**A Main Project Report**

**on**

***“PLANT DISEASE PREDICTION”***

**By**

**Aviral Gupta –220940325017**

**Spurthi Malayamarutham- 220940325077**

**Shubham Shivathare-220940325071**

**Shubham Bhawar-220940325070**

**Suraj Gaikwad-220940325082**

*In fulfilment of*

*the requirements for the award of the degree of*

**Post Graduate Diploma**

**In**

**Big Data Analytics**

**(PG-DBDA)**

****

**Year 2023**

**Abstract**

We have done data analysis, data visualisation and prediction of data in this project. For that we collected dataset from the sources GitHub and Kaggle. The dataset we used is of Plant Disease detection using Deep learning Models which includes various plants images of Tomato, Potato, and Strawberry etc. The whole project is based on Python, PySpark and BigDL for fast processing in which we created a python notebook in Jupyter Lab. We have created webpage using Flask and HTML for each plant and each disease. We did prediction using some Deep learning algorithms available in python libraries. We can store the data in s3 bucket for later use to process the data in Spark(BigDL). This whole process and the insights about the data are explained in detail in this report.

**Table Of Content**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Contents** | **Page No.** |
| 1 | Abstract | 2 |
| 2 | Problem Statement | 4 |
| 3 | List of Abbreviations | 5 |
| 4 | Introduction | 6 |
| 5 | Motivation | 7 |
| 6 | System Architecture | 11 |
| 7 | Program | 24 |
| 8 | Output Screenshots | 30 |
| 9 | Future Enhancement | 33 |
| 10 | Conclusion | 34 |
| 11 | References | 35 |

**Problem Statement**

Smart farming system is an innovative technology that helps improve the quality and quantity of agricultural production in the country. Plant leaf disease has been one of the major threats to food security since long ago because it reduces the crop yield and compromises its quality.

Moreover, Plant diseases affect the growth of their respective species; therefore their early identification is very important. Many Machine Learning (ML) models have been employed for the detection and classification of plant diseases but, after the advancements in a subset of ML, that is, Deep Learning (DL), this area of research appears to have great potential in terms of increased accuracy. Many developed/modified DL architectures are implemented along with several visualization techniques to detect and classify the symptoms of plant diseases. Moreover, several performance metrics are used for the evaluation of these architectures/techniques. This review provides a comprehensive explanation of DL models used to visualize various plant diseases. In addition, some research gaps are identified from which to obtain greater transparency for detecting diseases in plants, even before their symptoms appear clearly.

Diagnosis of accurate diseases has been a major challenge and the recent advances in computer vision made possible by deep learning has paved the way for camera assisted disease diagnosis for plant leaf. It described the innovative solution that provides efficient disease detection and deep learning with convolutional neural networks (CNNs) has achieved great success in the classification of various plant leaf diseases. A variety of neuron-wise and layer-wise visualization methods were applied and trained using a CNN, with a publicly available plant disease given image dataset. So, it observed that neural networks can capture the colours and textures of lesions specific to respective diseases upon diagnosis, which can act like human decision-making.

**Key Words**: Disease detection, Transfer learning, TensorFlow, Keras, Convolutional Neural Network (CNN)-2, Inception V3,VGG-16,Resnet-50.

List Of Abbreviations

Appendix A: Glossary

ML - Machine Learning

TL - Transfer Learning

CNN - Convolutional Neural Network

VGG - Visual Geometry Group

Strategy - a plan of action designed to achieve a long-term or overall aim.

Constraint - a limitation or restriction

Class - particular group of things

Analysis - implementation

### Introduction

### India is an agricultural country, where most of the people depend on agriculture. The purpose of Agriculture is not only to feed ever growing population but it’s an important source of energy and a solution to solve the problem of global warming. Research in agriculture is aimed towards increase of productivity and food quality at reduced expenditure and with increased profit, which has received importance in recent time. The naked eye observation of experts is the main approach adopted in practice for detection and identification of plant diseases. The classification and recognition of crop diseases are of the major technical and economic importance in the agricultural Industry.

Disease detection in plants plays an important role in agriculture as farmers have often to decide whether the crop they are harvesting is good enough. It is of utmost importance to take this seriously as it can lead to serious problems in plants due to which product quality, quantity or productivity is affected.

Plant diseases cause a periodic outbreak of diseases leading to large-scale death which severely affects the economy. Here computer vision algorithms can be used to provide image-based automatic inspection.

Manual identification is labor intensive, less accurate and can be done only in small areas at a time. By this method, the plant diseases can be identified at the initial stage itself and the pest and infection control tools can be used to solve pest problems while minimizing risks to people and the environment.

### Deep learning is a branch of machine learning which is completely based on artificial neural networks, deep learning is also a kind of mimic of human brain because the neural network can mimic the human brain. It’s on hype nowadays because earlier we had lot of data and not enough processing power.

### A formal definition of deep learning is- neurons Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.

### 

### In human brain approximately there are 100 billion neurons, all together this is a picture of an individual neuron and each neuron is connected through thousands of their neighbors. The question here is how it recreates these neurons in a computer. So, it creates an artificial structure called an artificial neural net where we have nodes or neurons. It has some neurons for input value and some for output value and in between, there may be lots of neurons interconnected in the hidden layer.

### Motivation

### Agricultural productivity is something on which the economy highly depends. In addition to this, plant diseases and pests are a major problem in the agricultural sector. Plants are considered as energy supply to mankind. Plant diseases can affect the agriculture which can be resulted in to huge loss on the crop yield. Therefore, leaf diseases detection plays a vital role in agricultural field. Their detection at the initial stage is required to get rid of all the diseases as quickly as possible and to save ourselves from the destruction of crops.

### However, it requires large manpower, more processing time and extensive knowledge and skills about plant diseases. Hence, machine learning comes in play in the detection of diseases in plant leaves as it analyzes the data from various areas, and classifies it into one of the predefined set of classes.

### The features and properties like color, intensity and dimensions of the plant leaves are considered as a major fact for classification and the various types of plant diseases and different classification techniques in machine learning that are used for identifying diseases in different plants leaf.

### Proposed System

### In this project, we will try to model a classifier to classify plant using its leaf image that it is healthy or not. Let us assume a scenario in which; we find a bunch of randomly images of tomato leaf on our hard disk. Our task is to find that if the plant is diseased or not. For that we created CONV-2, Inception V3, VGG-16, and Resnet-50 Models.

### 

### We have to upload the image of the plant leaf. In the result section it will show us the condition of plant. It will show that the plant is healthy or diseased. If the plant is diseased then it will classify the name of the diseases.

### We planned to design the module so that a person with no knowledge about programming can also be able to use and get the information about the plants disease. It proposed system to predicting leaf diseases. It explains about the experimental analysis of our methodology.

### 

### Samples of images are collected that are comprised of different plant diseases like Tomato and Healthy Leaves. Different number of images is collected for each disease that was classified into database images and input images. The primary attributes of the image are based upon the shape and texture oriented features.

### Models Developed

### Inception V3 Algorithm: Inception v3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for GoogLeNet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge.

### 

### ResNet-50 Algorithm: ResNet-50 is a pre-trained Deep Learning model for image classification of the Convolutional Neural Network (CNN, or ConvNet), which is a class of deep neural networks, most commonly applied to analyzing visual imagery.

The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters. Residual Network (ResNet) is a deep learning model used for computer vision applications. It is a Convolutional Neural Network (CNN) architecture designed to support hundreds or thousands of convolutional layers. Deep residual networks like the popular ResNet-50 model is a convolutional neural network (CNN) that is 50 layers deep. A Residual Neural Network (ResNet) is an Artificial Neural Network (ANN) of a kind that stacks residual blocks on top of each other to form a network.

### 

### BigDL: BigDL is a distributed deep learning framework for Apache Spark, The BigDL release provides the following features:

### DLlib: A distributed deep learning library for Apache Spark\* (that is, the original BigDL framework with a Keras\*-style API and Spark machine learning pipeline support)

### Orca: Seamlessly scales out TensorFlow\* and PyTorch\* pipelines for distributed big data

### Friesian: A large-scale end-to-end recommender framework

### Chronos: Scalable time-series analysis using AutoML

### PPML: Privacy preserving big data analysis and machine learning

### 

# Amazon S3: Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides management features so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.

### How Amazon S3 works?

### 

### System Architecture

### Architecture diagram of Dataflow:

### 

### Fig 1. System Architecture

### As shown in figure there is a dataset which consist of all the different plant leaf disease images which we have taken into account. The module is trained repetitively to attain the maximum accuracy. If a new image is given to the module, it’s features get compared with the features that are already trained in the database. It then provides the appropriate result.

### System Features: The system consists of a webpage, which will enable the farmers to take images of plants using their mobile phones or upload a clicked image and send it to a central server where the central system in the server will analyze the pictures based on visual symptoms using image processing algorithms in order to measure the disease type. An expert group will be available to check the status of the image analysis data and provide suggestions based on the report and their knowledge, which will be sent to the farmer as a notification in the application.

### 

### Fig 2. Sequence Diagram

### Plant Disease Prediction Project are mainly based on this point:

### User: User is a main part of our project they may be farmer, agriculturist, etc. User takes an image of leaves to check whether the plant is healthy or not.

### System Application: Here the system will process this image into proper format accepted by the model and then takes the output from the model to show to the user.

### Image database: In this step the images of leaves were taken from various sources and collected at one in this case Kaggle from where we trained the model and done cleaning resizing data balancing and data augmentation before training the actual part so that the results are more accurate and precise along with these factors it also depends on the model which we are training as results of various models do vary

### Train CNN VGG16 Model: CNN model like VGG16 was used to train the model as we are using transfer learning only last layers from flatten are trained this model adds weights on images and that will help us to predict disease

### Execution State: This is final step where already trained model is used for validation purpose here, we input an image to get the prediction and hence the result.

### Class Diagram:

### 

### Fig5.Class Diagram

### The normalization class comprises of raw image and it is feeded to the CNN model which comprises of dense and weight.The CNN model classifies and detects by using the training model. The training model class comprises of the image dataset. Leaf detection gets use of the features

### CNN MODEL STEPS:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

### Conv2D: It is the layer to convolve the image into multiple images activation is the activation function.

### MaxPooling2D: It is used to max pool the value from the given size matrix and same is used for the next 2 layers.

### Flatten: It is used to flatten the dimensions of the image obtained after convolving it.

### Dense: It is used to make this a fully connected model and is the hidden layer.

### Dropout: It is used to avoid over fitting on the dataset and dense is the output layer contains only one neuron which decide to which category image belongs.

### Image Data Generator: It is that rescales the image, applies shear in some range, zooms the image and does horizontal flipping with the image. This Image Data Generator includes all possible orientation of the image.

### Training Process:Train\_datagen. Flow\_from\_directory is the function that is used to prepare data from the train\_dataset directory Target\_size specifies the target size of the image. test\_datagen. flow\_from\_directory is used to prepare test data for the model and all is similar as above. fit\_generator is used to fit the data into the model made above, other factors used are steps\_per\_epochstells us about the number of times the model will execute for the training data.

### Epochs: It tells us the number of times model will be trained in forward and backward pass.

### Validation process:validation\_data is used to feed the validation/test data into the model. validation\_stepsdenotes the number of validation/test samples.

### VGG16 Model

VGG16 is a convolution neural net (CNN) architecture which was used to win ILSVR(Imagenet) competition in 2014. It is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.

### 

### Fig 6.VGG16 model architecture

# Transfer Learning

Transfer learning has been highly successful for classification problems. The main advantage in using transfer learning is that instead of starting the learning process from scratch, the model starts from patterns that have been learned when solving a different problem which is similar in nature to the one being solved. This way the model leverages previous learning and avoids starting from scratch. In image classification, transfer learning is usually expressed through the use of pre-trained models. A pre-trained model is a model that was trained on a large benchmark dataset to solve a similar problem to the one that we want to solve.

# Data Augmentation

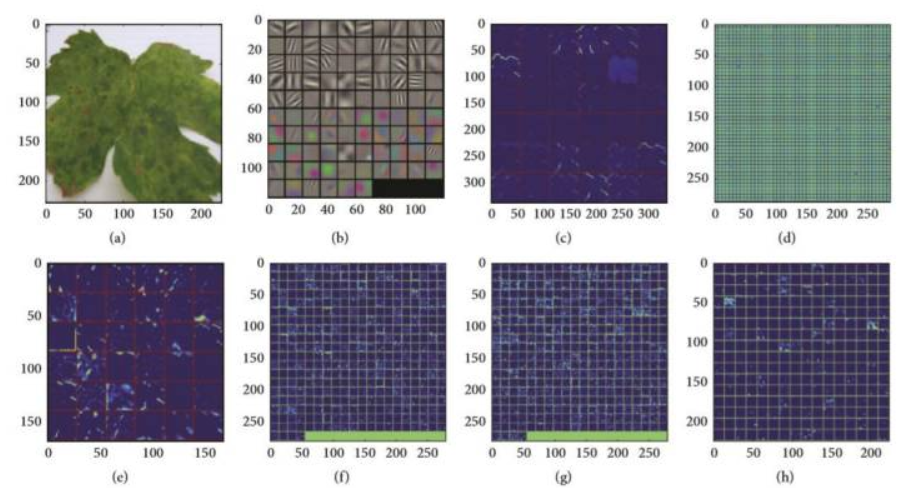
The images are resized to 256×256 pixels. I used data augmentation techniques like shearing, zooming, flipping and brightness change to increase the dataset size to more than double the original dataset size. The image rotation degree was set to be randomly generated from 0 to 45. The number of normal samples in the dataset was increased from 1,583 to 4,266 by performing the image augmentation techniques. In this manner, the number of samples for each class was equalized. This equal distribution makes it possible to use all of the data instead of selecting random data during the training process. It is expected that this situation increases the accuracy of the training and positively affects the classification results.

# Network Architecture

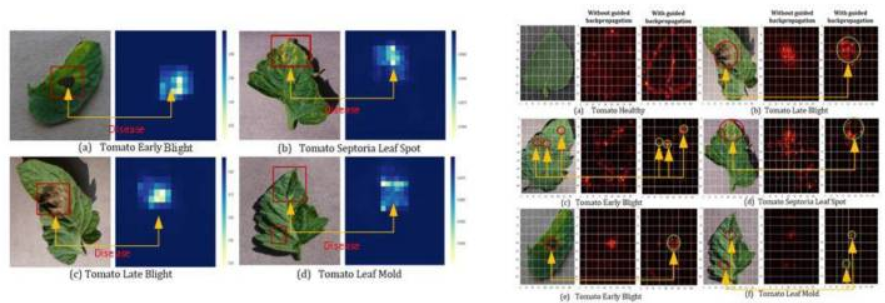
The network architecture can be explained in the following points:

1. We split the data-set into three sets — train, validation and test sets.
2. We tried with pre trained models like Inception v3, ResNet 50 and VGG16. The last layer is used for the classification with softmax as the activation function.
3. The loss function used is binary cross-entropy and trained the model for 10 epochs. Multi class log loss is chosen as the evaluation metric. Activation function used was Relu throughout except for the last layer where it was Sigmoid as this is a binary classification problem.
4. We used 30 percent dropouts to reduce overfitting in between the layers and batch normalization to reduce internal covariate shift.

For the practical experimentation of detection of plants’ diseases, an actual/real background/environment should be considered in order to evaluate the performance of the DL model more accurately. In most of the above approaches, the selected datasets considered plain backgrounds which are not realistic scenarios for identification and classification of the diseases, except for a few of them that have considered the original backgrounds. The output of the visualization techniques used in several researches are shown:



Feature maps after the application of convolution to an image: (a) real image, (b) first convolutional layer filter, (c) rectified output from first layer, (d) second convolutional layer filter, (e) output from second layer, (f) output of third layer, (g) output of fourth layer, (h) output of fifth layer.



Tomato plant disease detection by heat map: on left hand side (a) tomato early blight, (b) tomato septoria leaf spot, (c) tomato late blight and (d) tomato leaf mold) and saliency map; on right hand side (a) tomato healthy, (b) tomato late blight, (c) tomato early blight, (d) tomato septoria leaf spot, (e) tomato early blight, (f) tomato leaf mold).

## Design and Implementation Constraints:

* Using machine learning and training dataset consisting of images is used. Image features are extracted using several models and CNN is applied for the classiﬁcation. We wanted to build a model that could automatically classify a plant disease using leaf images.

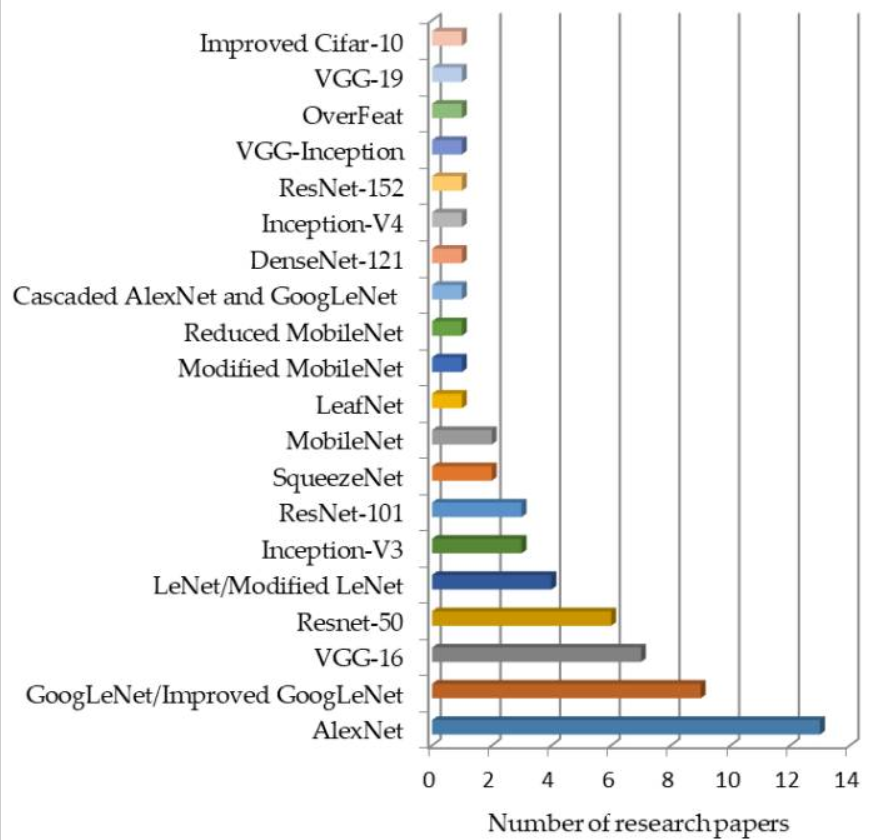
### We planned to design the module so that a person with no knowledge about programming can also be able to use and get the information about the plants disease.

### It proposed system to predicting leaf diseases. It explains about the experimental analysis of our methodology. Samples of 10 images are collected that are comprised of different plant diseases like Tomato and Healthy Leaves.

### Different number of images is collected for each disease that was classified into database images and input images. The primary attributes of the image are based upon the shape and texture-oriented features.

### This way we could address the real-time problem faced by the Farmers.

### We have used flask framework for UI webpage linking wih model.



### Fig 7. Popular CNN models

## SOFTWARE AND HARDWARE REQUIREMENTS

## Hardware Interfaces:

This module is intended to work as a part of the 64-bit distribution package only. The algorithm of the module is based on the CNN.

Consequently, it is recommended to use video cards (GPUs) for better performance.

The NVidia GPU with the computing power index of at least 5.0 is required for the module’s operation.

## Software Interfaces:

**Software Requirements:**

* Anaconda 3.8 (for creating virtual env. And Jupyter notebook)
* Tensorflow (Python library for deep learning model)
* Python 3.6 (for development of algorithm)
* Keras (for building CNN Models)
* BigDL Library
* Spark(PySpark)
* Java JDK
* Matplotlib and Seaborn Python Visualization libraries

**Algorithm:**CNN with Transfer learning on VGG 16.

### Functional Requirements:

**REQ-1: Anaconda 3.8**

Anaconda is a [distribution](https://en.wikipedia.org/wiki/Software_distribution) of the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)) programming languages for [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing) ([data science](https://en.wikipedia.org/wiki/Data_science), [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications, large-scale data processing, [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics), etc.), that aims to simplify [package management](https://en.wikipedia.org/wiki/Package_management) and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

Package versions in Anaconda are managed by the [package management system](https://en.wikipedia.org/wiki/Package_manager) [*conda*](https://en.wikipedia.org/wiki/Conda_(package_manager)). This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python.

**REQ-2: VGG-16**

**VGG16** is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to [ILSVRC-2014](http://www.image-net.org/challenges/LSVRC/2014/results). It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU’s.

**REQ-3: Keras**

Keras gives fundamental reflections and building units for creation and transportation of ML arrangements with high iteration velocity. It takes full advantage of the scalability and cross-platform capabilities of TensorFlow. The core data structures of Keras are layers and models . All the layers used in the CNN model are implemented using Keras. Along with the conversion of the class vector to the binary class matrix in data processing, it helps to compile the overall model.

**REQ-4: Tensorflow**

[TensorFlow](https://www.tensorflow.org/) is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of [tools](https://www.tensorflow.org/resources/tools), [libraries](https://www.tensorflow.org/resources/libraries-extensions), and [community](https://www.tensorflow.org/community) resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications. TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence Research organization to conduct machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well.

**REQ-5: BigDL Library**

BigDL is an open-source distributed deep learning framework for Apache Spark. It allows users to write deep learning applications as Scala or Python programs and run them on existing Spark or Hadoop clusters. BigDL provides a high-level API that enables developers to easily create and train deep neural networks on large datasets. It supports popular deep learning models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory networks (LSTMs).

**REQ-6: Java(JDK)**

Java JDK (Java Development Kit) is a software development kit used for developing Java applications. It includes a set of tools, libraries, and utilities that are necessary for developing, testing, and running Java applications. The Java JDK contains the Java Virtual Machine (JVM), which is responsible for executing Java programs. It also includes the Java compiler, which converts Java source code into bytecode that can be run on the JVM. Other important tools included in the JDK are the debug.

**REQ-7:****Matplotlib and Seaborn**

Matplotlib and Seaborn are two popular Python libraries used for data visualization. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It provides a range of plots and charts such as line plots, scatter plots, bar plots, histograms, pie charts, and more. Matplotlib allows users to customize the visual appearance of plots and charts by setting properties such as color, font size, labels, and more.

**REQ-8:Spark (PySpark)**

Spark, also known as Apache Spark, is an open-source distributed computing system used for processing large-scale data sets. It was developed at the University of California, Berkeley, and is maintained by the Apache Software Foundation. Spark provides an easy-to-use programming interface for processing large data sets in parallel across a cluster of computers. It allows users to write applications in several programming languages including Scala, Java, Python, and R.

PySpark is the Python API for Apache Spark, an open-source distributed computing system used for processing large-scale data sets. PySpark allows Python developers to write Spark applications using Python code. PySpark provides an easy-to-use programming interface for processing large data sets in parallel across a cluster of computers. It allows users to write applications in Python that can be distributed and processed across a Spark cluster. PySpark provides APIs for working with RDDs (Resilient distributed datasets).

# Other Non-functional Requirements:

* + 1. **PC/Laptop (Windows -64bit)**

To perform this Project it required only 64 bit laptop. This is compatible to run Machine Learning Program and carrying Compatible tool and Software.

* + 1. **NVIDIA GEFORCE GPU Drivers**

Enterprise customers with a current vGPU software license (GRID vPC, GRID vApps or Quadro vDWS), can log into the enterprise software

* + 1. **CPU Quad Core**
* A quad-core [CPU](https://techterms.com/definition/cpu) has four processing cores in a single chip. It is similar to a [dual-core](https://techterms.com/definition/dualcore) CPU, but has four separate [processors](https://techterms.com/definition/processor) (rather than two), which can process instructions at the same time.
* Quad-core CPUs have become more popular in recent years as the [clock speeds](https://techterms.com/definition/clockspeed) of processors have plateaued. By including multiple cores in a single CPU, chip manufacturers can generate higher performance without boosting the clock speed. However, the performance gain can only be realized if the computer's software supports [multiprocessing](https://techterms.com/definition/multiprocessing). This allows the software to split the processing load between multiple processors (or "cores") instead of only using one processor at a time. Fortunately, most modern operating systems and many programs provide support for multiprocessing.

### TRAINING AND TESTING MODEL

### The dataset is preprocessed such as Image reshaping, resizing and conversion to an array form. Similar processing is also done on the test image. A dataset consisting of about different plant leaf diseases is obtained, out of which any image can be used as a test image for the software.

### Collected 2 datasets out of which 1 dataset has images of Tomato of 10 different classes out of which 1 is healthy and remaining 9 are various diseases that attack tomato plant. Other dataset has various leaf images of 14 plants include Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, and Tomato. Each plant has healthy and diseased leaf images.

### 

### 

### Fig 8. Training model

### The train dataset is used to train the model (CNN) so that it can identify the test image and the disease it has CNN has different layers that are Dense, Dropout, Activation, Flatten, Convolution2D, and MaxPooling2D. After the model is trained successfully, the software can identify the disease if the plant species is contained in the dataset. After successful training and preprocessing, comparison of the test image and trained model takes place to predict the disease.

### 

### Fig 9. Testing Model

### Website Building

### Flask:Flask is a web framework, it’s a Python module that lets you develop web applications easily. It’s has a small and easy-to-extend core: it’s a micro framework that doesn’t include an ORM (Object Relational Manager) or such features. It does have many cool features like url routing, template engine. It is a WSGI web app framework. Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine.Both are Pocco projects.

### WSGI: The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

### Werkzeug: Werkzeug is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeg as one of its bases.

### Jinja2: jinja2 is a popular template engine for Python.A web template system combines a template with a specific data source to render a dynamic web page.

### Streamlit: Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers. Streamlit Components have two parts: A frontend which is implemented in any web technology you prefer (JavaScript, React, Vue, etc.) and gets rendered in Streamlit apps via an iframe tag. A Python API which Streamlit client apps use to instantiate the frontend and communicate with it.

### Output

### On running website building python code in Flask, it creates a app which will render a website page and user can select any image for prediction as shown below.

### 

### Below is the output page which gives information of the disease that leaf is affected and treatment as well as shown below.

### 

### Conclusion

### It focused how image from given dataset (trained dataset) in field and past data set used predict the pattern of plant diseases using CNN model. This brings some of the following insights about plant leaf disease detection. As maximum types of plant leaves will be covered under this system, farmer may get to know about the leaf which may never have been cultivated and lists out all possible plant leaves, it helps the farmer in decision making of which crop to cultivate. Also, this system takes into consideration the past production of data which will help the farmer get insight into the demand and the cost of various plants in market.

### In India where farmers are the backbone of economy as well as of food production, early insights about plant cultivated would play a major role in making our backbone strong.