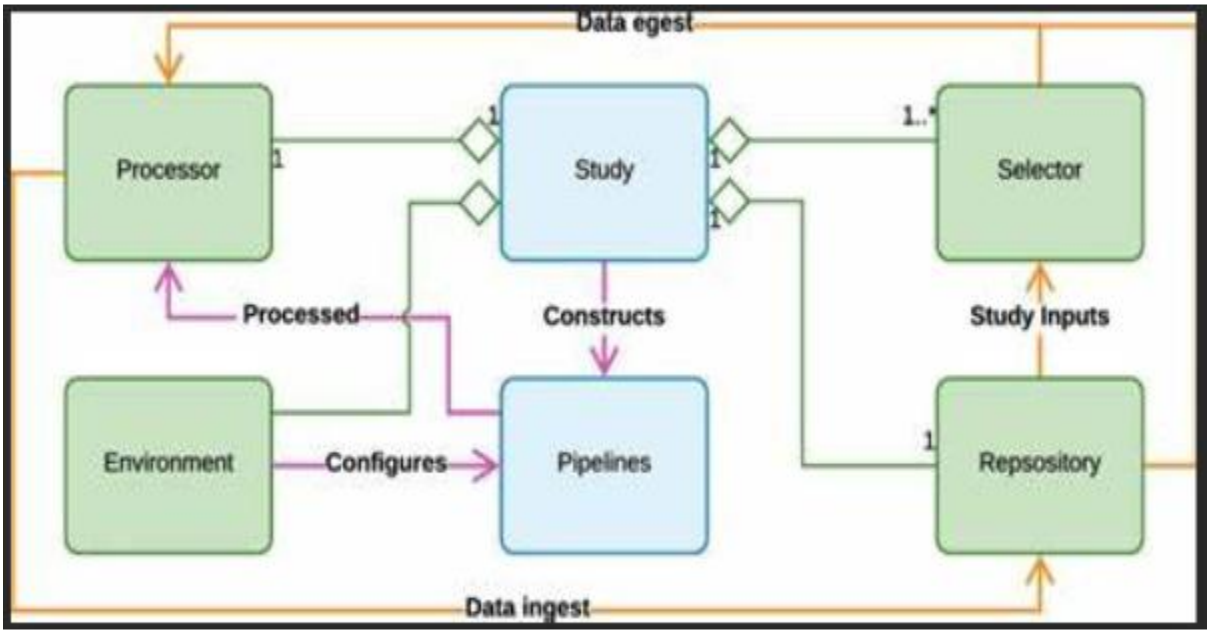


Figure 3.2: *UML diagram*

3.2.3.1: DFD

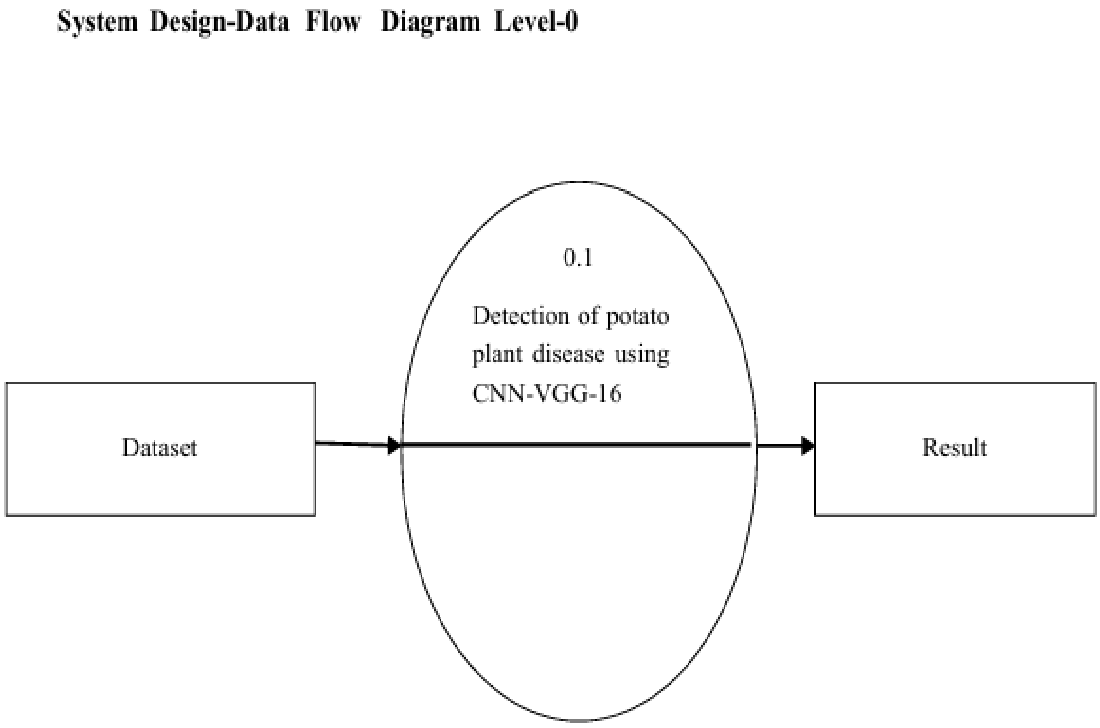


Figure 3.3: *data flow diagram level-0*

Level: 0 describes the overall process of the project. We are using potato leaf disease image dataset as input. System will use VGG-16 algorithm to predict the potato disease or not.

3.2.3.2: Data flow diagram level-1

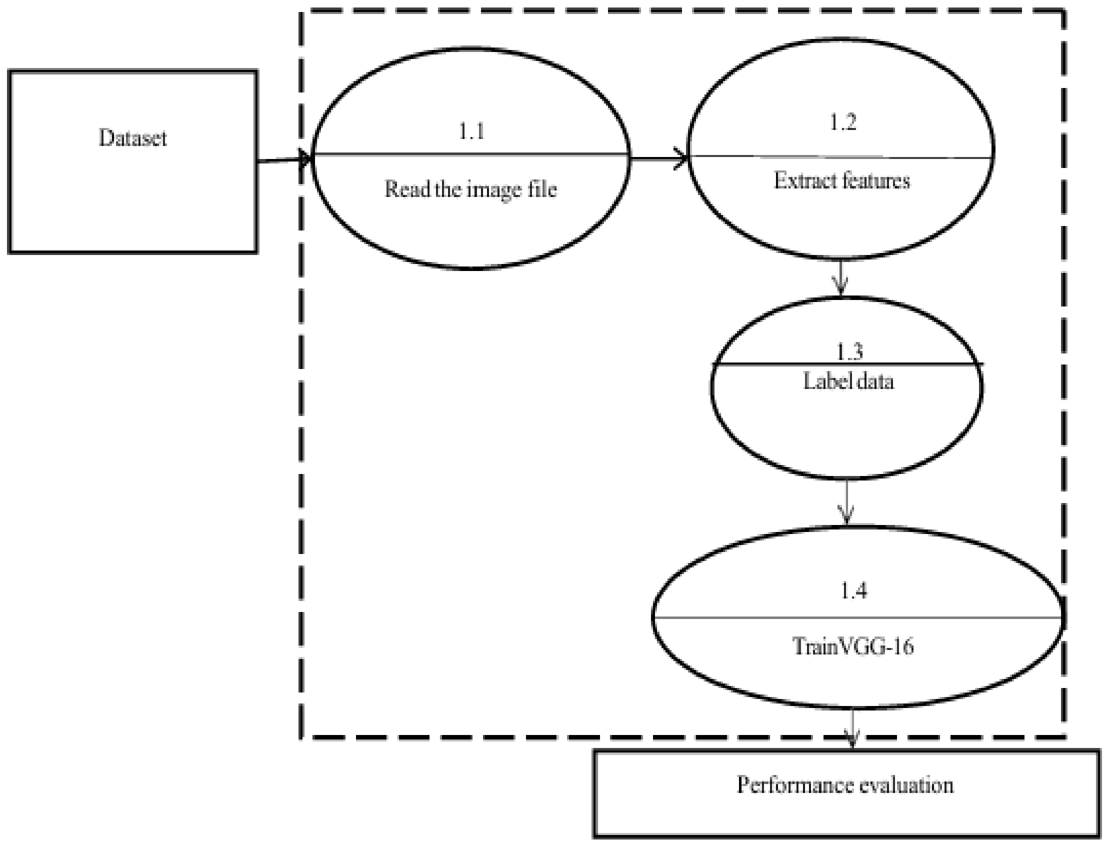


Figure 3.4: *data flow diagram level-1*

Level: 1 describes the first step of the project. We are using image dataset as input. System will use Image generator to extract the image features and train the VGG-16 model and shows the performance graph as output.

3.2.3.3: Data flow diagram level-2

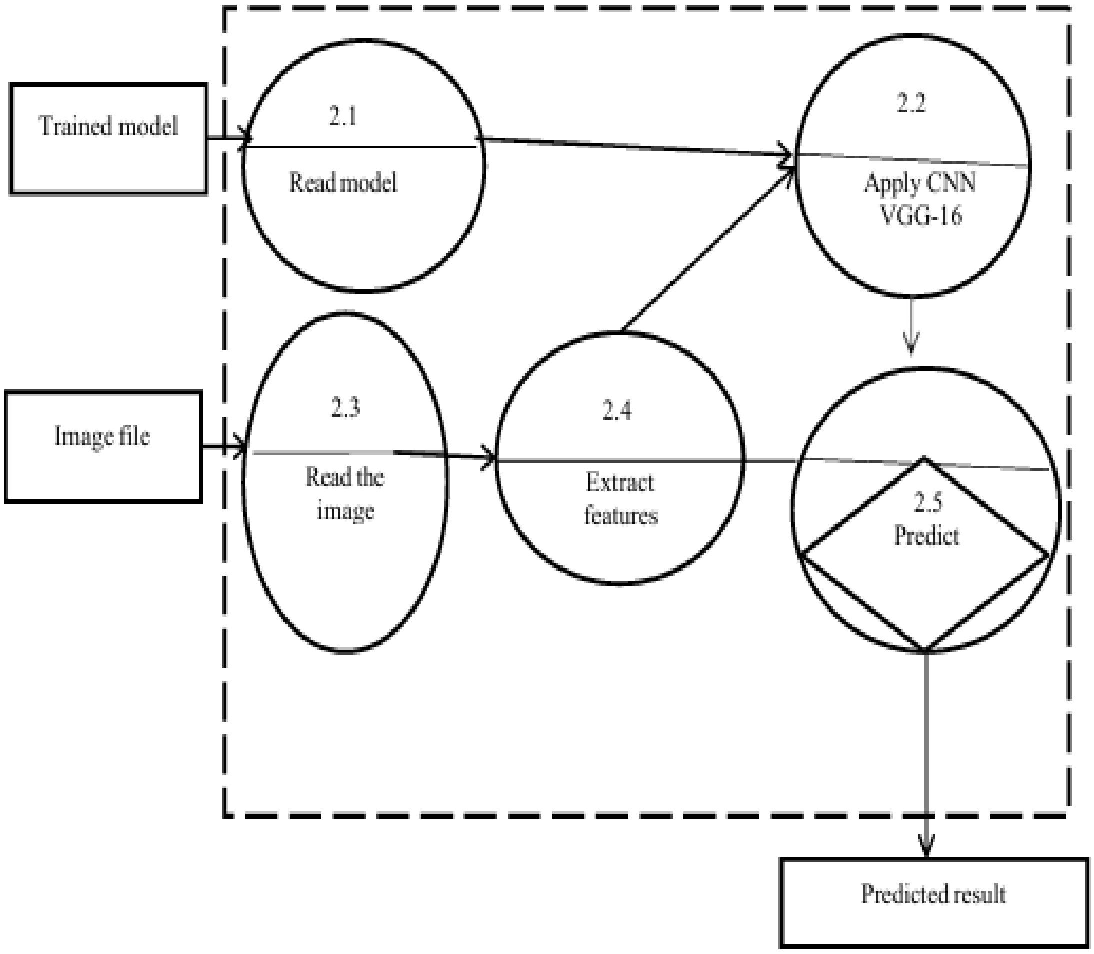


Figure 3.5: *Data flow diagram level-2*

Level: 2 describe the final step of the project. We are using trained model from level1 and image as input. System will use VGG-16 predicts the plant leaf is safe or unsafe.

3.2.4: Sequence Diagrams

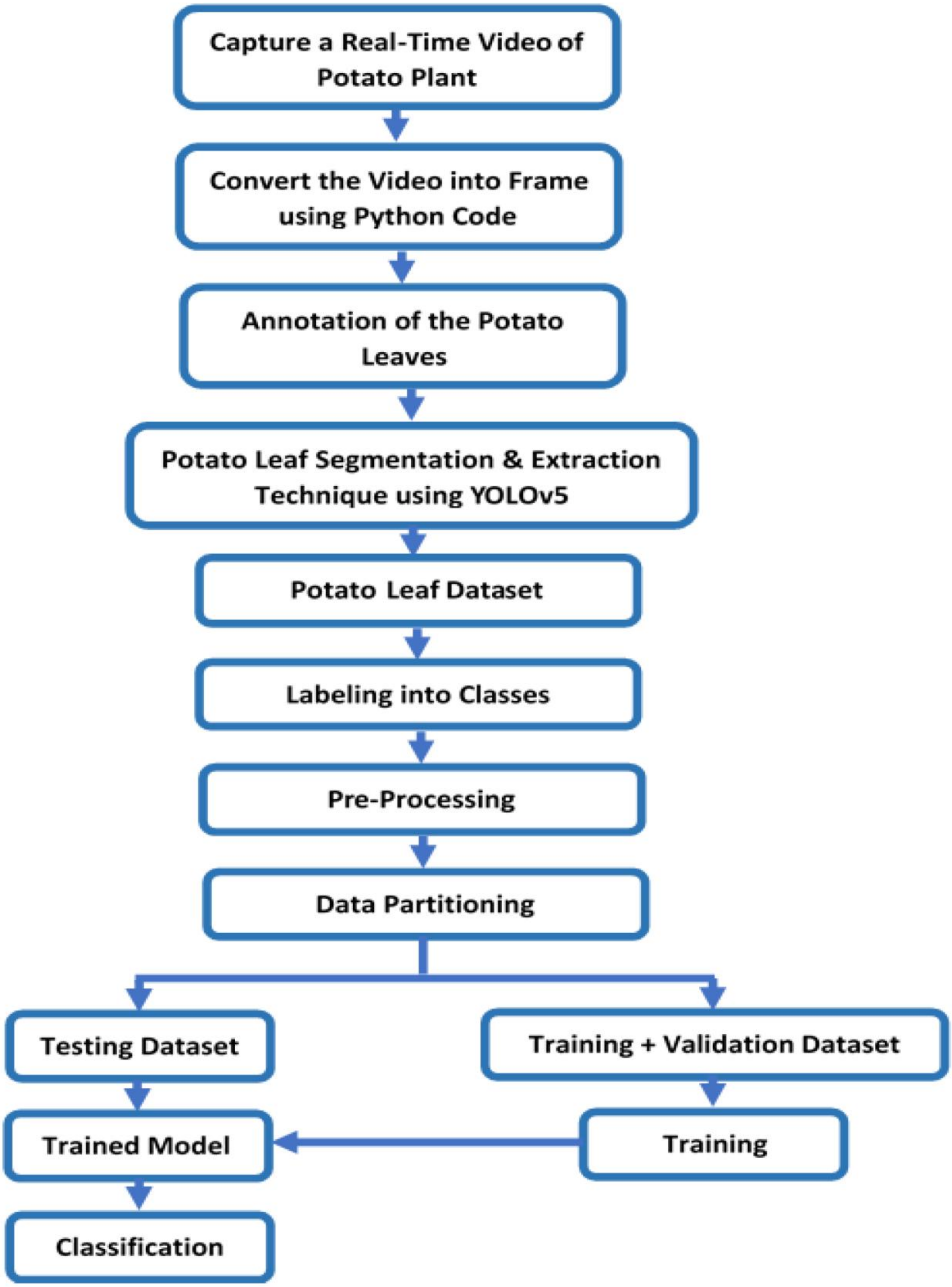


Figure 3.6: *sequence diagram*

3.2.5: Use case

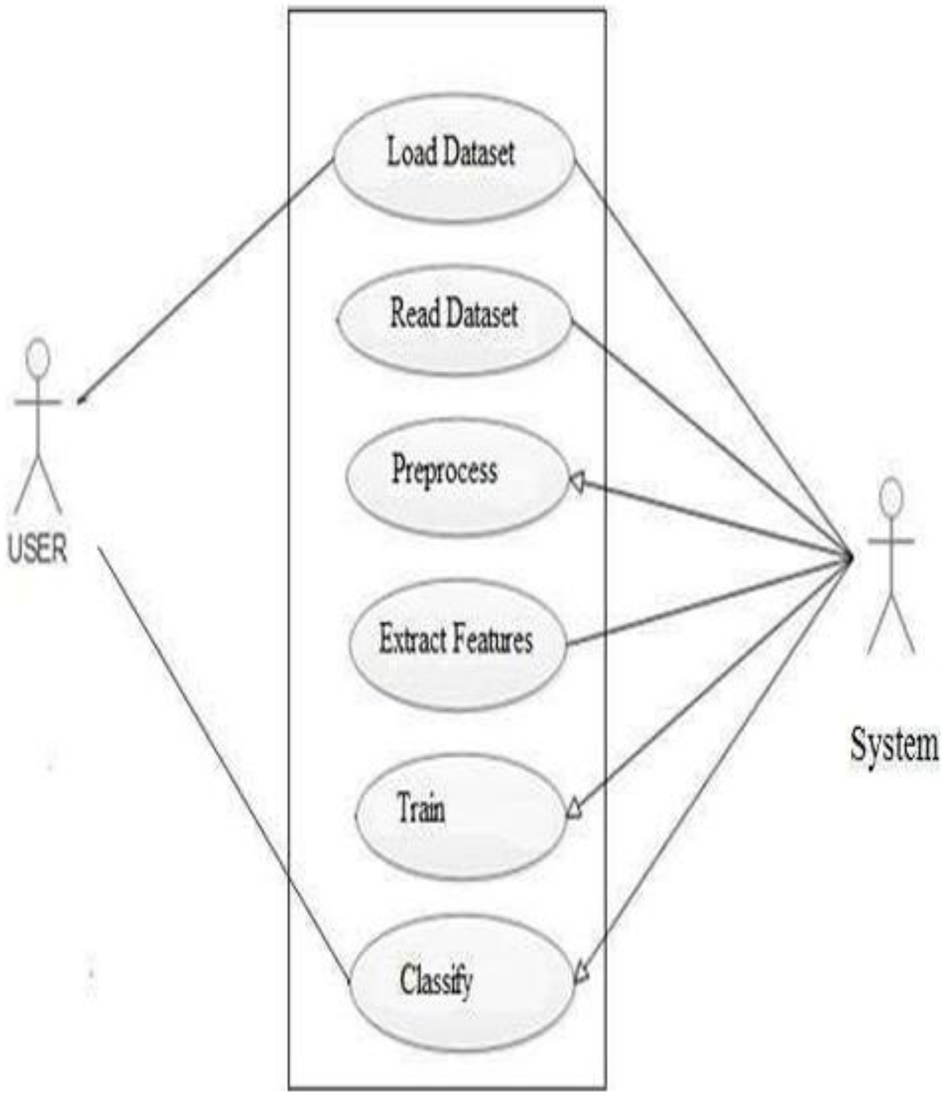


Figure 3.7: *use case*

3.2.6: Activity Diagrams

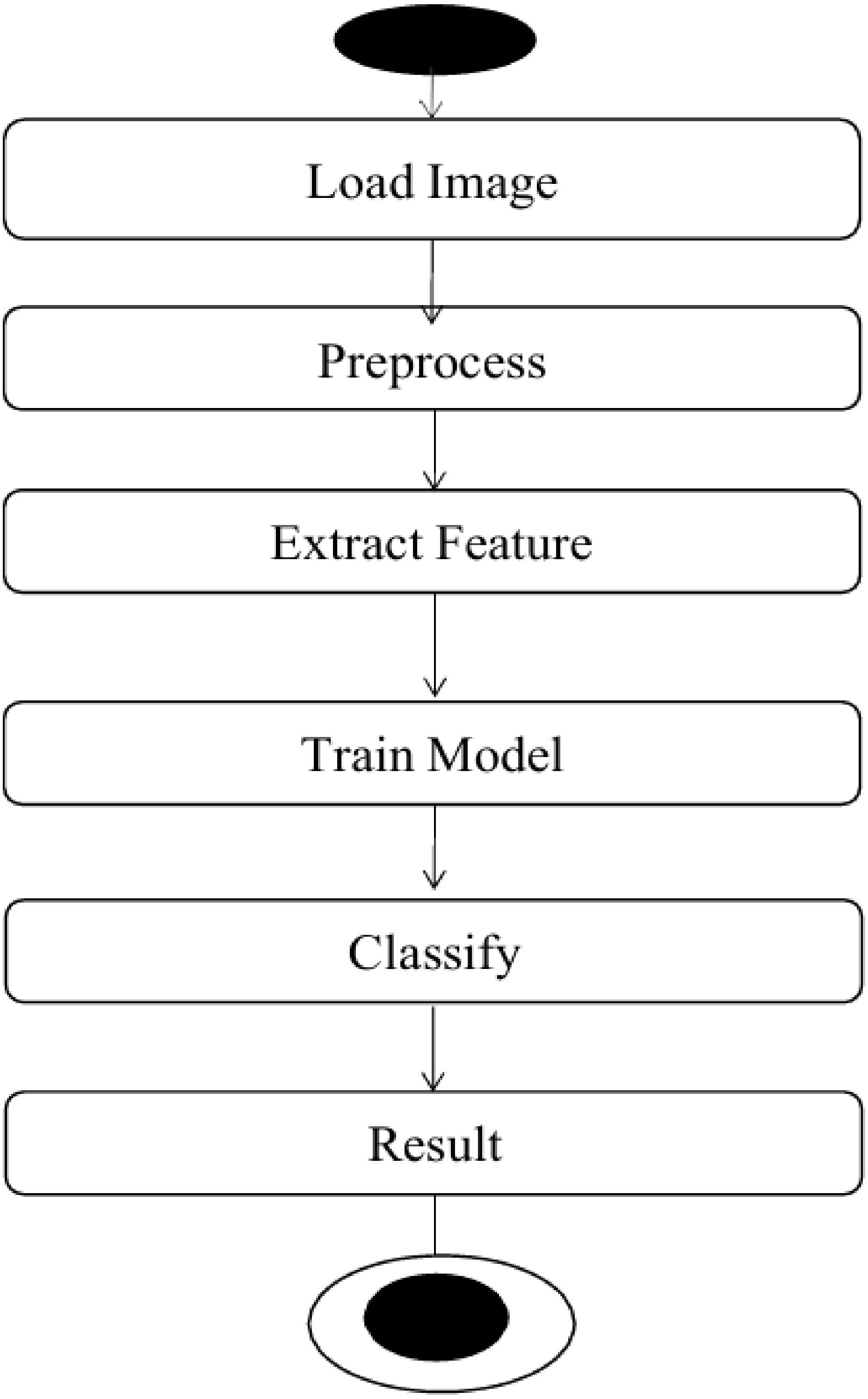


Figure 3.8: *activity diagram*

3.2.7: Deployment Diagrams

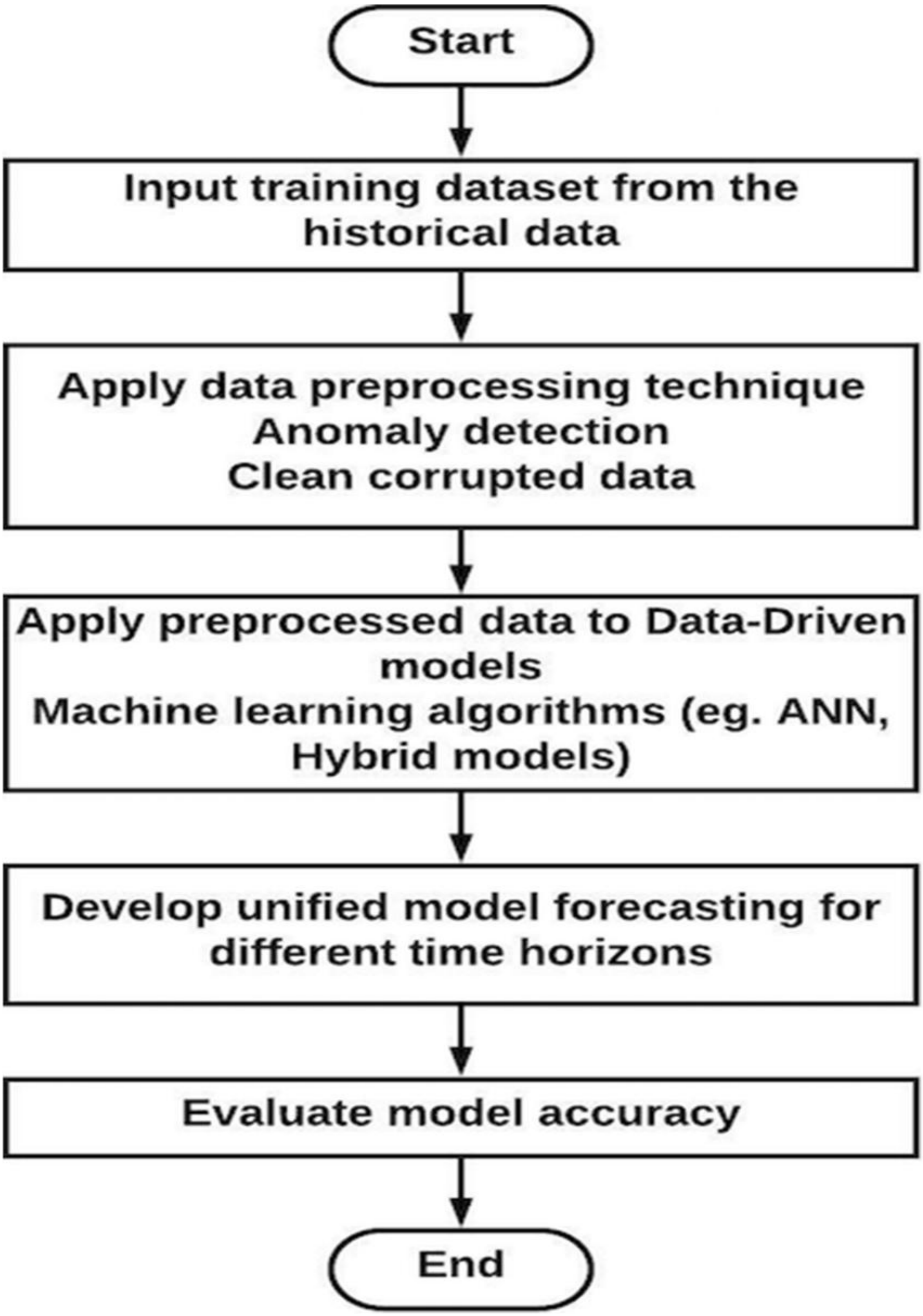


Figure 3.9: *deployment diagram*

CNN

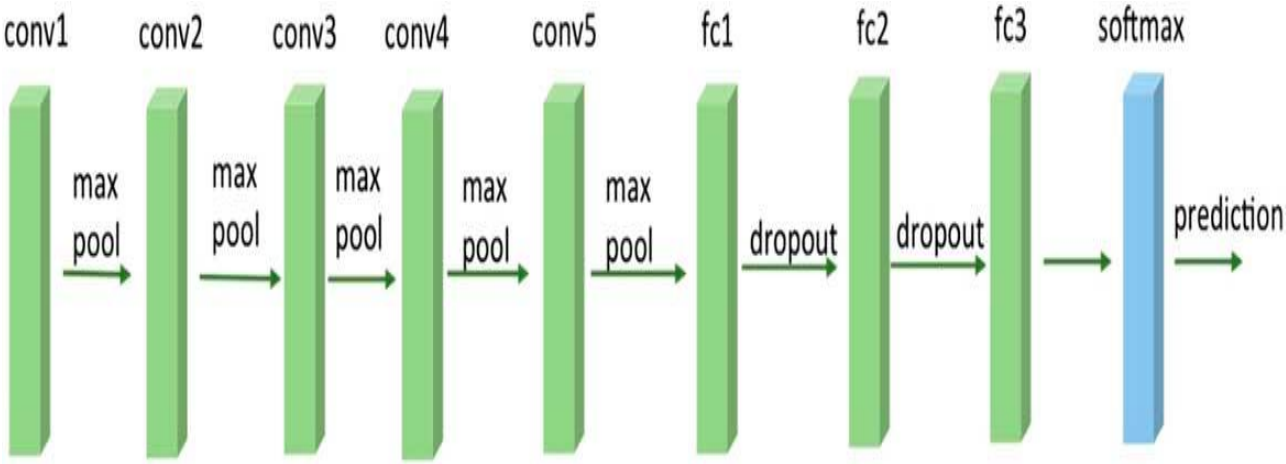


***Figure 3.10**: CNN design*

3.2.8: E-R Diagram

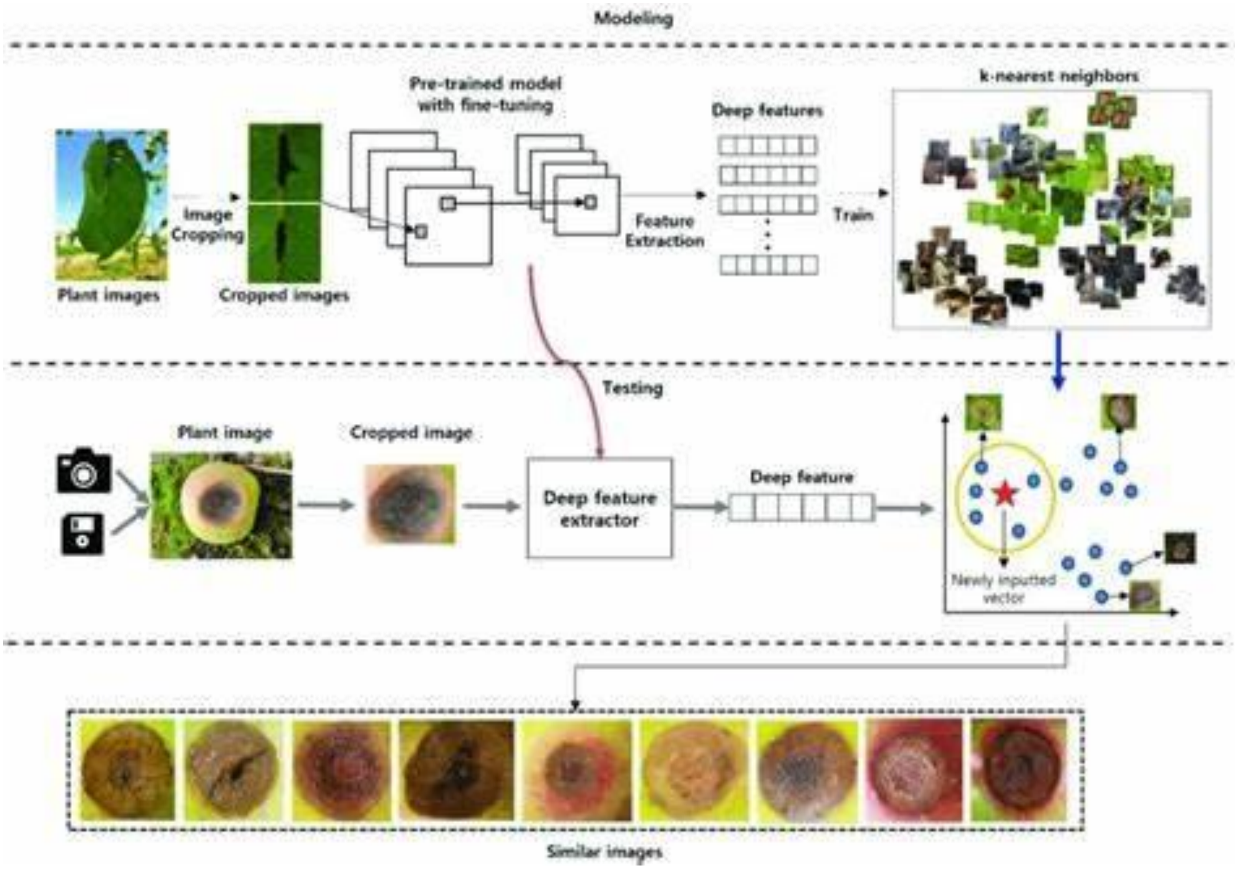
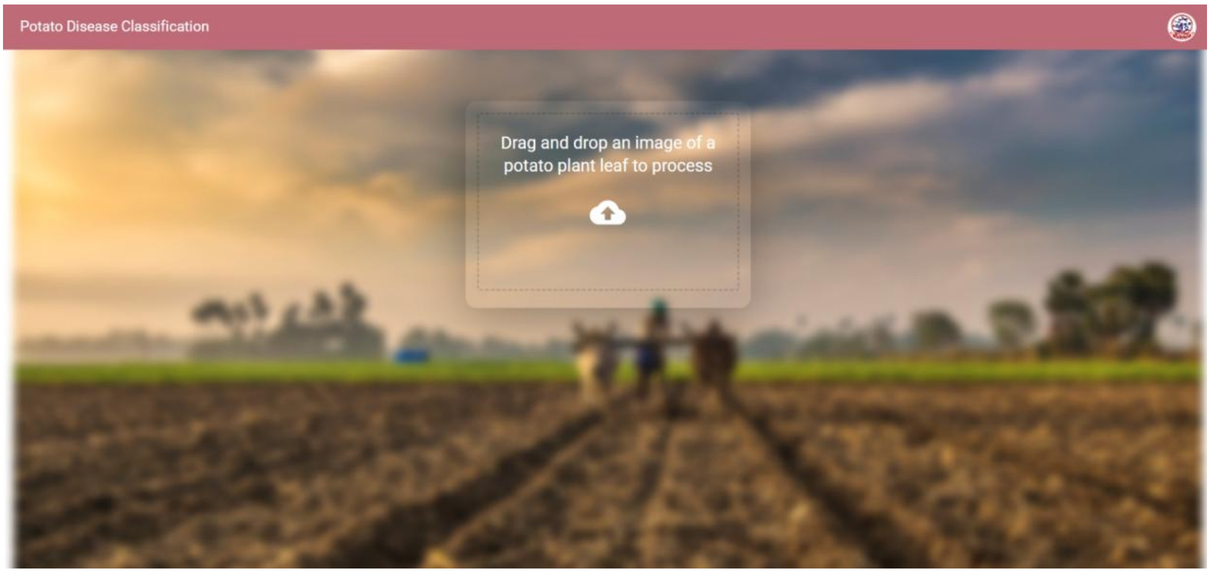


Figure 3.11

3.3 User Interface Design



3.4 Methodology

Karen Simonyan and Andrew Zisserman proposed the idea of the VGG network in 2013 and submitted the actual model based on the idea in the 2014 ImageNet Challenge. They called it VGG after the department of Visual Geometry Group in the University of Oxford that they belonged to.

So, what was new in this model compared to the top-performing models AlexNet-2012 and ZFNet-2013 of the past years? First and foremost, compared to the large receptive fields in the first convolutional layer, this model proposed the use of a very small 3 x 3 receptive field (filters) throughout the entire network with the stride of 1 pixel. Please note that the receptive field in the first layer in AlexNet was 11 x 11 with stride 4, and the same was 7 x 7 in ZFNet with stride 2[5].

The idea behind using 3 x 3 filters uniformly is something that makes the VGG stand out. Two consecutive 3 x 3 filters provide for an effective receptive field of 5 x 5[8]. Similarly, three 3 x 3 filters make up for a receptive field of 7 x 7. This way, a combination of multiple 3 x 3 filters can stand in for a receptive area of a larger size.

But then, what is the benefit of using three 3 x 3 layers instead of a single 7 x 7 layer? Isn't it increasing the no. of layers, and in turn, the complexity unnecessarily? No. In addition to the three convolution layers, there are also three non-linear activation layers instead of a single one you would have in 7 x 7[1]. This makes the decision functions more discriminative. It would impart the ability to the network to converge faster.

Secondly, it also reduces the number of weight parameters in the model significantly. Assuming that the input and output of a three-layer 3 x 3 convolutional stack have C channels, the total number of weight parameters will be 3 * 32 C2 = 27 C2. If we compare this to a 7 x 7 convolutional layer, it would require 72 C2 = 49 C2, which is almost twice the 3 x 3 layers [3]. Additionally, this can be seen as a regularization on the 7 x 7 convolutional filters forcing them to have a decomposition through the 3 x 3 filters, with, of course, the non-linearity added in-between by means of ReLU activations. This would reduce the tendency of the network to over-fit during the training exercise.

Another question is – can we go lower than 3 x 3 receptive size filters if it provides so many benefits? The answer is N o.‖ 3 x 3 is considered to be the smallest size to capture the notion of left to right, top to down, etc. So, lowering the filter size further could impact the ability of the model to understand the spatial features of the image [2].

The consistent use of 3 x 3 convolutions across the network made the network very simple, elegant, and easy to work with.

3.4.1: VGG Configurations

The authors proposed various configurations of the network based on the depth of the network. They experimented with several such configurations, and the following ones were submitted during the ImageNet Challenge.

A stack of multiple (usually 1, 2, or 3) convolution layers of filter size 3 x 3, stride one, and padding 1, followed by a max-pooling layer of size 2 x 2, is the basic building block for all of these configurations. Different configurations of this stack were repeated in the network configurations to achieve different depths. The number associated with each of the configurations is the number of layers with weight parameters in them.

The convolution stacks are followed by three fully connected layers, two with size 4,096 and the last one with size 1,000. The last one is the output layer with SoftMax activation. The size of 1,000 refers to the total number of possible classes in ImageNet.

VGG16 refers to the configuration ―D‖ in the table listed below. The configuration ―C‖ also has 16 weight layers. However, it uses a 1 x 1 filter as the last convolution layer in stacks 3, 4, and 5. This layer was used to increase the non-linearity of the decision functions without affecting the receptive field of the layer.

3.4.2: VGG 16 Architecture

Of all the configurations, VGG16 was identified to be the best performing model on the ImageNet dataset. Let's review the actual architecture of this configuration.

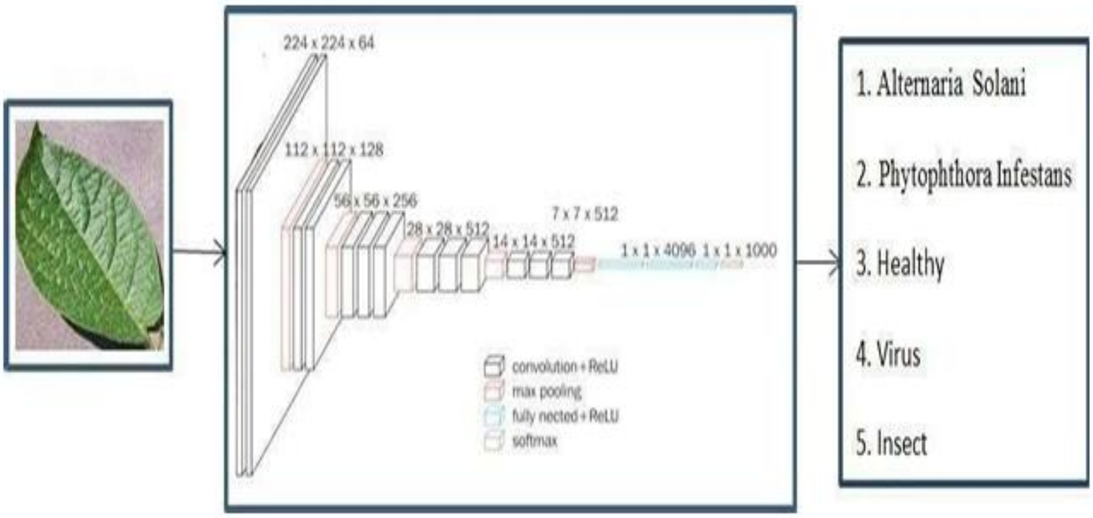


Figure 3.12: *VGG-16 Architecture*

The input to any of the network configurations is considered to be a fixed size 224 x 224 image with three channels – R, G, and B. The only pre-processing done is normalizing the RGB values for every pixel. This is achieved by subtracting the mean value from every pixel.

Image is passed through the first stack of 2 convolution layers of the very small receptive size of 3 x 3, followed by ReLU activations. Each of these two layers contains 64 filters. The convolution stride is fixed at 1 pixel, and the padding is 1 pixel. This configuration preserves the spatial resolution, and the size of the output activation map is the same as the input image dimensions. The activation maps are then passed through spatial max pooling over a 2 x 2-pixel window, with a stride of 2 pixels. This halves the size of the activations. Thus, the size of the activations at the end of the first stack is 112 x 112 x 64[1].

The activations then flow through a similar second stack, but with 128 filters as against 64 in the first one. Consequently, the size after the second stack becomes 56 x 56 x 128. This is followed by the third stack with three convolutional layers and a max pool layer. The no. of filters applied here are 256, making the output size of the stack 28 x 28 x 256. This is followed by two stacks of three convolutional layers, with each containing 512 filters. The output at the end of both these stacks will be 7 x 7 x 512[1].

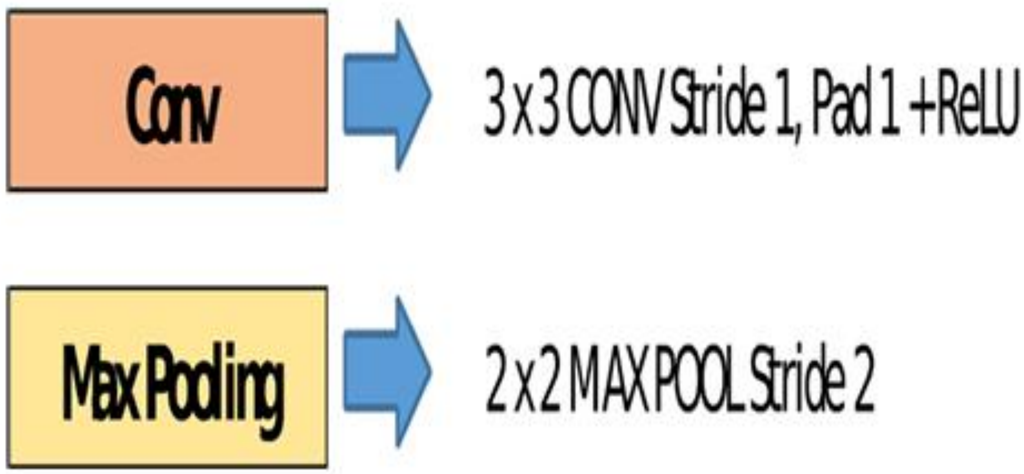


Figure 3.13: *conv MaxPoding*

The stacks of convolutional layers are followed by three fully connected layers with a flattening layer in-between. The first two have 4,096 neurons each, and the last fully connected layer serves as the output layer and has 1,000 neurons corresponding to the 1,000 possible classes for the ImageNet dataset. The output layer is followed by the Softmax activation layer used for categorical classification.

3.4.3: CNN Model

CNNs thinks about piece by piece of picture [1]. The pieces that CNN looks for are called highlights. It finding the harsh element matches in two pictures in similar positions, CNNs improve at seeing closeness than entire picture coordinating plans. Each component resembles a smaller than normal picture, a little two-dimensional cluster of qualities [1].

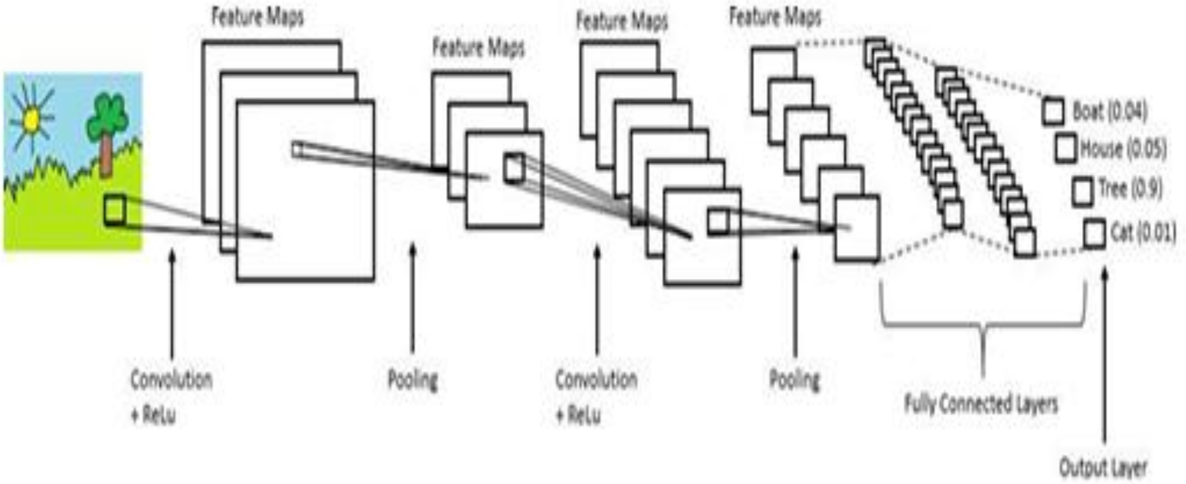


Figure 3.14: *CNN Architecture*

We first create a sequel model of CNN with 7 levels. We used Adam Optimizer to measure performance error and tune cross entropy. We then use transfer learning to create the model. To do that we only use built in module keras, keras applications that provide pre-trained weights. We modified the Deep Learning model so that the pre-trained weights matched our desired output dimension by dropping the last few layers and adding few Junne lavens[1].

• Give input photograph into convolution layer

• Choose boundaries, apply channels with steps, cushioning if requires. Perform Convolution on the picture and apply ReLU enactment to the grid.

• Execute pooling to decrease Dimension size.

• Add as numerous convolutional layers until satisfied.

• Flatten the yield and feed into a completely associated layer.

• Output the class utilizing enactment and order pictures.

Chapter 4. Implementation and Testing

• Data Collection

• Pre-processing

• Building and Training Model

• Classification of Disease

Data Collection

We collected this information from kaggle website. We were able to collect data on about 3 types of diseases of potato leaves.

• Phytopthora Infestans

• Alternaria Solani

• Healthy potato leaves

• Virus

• Insect.

Preprocessing

In this project we have to take four types of image processing steps to normalize the image, change the color of the image, and identify the properties, Image processing such as filtering and transformation of the image. We have used Python's Opencv Library for this purpose. The features of the OpenCV library are:

• Read & write images

• Capture and save the images

• Image-processing such as filtering and transformation

• Detection the feature image or picture object detection

The picture document is perused with the OpenCV work the request for colors is BGR. Then again, in Pillow, the request for colors is thought to be RGB.

Building and Training

CNNs thinks about piece by piece of picture. The pieces that CNN looks for are called highlights. It finding the harsh element matches in two pictures in similar positions, CNNs improve at seeing closeness than entire picture coordinating plans. Each component resembles a smaller than normal picture, a little two-dimensional cluster of qualities.

• Give input photograph into convolution layer Choose boundaries,

• apply channels with steps, cushioning if requires.

• Perform Convolution on the picture and apply ReLU enactment to the grid.

• Execute pooling to decrease Dimension size. Add as numerous convolutional layers until satisfied.

• Flatten the yield and feed into a completely associated layer. Output the class utilizing enactment and order pictures.

Classification of Disease

By applying the CNN-VGG-16 model our system will automatically classify the potato disease in the input image.

4.1 Introduction to Languages

4.1.1 Python technology

Python is an interpreted, object- oriented programming language similar to PERL, that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of operating systems, including UNIX- based systems, Mac OS, MS- DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands, whose favourite comedy group at the time was Monty Python's Flying Circus. The source code is freely available and open for modification and reuse. Python has a significant number of users. Technology A notable feature of Python is its indenting of source statements to make the code easier to read. Python offers dynamic data type, ready-made class, and interfaces to many system calls and libraries. It can be extended, using the C or C++language. Python can be used as the script in Microsoft's Active Server Page (ASP) technology. The scoreboard system for the Melbourne (Australia) Cricket Ground is written in Python. Z Object Publishing Environment, a popular Web application server, is also written in the Python language's.

4.1.2 Python Platform

Apart from Windows, Linux and MacOS, C Python implementation runs on 21 different platforms. Iron Python is a .NET framework based Python implementation and it is capable of running in both Windows, Linux and in other environments where .NET framework is available.

4.1.3 Python Library

Machine Learning, as the name suggests, is the science of programming a computer by which they are able to learn from different kinds of data. A more general definition given by Arthur Samuel is –"Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed." They are typically used to solve various types of life problems. In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formula. This made the process t ime consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- NumPy
- SciPy
- Scikit-Learn
- TensorFlow
- Kera's
- Pytorch
- Pandas

• Matplotlib

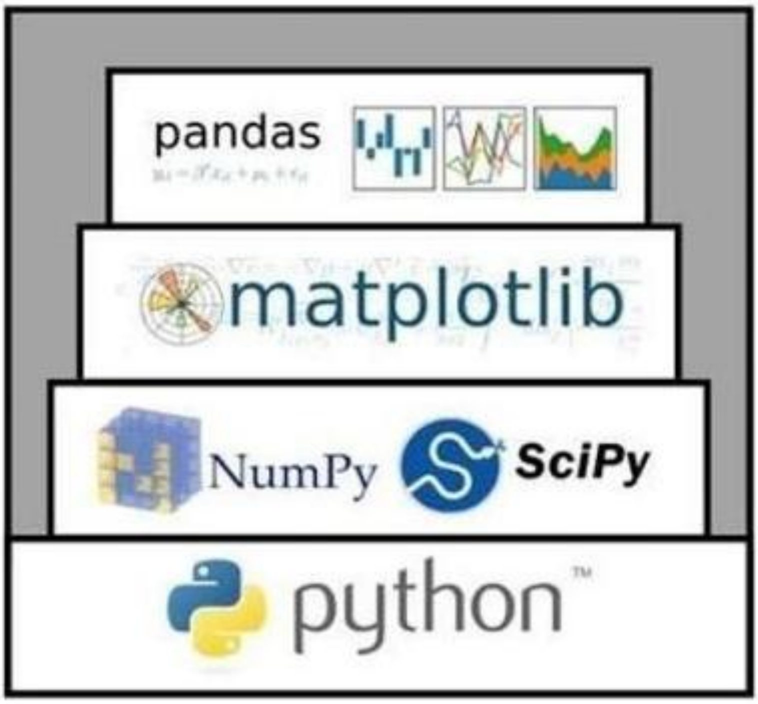


Figure 4.1: python library

Figure4.2 python library 2

4.1.4 Anaconda python installation OF IDE

Step1:

open any web browser for example open google chrome type anaconda download

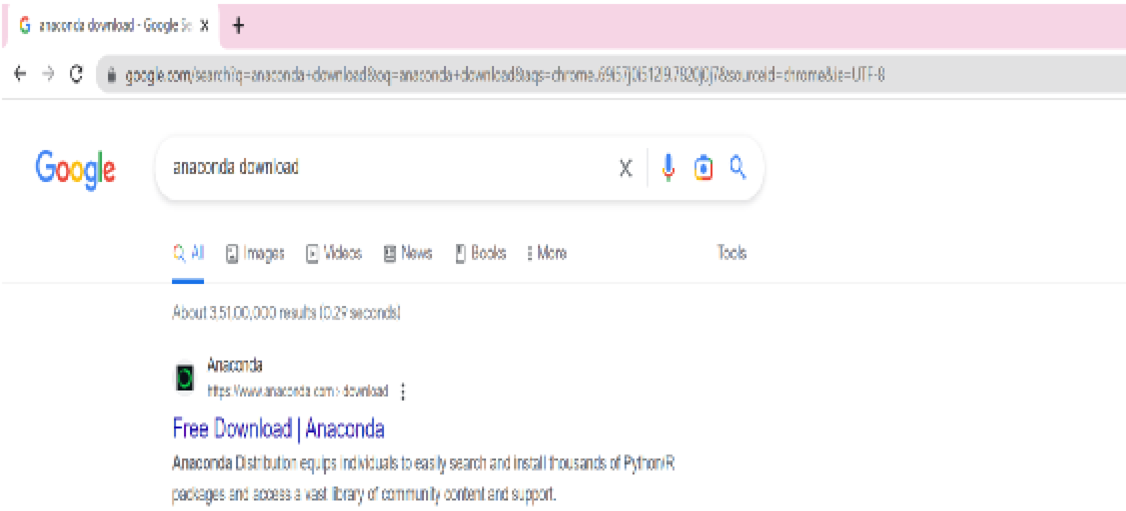


Figure4.3 anaconda installation

Step2:

Click on free download like shows above figure, then the page will be redirected to shown as below figure.

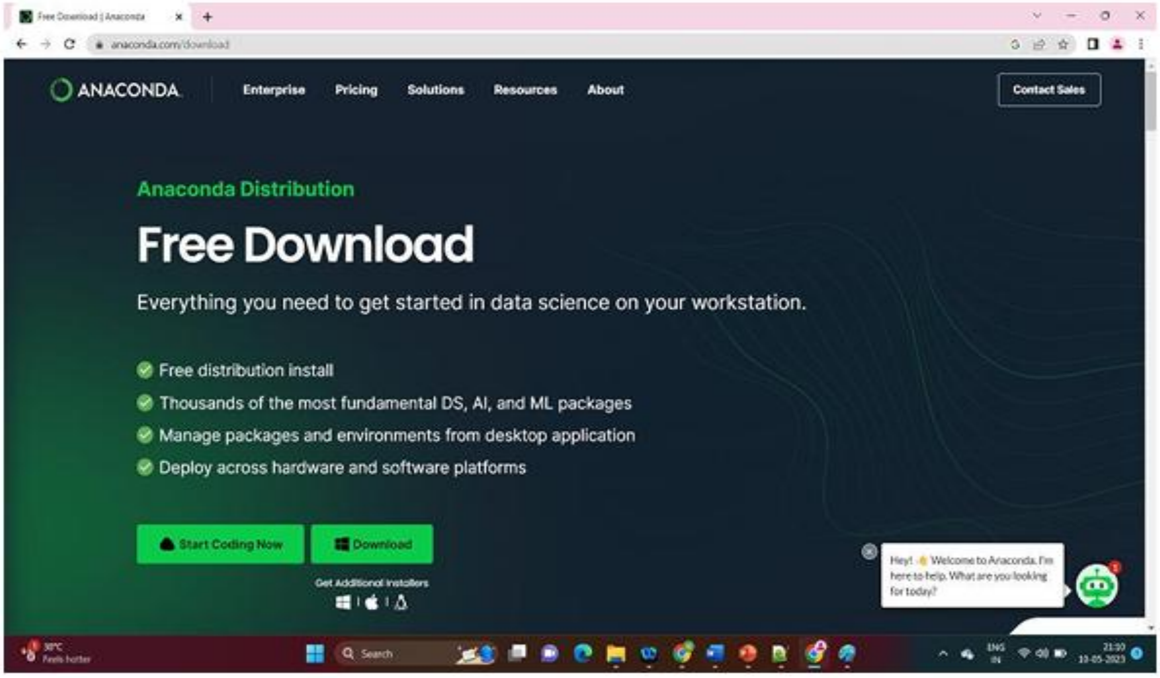
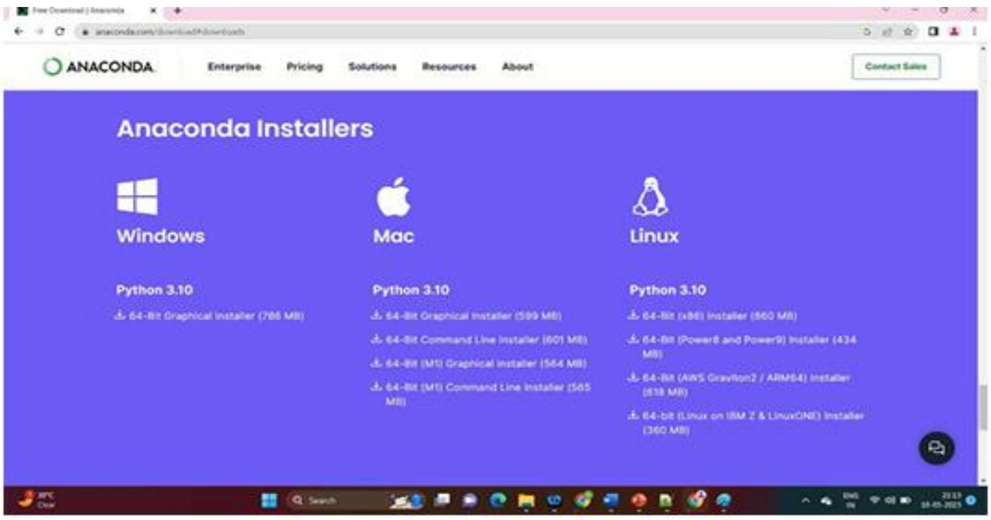


Figure 4.4 anaconda installation

Step3:

Click on download button like shown above figure it will be redirected to download page its show as below figure.



Step4:

we can see anaconda exe file downloading

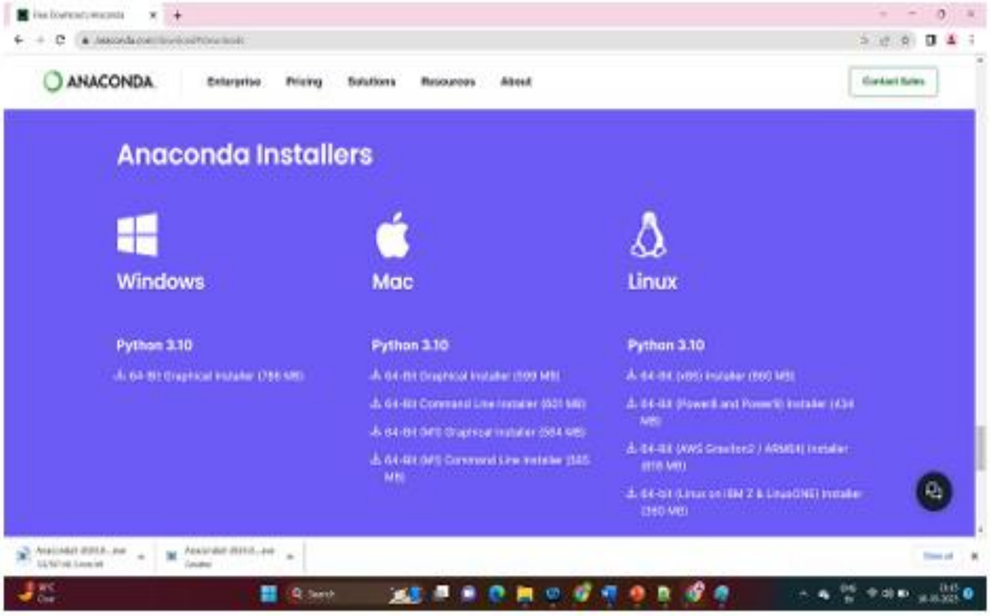


Figure 4.6 anaconda installation

Step-5:

now you can open the exe file to install anaconda, the exe file will be in downloads

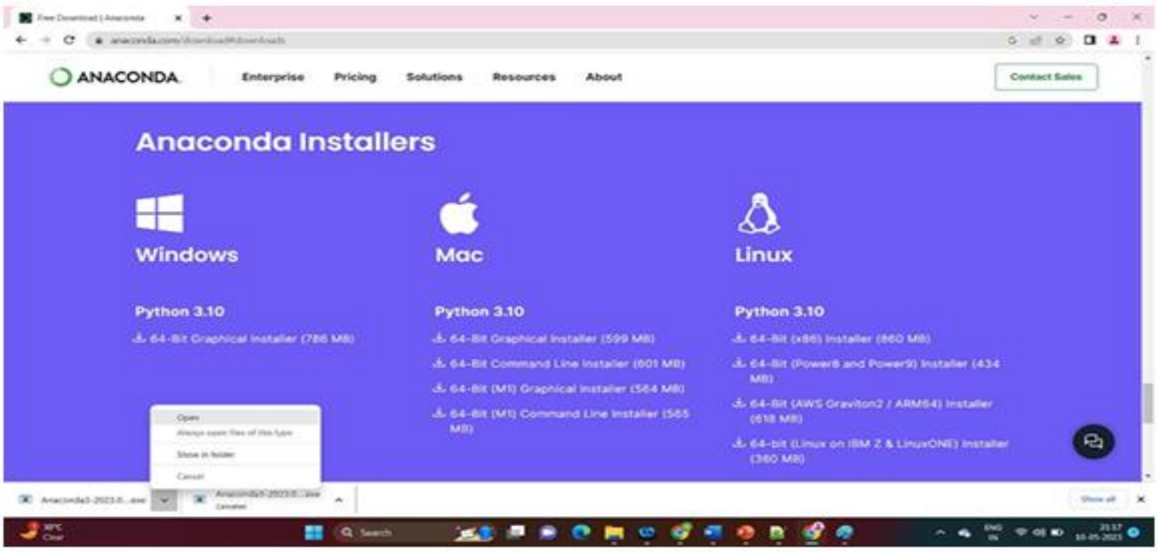


Figure 4.7 anaconda installation

Step-6:

when you open the exe file a dialogue box will open like below figure, click on next

Figure 4.8 anaconda installation

Step-7:

When you click on next it will be redirected to another dialogue box like shown below figure

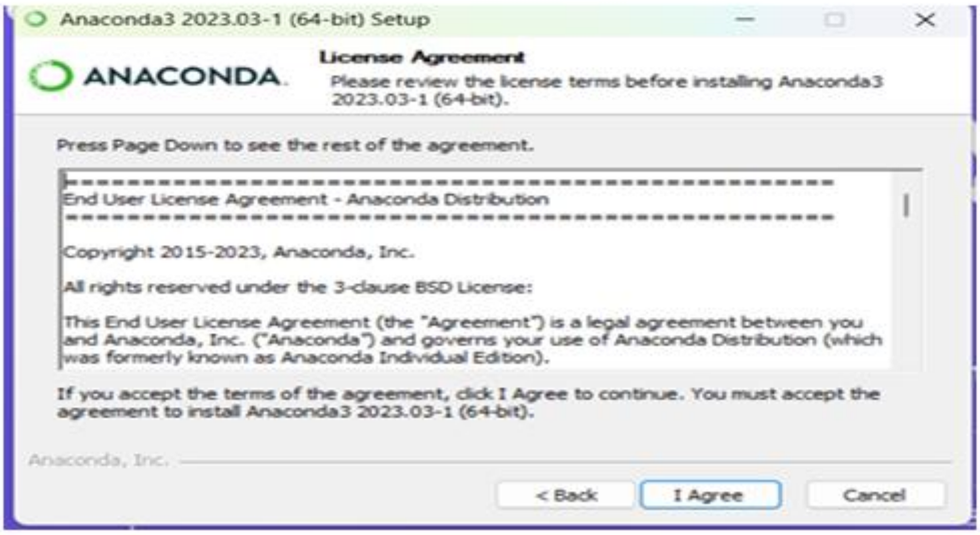


Figure 4.9 anaconda installation

Step-8:

Click on I agree then select just me(recommend) and click on the next button

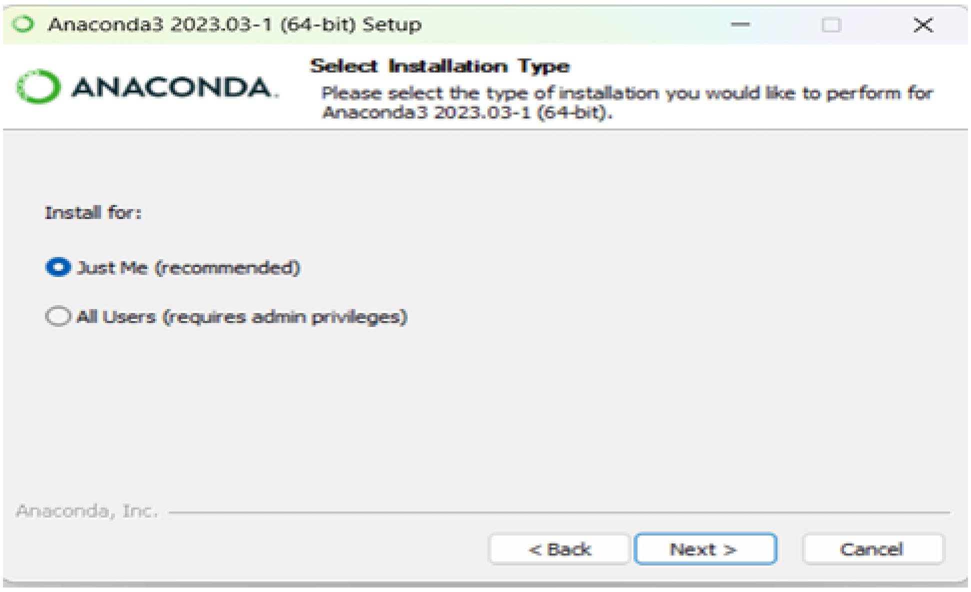


Figure 4.10

Step-9:

if you want change the destination folder or keep it is as it is generally it takes c:\\users\\anaconda3

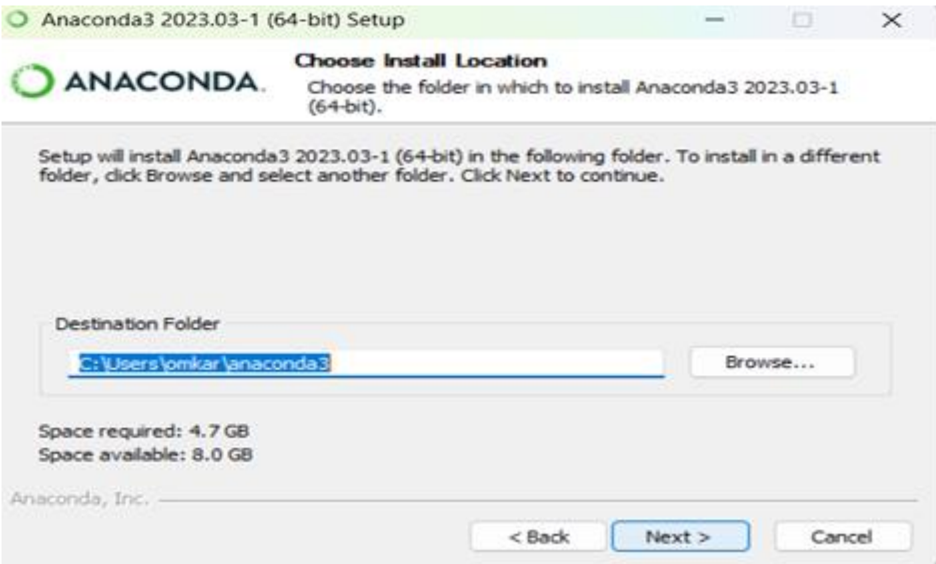


Figure 4.11

Step-10:

Just click on install button

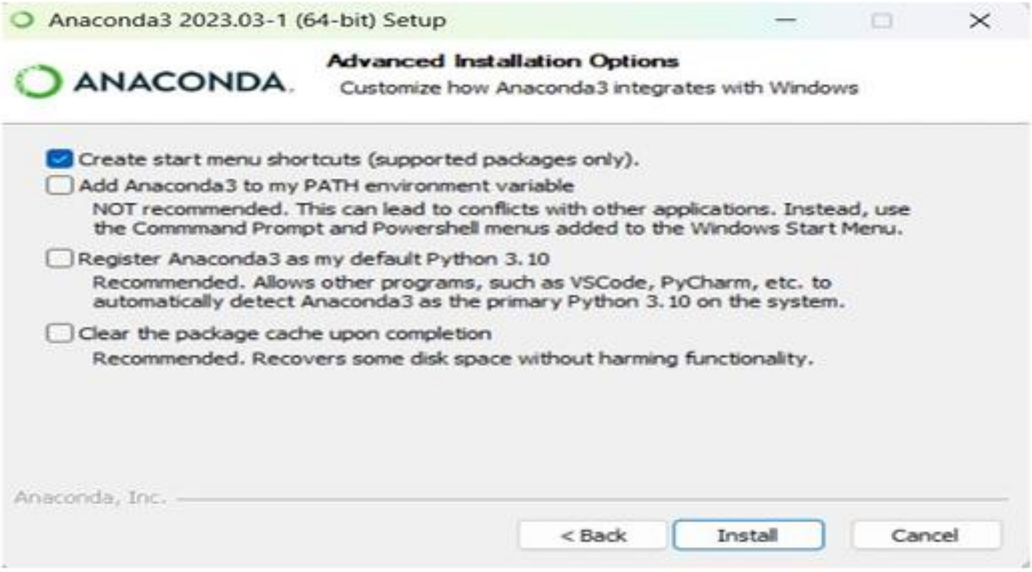


Figure 4.12

Step-11:

Just wait for some time till the green line complete

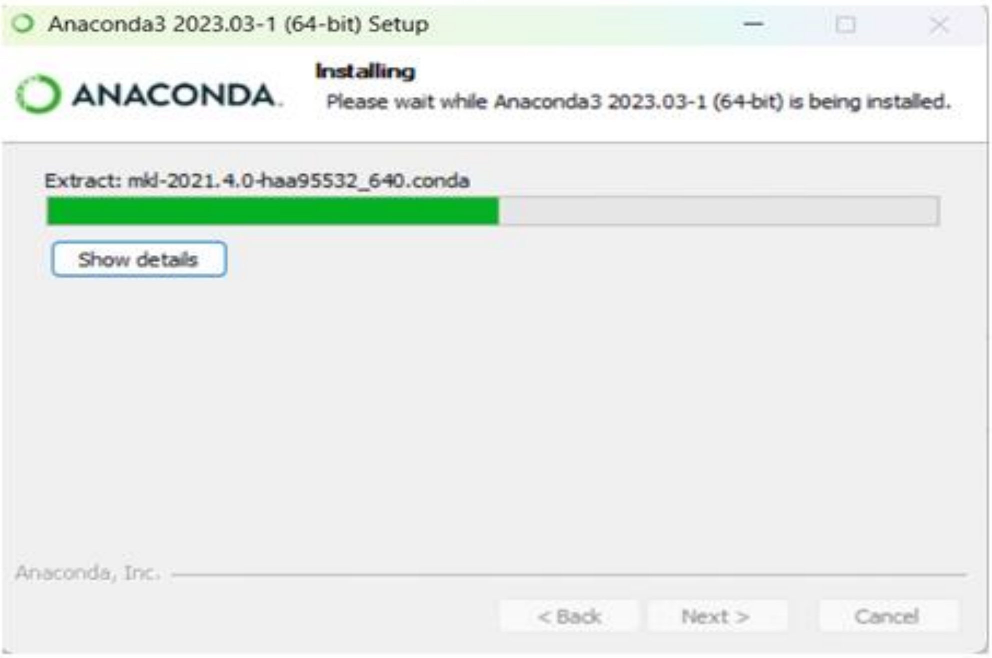


Figure 4.13

Step-12:

Below figure shows how the installation complete, then click on next

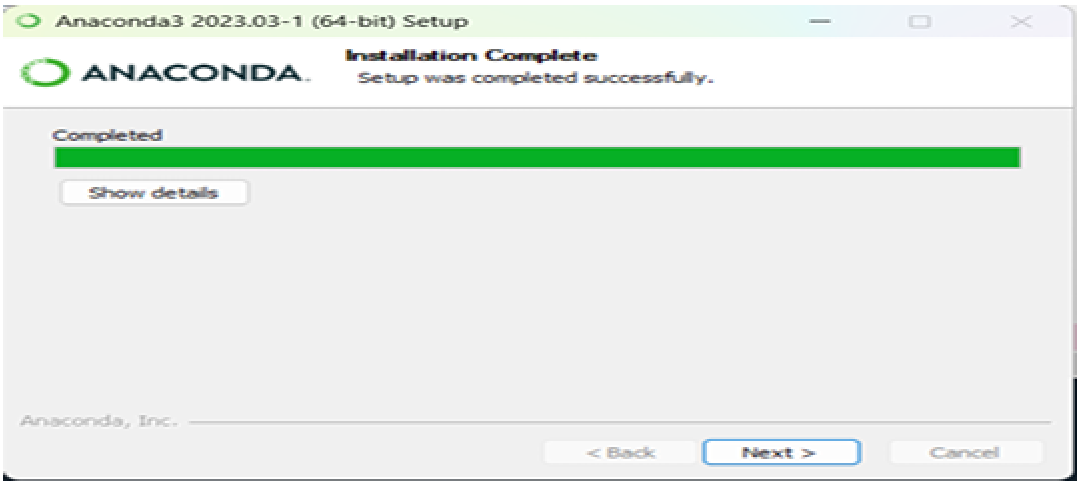


Figure 4.14

Step-13:

Click on next

Figure 4.15

Step-14:

Click on finish

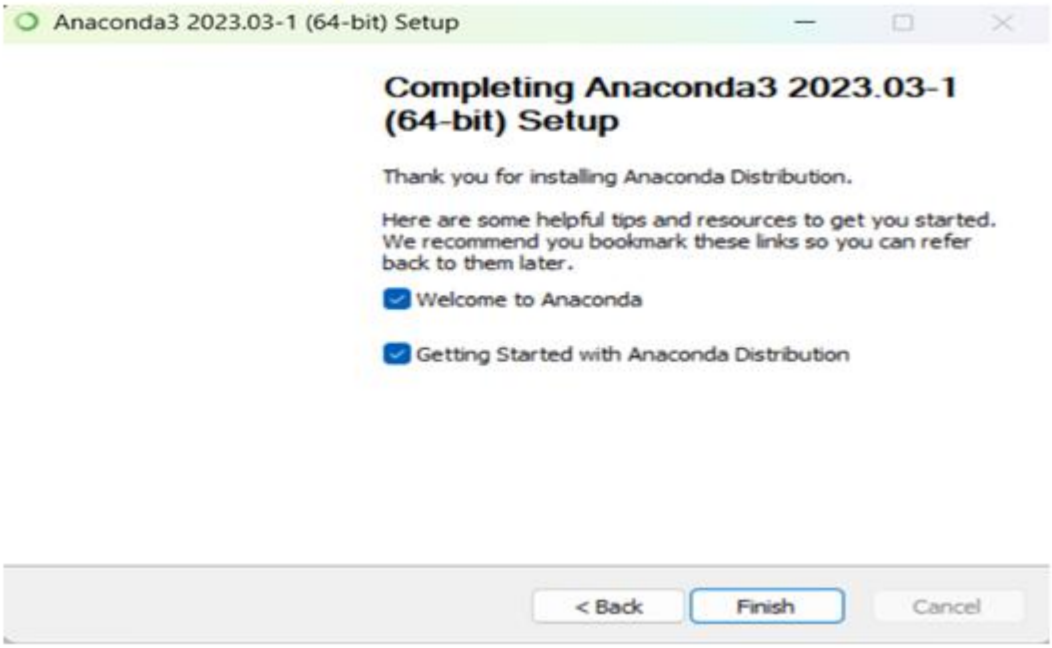


Figure 4.16

Step-16:

Search Anaconda navigator in search bar, click on open

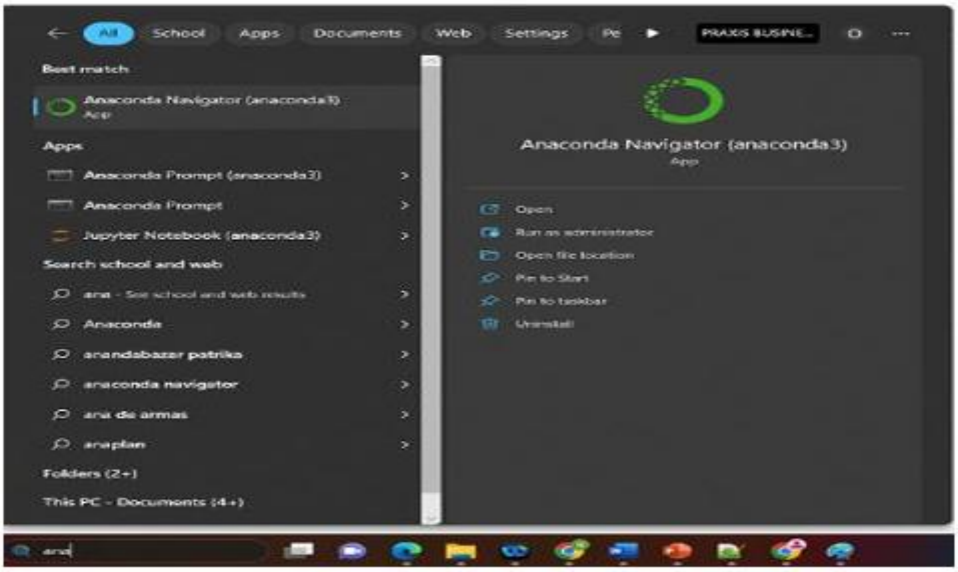


Figure 4.17

Step-17:

It will take few minutes to open, while opening you can see a anaconda initialization symbol like below

Figure 4.17

Step-18:

Ones it is opened you can see the anaconda initialization symbol in task bar, click on that you will see a anaconda navigator application, Next search for Jupyter note book click on launch

Note**:

If under jupyter note book if you see install, then first install it. Then you will see launch

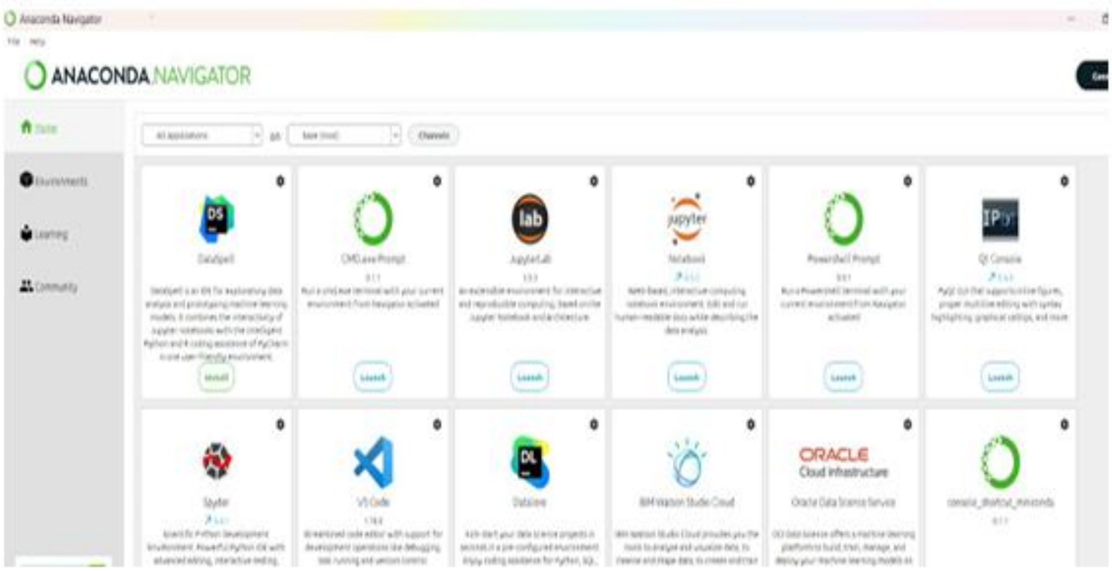


Figure 4.18 anaconda installation

Step-19:

Ones you launch jupyter notebook , a jupyter interface will open web browser either edge or chrome

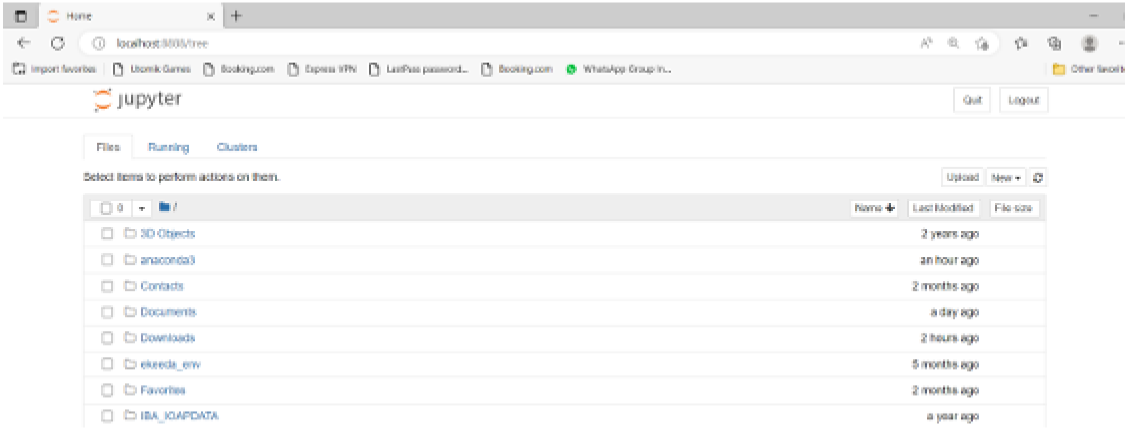


Figure 4.19 jupyter notebook

Step-20:

Click on new icon it's on top right corner like shown in above figure, a new == > Python 3

A python notebook will open

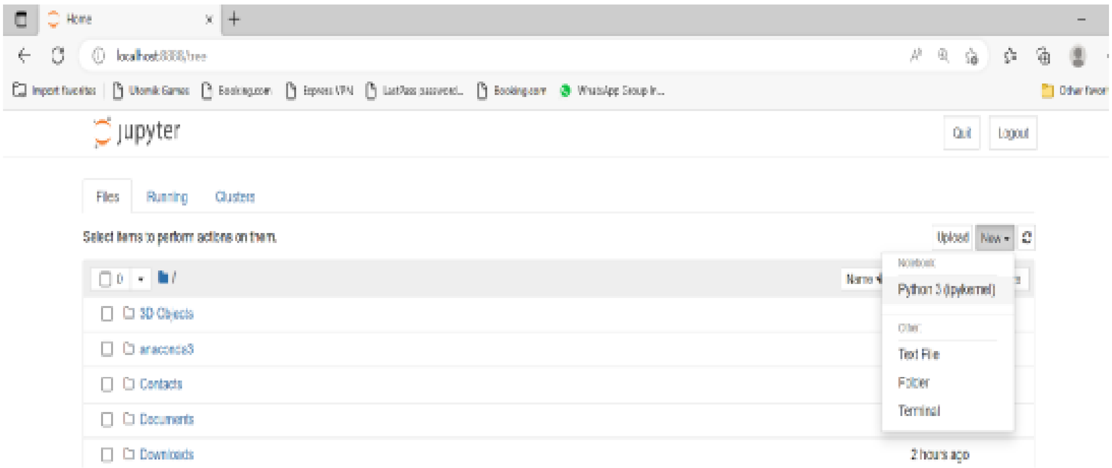


Figure 4.20 python notebook

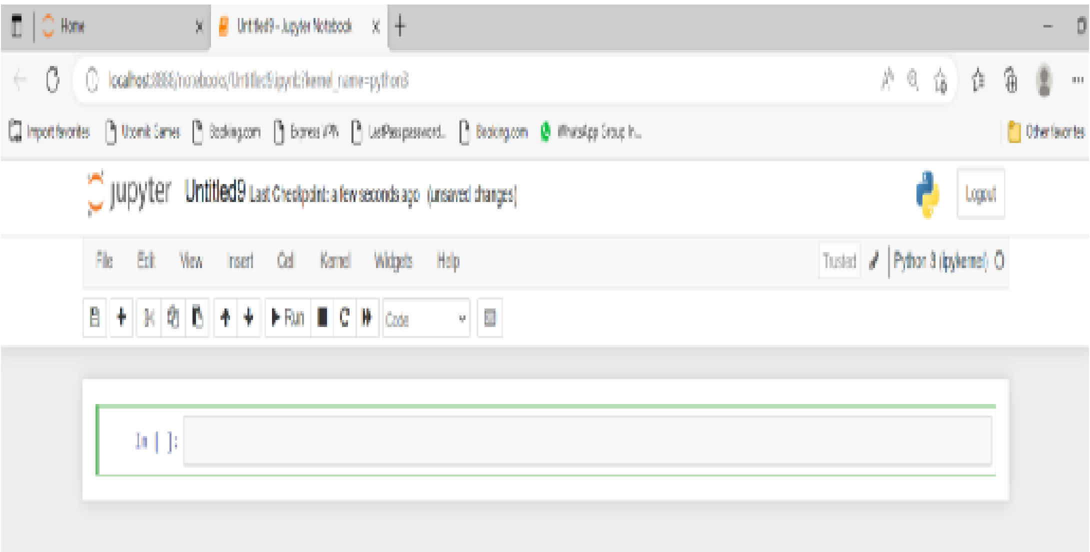


Figure 4.21 jupyter notebook

4.1.5: Tools

4.1.5.1 NumPy

NumPy is a very popular python library for large multi- dimensional array and matrix processing, with the help of a large collection of high- level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High- end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

Ex:-

Creating Array

import numpy as np

a = np.array([10, 20, 30])

b = np.array([1, 77, 2, 3]) print(a[0])

print(b[2])

Creating a Multi-Dimensional Array

```
import numpy as np

a = np.array([

[10, 20, 30],

[40, 50, 60]

])

print(a)
```

4.1.5.2 SciPy

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

4.1.5.3 TensorFlow

TensorFlow is a very popular open- source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, TensorFlow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

4.1.5.4 Kera's

Kera's is a very popular Machine Learning library for Python. It is a high- level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Kera's makes it really for ML beginners to build and design a Neural Network. One of the best thing about Kera's is that it allows for easy and fast prototyping.

4.1.5.5 Pandas

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high- level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

4.1.5.6 Matplotlib

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats, etc.

4.2 Algorithm/Pseudocode used

For potato disease classification, common approaches use machine learning (ML) and deep learning (DL). Below is a general pseudocode for a deep learning-based classification approach using Convolutional Neural Networks (CNNs).

4.2.1 Algorithm

CNN (Convolutional Neural Network) – Most common for image-based classification

Transfer Learning – Using pre-trained models like MobileNet, ResNet, or VGG16

SVM (Support Vector Machine) – Traditional ML approach for feature-based classification

Random Forest & Decision Trees – Sometimes used with extracted features

4.2.2 Pseudocode used

1. Import necessary libraries

2. Load the dataset (images of potato leaves with diseases and healthy ones)

3. Preprocess the dataset:

   a. Resize images to a fixed size

   b. Normalize pixel values

   c. Split data into training, validation, and test sets

   d. Apply data augmentation (rotation, flipping, etc.)

4. Define the CNN model:

   a. Input layer (image input)

   b. Convolutional layers with ReLU activation

   c. Pooling layers (MaxPooling)

   d. Fully connected layers with Softmax activation

5. Compile the model:

   a. Choose an optimizer (Adam, SGD)

   b. Define loss function (categorical cross-entropy)

   c. Set performance metrics (accuracy)

6. Train the model with training data:

   a. Feed images into the network

   b. Adjust weights using backpropagation

   c. Validate using validation data

   d. Monitor accuracy and loss

7. Evaluate the model on test data

8. Deploy the model for real-time predictions

9. Use the model to classify new images

10. Output the predicted disease class

4.3 Testing Techniques:

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components subassemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement

4.3.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results. Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page

4.3.2 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

4.3.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items: Valid Input: identified classes of valid input must be accepted. Invalid Input: identified classes of invalid input must be rejected. Functions: identified functions must be exercised. Output: identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined

4.3.4 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

4.3.5 USER ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

4.4 Test Cases designed for the project work

Table 4.4.1: Unit Testing Case 1

Test Case#	UTC01
Test Name	User input format
Test Description	To test user input values
Input	Potato leaf images
Expected Output	The file should be read by the program and display the image.
Actual Output	The file is read and display contents accordingly
Test Result	Success

Table 4.4.2: Unit Testing Case 2

Test Case#	UTC02
Test Name	User input format
Test Description	To test user input values
Input	Image as null
Expected Output	Show alert messages select image
Actual Output	Show alert messages select image
Test Result	Success

Table 4.4.3: Unit Testing Case 3

Test Case#	UTC03
Test Name	Preprocess
Test Description	To test whether resize frame size
Input	Image
Expected Output	It Should resize image
Actual Output	Resized image
Test Result	Success

Table 4.4.4: Unit Testing Case 4

Test Case#	UTC04
Test Name	Feature extraction
Test Description	To test whether it's is extracting features from image
Input	Resized image
Expected Output	It extract features and pass to vgg16 model
Actual Output	It extracted features and passed to vgg16 model
Test Result	Success

Table 4.4.5: Unit Testing Case 5

Test Case#	UTC05
Test Name	Test case to determine the potato leaf disease
Test Description	To test whether a given statistical model predicts the disease name of given user input.
Input	A preprocessed image
Expected Output	The algorithm should predict the disease name as per historical data collected
Actual Output	The predicted value by algorithm is closer to the specified value in the historical data.
Test Result	Success

Table 4.4.6: Unit Testing Case 6

Test Case#	UTC06
Test Name	Test case for importing valid python libraries
Test Description	To test whether an algorithm to implement congestion nodes works without sklearn and keras models
Input	Import all valid libraries sklearn and keras libraries
Expected Output	An error should be thrown specifying –error importing libraries sklearn and keras libraries‖
Actual Output	An error is thrown
Test Result	Success

Chapter 5. Results and Discussions

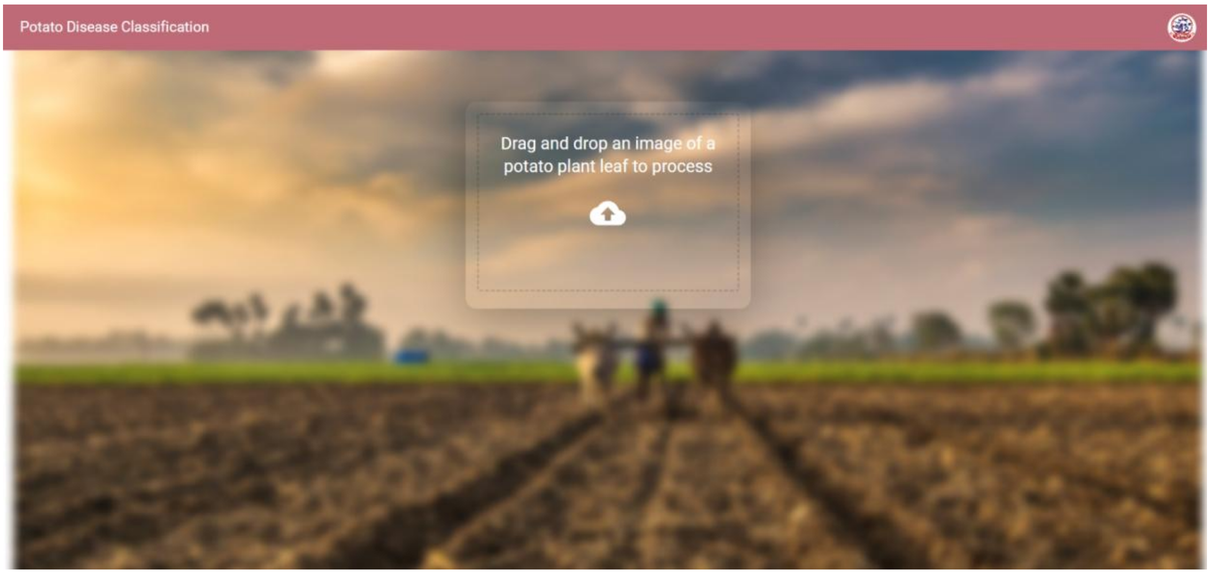


Figure:5.1

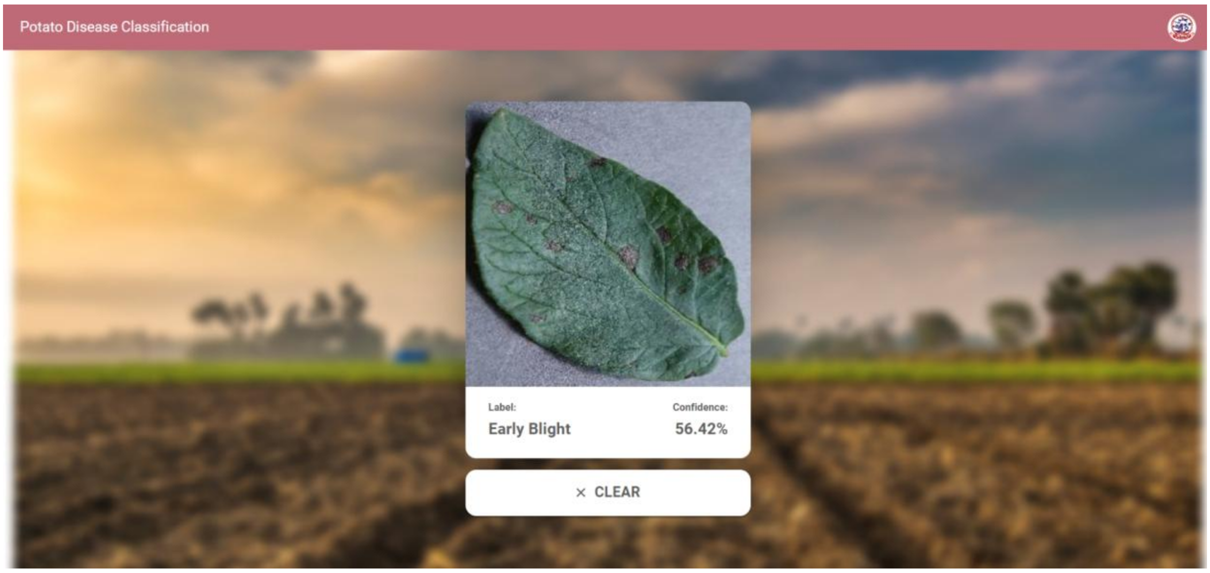


Figure:5.2

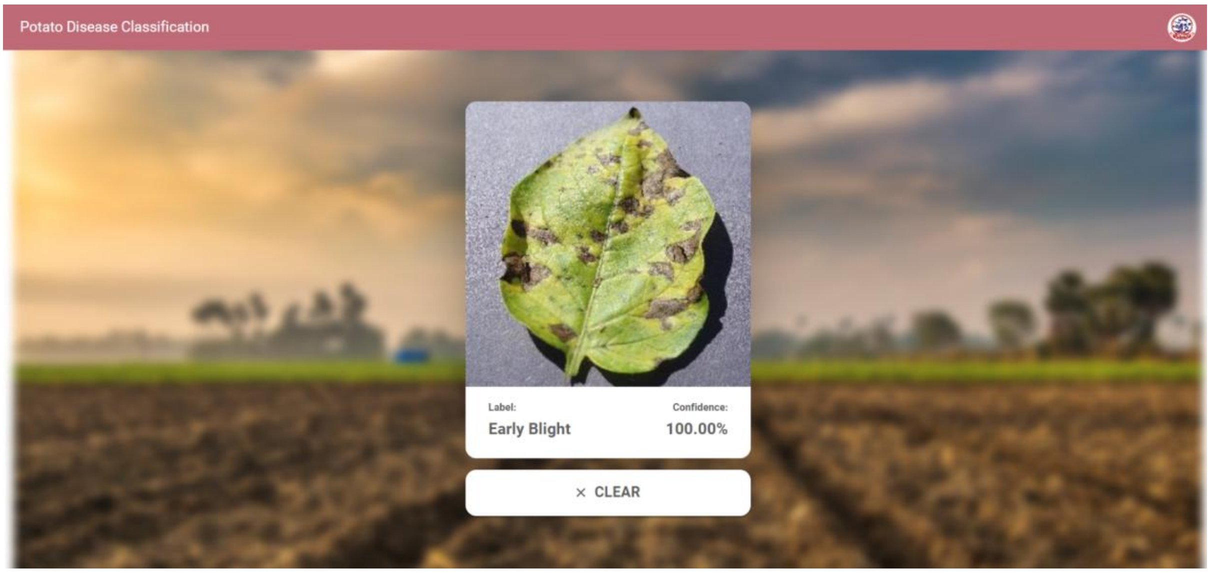


Figure:5.3

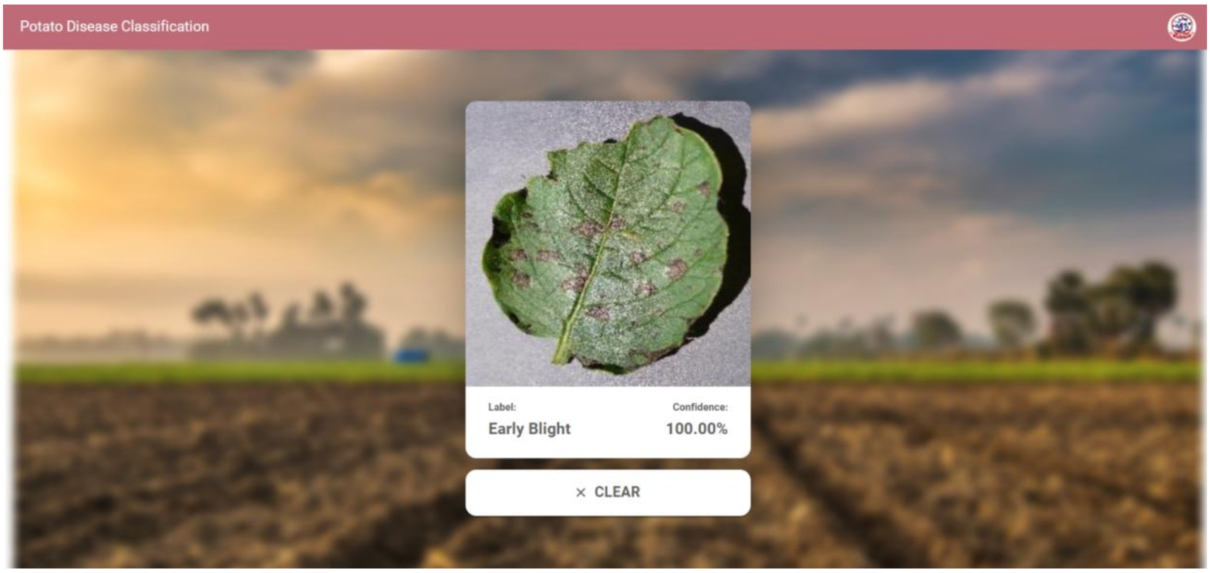


Figure:5.4

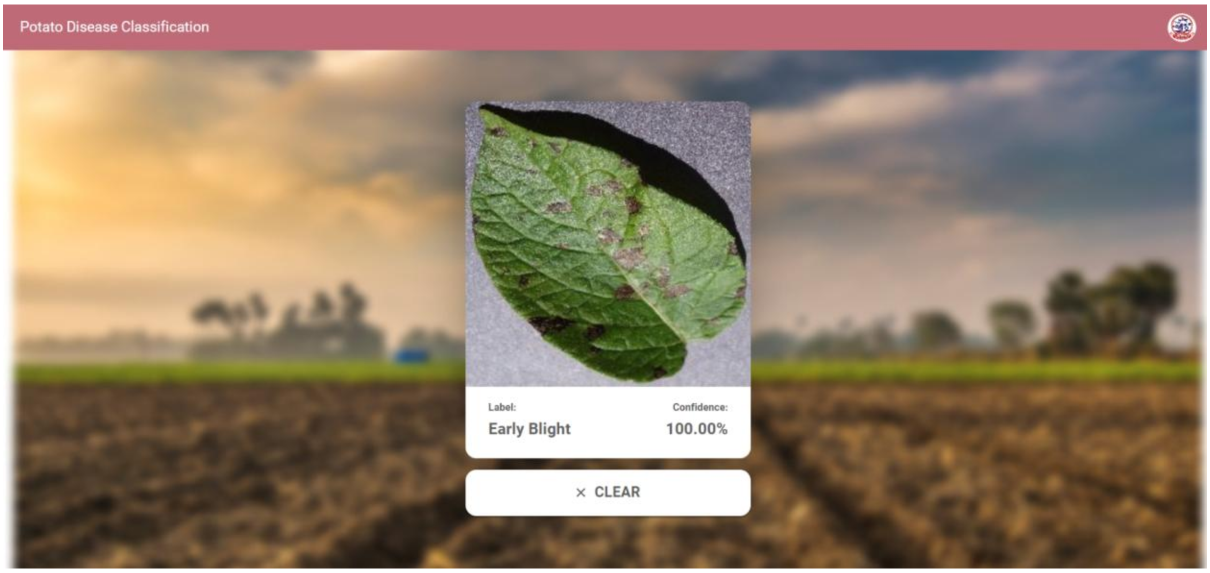


Figure:5.6

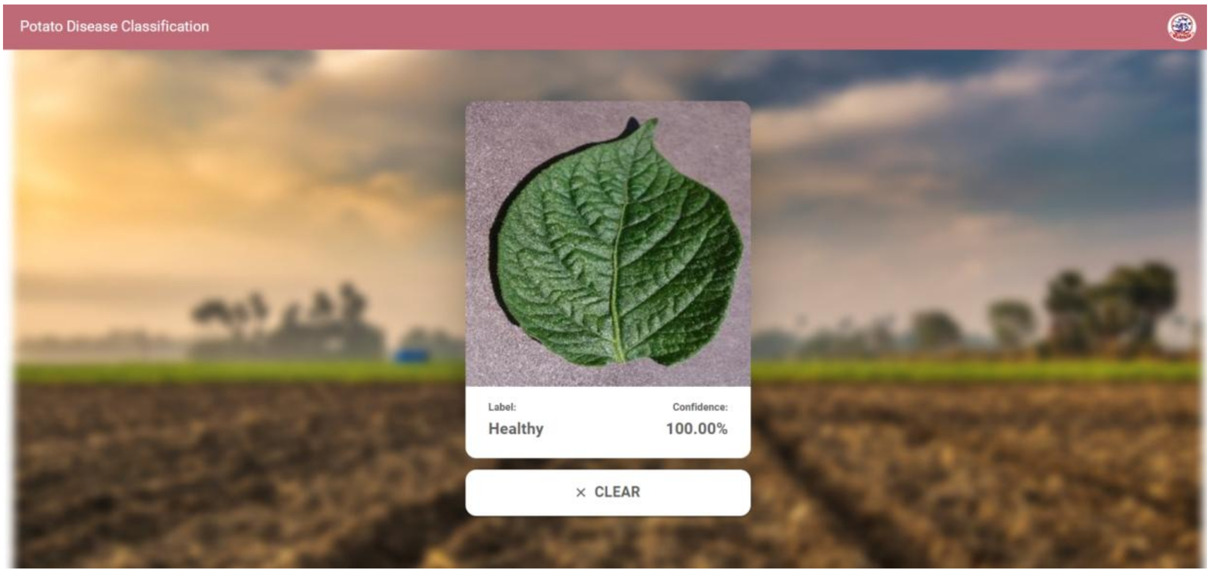


Figure:5.7

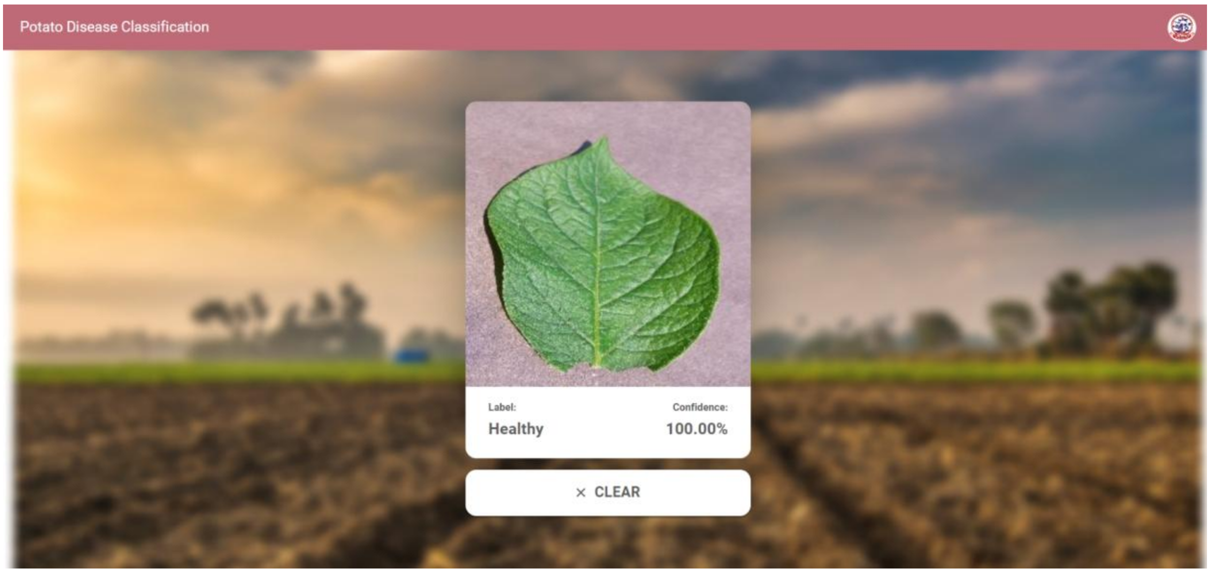


Figure:5.8

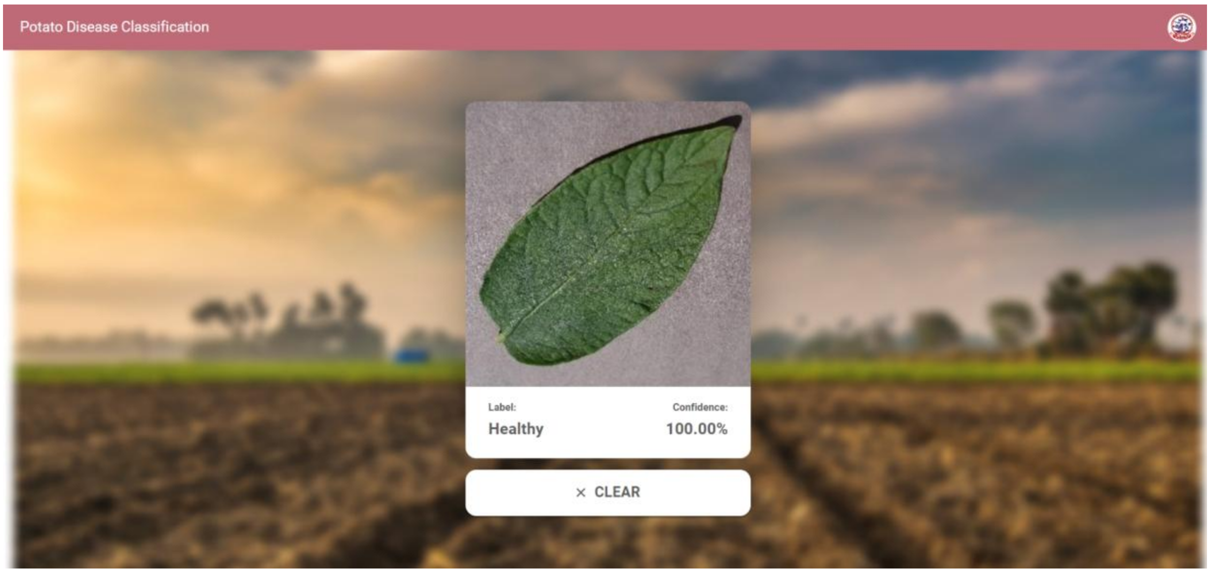


Figure:5.9

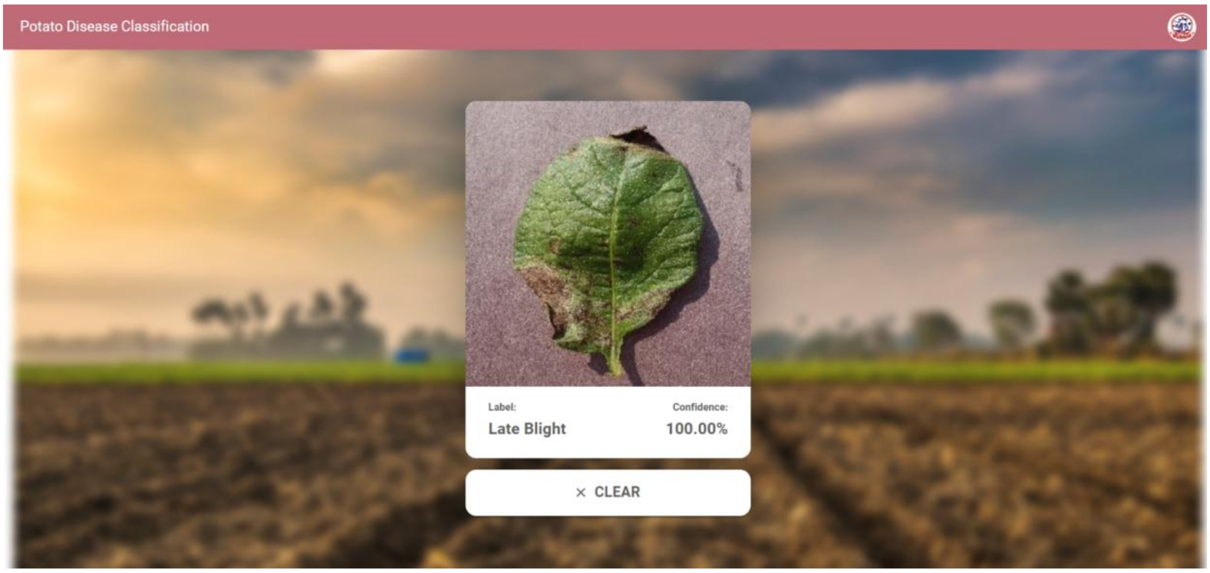


Figure:5.10

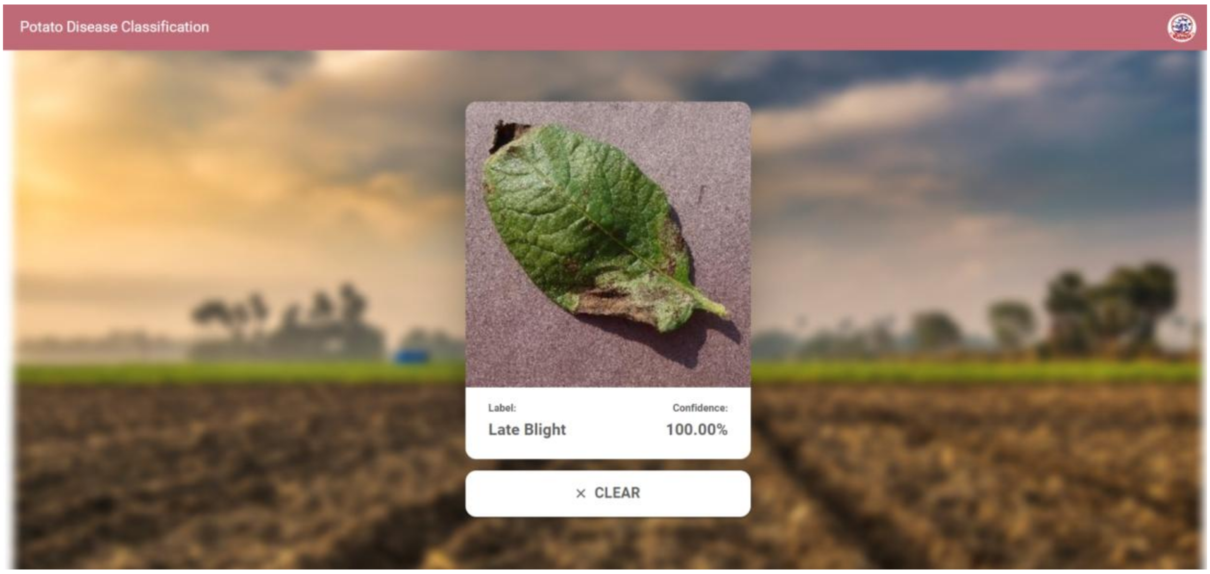


Figure:5.11

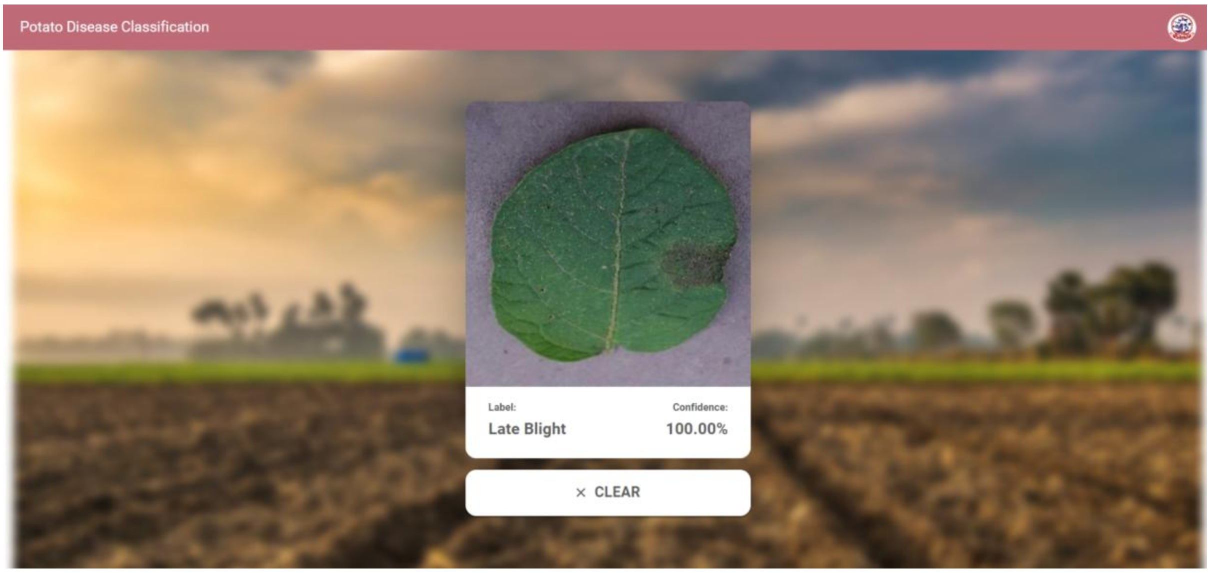


Figure:5.12

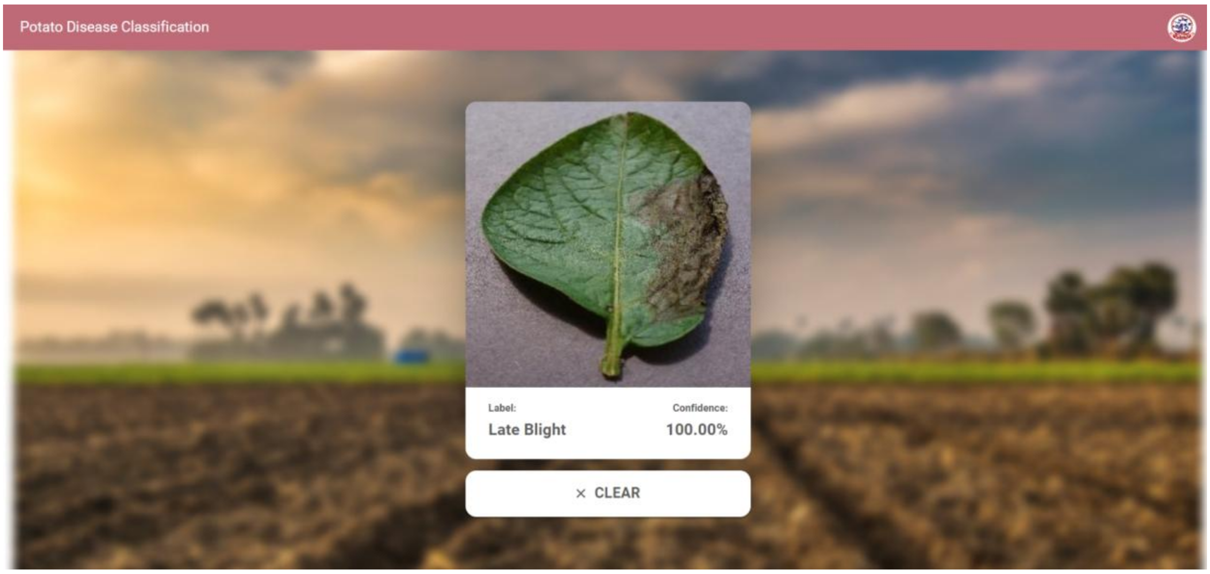


Figure:5.13

Chapter 6. Conclusion and Future Scope

6.1 CONCLUSION

Digitalization increasing across all the fields and it is high time to adopt digitalization into the field of agriculture as well to obtain better protection in terms of growth and yield. Keeping this intention as the motivation for the proposed model to detect and classify the affected and unaffected leaves of potato. We think this type of project will play a vital role in our agriculture sector The concept of activation functions, batch normalizations, convolutional layers, and fully connected layers are playing a key role in CNN-VGG16 architectures to attain better accuracy Most of the farmers of the in India are not literate and they can't know about the disease properly. They can't know the method of detecting disease. That's why the insect is destroying the potato and our farmers get to suffer from it. We think that, this work can change the situation of the potato grower in India

6.2 Future Scope

It is very important to detect a disease in a plant in the budding stage so that productivity and quality of the yield can be upgraded. Since disease detection needs a lot of expertise so it would be very beneficial if we could implement this system on the smartphone in which farmers can click a picture of the leaf and send it to the server. The server will automatically identify and classify the type of disease and send results along with prescribed medicines back to the smartphone. By building an android app we will continue the development process. And we will create a system where the farmers of India can easily get instant service and advice on their problem by detecting the disease In future, such research would be extended to multiple diseases detection on a single leaf and to localise the diseases, disease severity estimation, enhance the dataset, develop IoT-based real-time monitoring system, develop a website and launch a mobile application

References

[1] Md. Marjanul Islam Tarik, Sadia Akter, Abdullah Al Mamun, Abdus Sattar, —Potato Disease Detection Using Machine Learning, Department of Computer Science and Engineering Daffodil International University, 2017 IEEE Xplore part number: CFP21ONG-ART; doi: 10.1109/CCECE.2017.7946594

[2] Damian Bienkowskia, Matt J. Aitkenheadb, Alison K. Leesc, Christopher Gallaghera, Roy Neilsona. Detection and differentiation between potato diseases using calibration models trained with non-imaging spectrometry data, Vol. 167, Issue 6, 2020

[3] Monzurul Islam, Anh Dinh, Khan Wahid, Pankaj Bhowmik —Detection of Potato Diseases Using Image Segmentation and Multiclass Support Vector Machine, 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE).

[4] Kurniawati NN, Abdullah SNHS, Abdullah S, Abdullah S, —Investigation on image processing techniques for diagnosing paddy diseases, 2009 International Conference of Soft Computing and Pattern Recognition, IEEE, Malacca, 2009, pp. 272–277.

[5] Divyansh Tiwari, Mritunjay Ashish, Nitish Gangwar, Abhishek Sharma, Suhanshu Patel, Dr. Suyash Bhardwaj. —Potato Leaf Diseases Detection Using Deep Learning‖. Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020) IEEE Xplore Part Number: CFP20K74- ART; ISBN: 978-1- 7281-4876-2.

[6] Monzurul Islam, Anh Dinh, Khan Wahid, Detection of Potato Diseases Using Image Segmentation and Multiclass Support Vector Machine, Department of Electrical and Computer Engineering University of Saskatchewan Saskatoon, Canada, 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 978-1- 5090-5538-8/17/$31.00.

[7] Hu YH, Ping XW, Xu MZ, Shan WX, He Y, —Detection of Late Blight Disease on Potato Leaves Using Hyperspectral Imaging Technique, PubMed, 36(2), 2016, pp. 515-519 PMID: 27209760.