A Major Project

on

PLANT DISEASE DETECTION AND CLASSIFICATION WITH PESTICIDES SUGGESTION FOR E-AGRICULTURE APPLICATIONS

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In Partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

By

V. RETESH KUMAR – 19RA1A0530

L.V.N.S GOWTHAM – 19RA1A0550

V. YUVA RAJ – 20RA5A0510

Under the guidance of

Dr. J. Srikanth

Professor, CSE Dept

Department of Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUH, Ghanpur(V), Ghatkesar(M), Medchal(D)-500088)

2019 -2023

KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUH, Ghanpur(V), Ghatkesar(M), Medchal(D)-500088)

CERTIFICATE

This is to certify that the project work entitled **"PLANT DISEASE DETECTION AND CLASSIFICATION WITH PESTICIDE SUGGESTION FOR E-AGRICULTURE APPLICATIONS"** is submitted by Mr. V. RETESH KUMAR, Mr. L.V.N.S. GOWTHAM, Mr. V. VUYA RAJ bonafied students of **Kommuri Pratap Reddy Institute of Technology** in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in **Computer Science and Engineering** of the **Jawaharlal Nehru Technological University Hyderabad,** during the year 2022-23.

Internal Guide
Dr. J. Srikanth

HOD
Mrs.G.Sujatha

Project Coordinator
Dr. C. Veena

External Examiner

DECLARATION

We hereby declare that this project work entitled **"PLANT DISEASE DETECTION AND CLASSIFICATION WITH PESTICIDE SUGGESTION FOR E-AGRICULTURE APPLICATIONS"** in partial fulfillment of requirements for the award of degree of **Computer Science and Engineering** is a bonafide work carried out by us during the academic year 2022- 23.

We further declare that this project is a result of our effort and has not been submitted for the award of any degree by us to any institute.

By

V. Retesh Kumar (19RA1A0530)
L.V.N.S.Gowtham (19RA1A0550)
V. Yuva raj (20RA5A0510)

ACKNOWLEDGEMENT

It gives us immense pleasure to acknowledge with gratitude, the help and support extended throughoutthe project from the following:

We will be very much grateful to almighty our **Parents** who have made us capable of carrying out ourjob.

We express our profound gratitude to **Dr.D. Eshwar, Principal of Kommuri Pratap Reddy Institute of Technology**, who has provided necessary infrastructure and resources in completing our project successfully.

We are grateful to Mrs.**G. Sujatha** who is our **Head of the Department, CSE** for her amiable ingenious and adept suggestions and pioneering guidance during the project.

We express our gratitude and thanks to the Project Coordinator **Dr.C. Veena, Professor** of our department for her contribution for making it success with in the given time duration.

We express our deep sense of gratitude and thanks to **Internal Guide, Dr. J. Srikanth Professor,** forhis guidance during the project.

We are also very thankful to our **Management, Staff Members** and all **Our Friends** for their valuable suggestions and timely guidance without which we would not have been completed it.

By

V. Retesh Kumar (19RA1A0530)
L.V.N.S.Gowtham (19RA1A0550)
V. Yuva raj (20RA5A0510)

Vision of the Institute

To emerge as a premier institute for high quality professional graduates who can contribute to economic and social developments of the Nation.

Mission of the Institute

Mission	Statement
IM_1	To have holistic approach in curriculum and pedagogy through industry interface to meet the needs of Global Competency.
IM_2	To develop students with knowledge, attitude, employability skills, entrepreneurship, research potential and professionally Ethical citizens.
IM_3	To contribute to advancement of Engineering & Technology that would help to satisfy the societal needs.
IM_4	To preserve, promote cultural heritage, humanistic values and Spiritual values thus helping in peace and harmony in the society.

Vision of the Department

To Provide Quality Education in Computer Science for the innovative professionals to work for the development of the nation.

Mission of the Department

Mission	Statement
DM_1	Laying the path for rich skills in Computer Science through The basic knowledge of mathematics and fundamentals of engineering
DM_2	Provide latest tools and technology to the students as a part of learning infrastructure
DM_3	Training the students towards employability and entrepreneurship to meet the societal needs.
DM_4	Grooming the students with professional and social ethics.

Program Educational Objectives (PEOs)

PEO's	Statement
PEO1	The graduates of Computer Science and Engineering will have successful career in technology.
PEO2	The graduates of the program will have solid technical and professional foundation to continue higher studies.
PEO3	The graduate of the program will have skills to develop products, offer services and innovation.
PEO4	The graduates of the program will have fundamental awareness of industry process, tools and technologies.

Program Outcomes

PO1	**Engineering Knowledge:** Apply the knowledge of mathematics, science, Engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental context, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9	Individual and team network: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, being able to comprehend and write effective reports and design documentation, make Effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work,as a member and leader in a team, to manage projects and in multidisciplinary environment.
PO12	Life-Long learning: Recognize the need for, and have the preparation and able to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES

PSO1	Foundation of mathematical concepts: To use mathematical methodologies to crack problem using suitable mathematical analysis, data structure and suitable algorithm.
PSO2	Foundation of Computer Science: The ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.
PSO3	Foundation of Software development: The ability to grasp the software development lifecycle and methodologies of software systems. Possess competent skills and knowledge of software design process.

ABSTRACT

A country's inventive growth is dependent on the agricultural sector. Agriculture, the foundation of all nations, offers food and raw resources. Agriculture is hugely important to humans as a food source. As a result, plant diseases detection has become a major concern. Traditional methods for identifying plant disease are available. However, agriculture professionals or plant pathologists have traditionally employed empty eye inspection to detect leaf disease. This approach of detecting plant leaf disease traditionally can be subjective, time-consuming, as well as expensive, and requires a lot of people and a lot of information about plant diseases. It is also possible to detect plant leaf diseases using an experimentally evaluated software solution. Currently, machine learning and deep learning are being used in recent years. This work is focusedon implementation of Plant disease detection and classification (PDDC-Net) using deep learning models. The preprocessing operation was also performed to remove the different types of noises, which also normalizes the dataset images. Further, the PDDC-Net implements the operation using residual network based convolutional neural network (ResNet-CNN) for feature extraction and classification. Experimental results have shown that the proposed PDDC-Net model achieved a good accuracy rate for plant leaf disease detection and classification.

TABLE OF CONTENTS

List of figures

List of Screenshots

1. INTRODUCTION

1.1 Overview

As a superpower with more than 20% of the world's total population, China has been facing the problem of insufficient arable land resources. According to the survey data of the Ministry of Agriculture, the proportion of cultivated land in China is even less than 10% of China's land area. According to statistics data, the mountainous area accounts for about two-thirds of the total land area in China, while the plain area accounts for only one-third. About one third of the country's agricultural population and arable land are in mountainous areas. This situation has resulted in the relatively poor production conditions of agriculture, forestry, and animal husbandry in China. According to the statistics of the Food and Agriculture Organization of the United Nations, the per capita cultivated land area in China is less than half of the world average level and shows a decreasing trend year by year. Once the natural disasters cause agricultural production reduction, it will seriously affect the output of agricultural products and agricultural development. So how to develop agriculture stably, especially in the complex environment, is extremely important for China.

Although with the development of science and technology, agricultural production is progressing. But due to various natural factors and non-natural factors, the yield of crops has not been greatly improved. Among the various factors, the largest proportion is the problem of crop diseases and insect pests. According to statistics, the area of crops affected by pests and diseases in China is as high as 280 million km2 every year,and the direct yield loss is at least 25 billion kg [1]. In recent years, this problem is on the rise and seriously threatens the development of planting industry. Timely diagnosis and prevention of crop diseases has become particularly important. At present, agricultural workers often use books and network, contact local experts, and use other methods to protect and manage crop diseases. But for various reasons, misjudgements and other problems often occur, resulting in agricultural production is deeply affected.

1.2 Problem statement

At present, the research on crop diseases is mainly divided into two directions. The first one is the traditional physical method, which is mainly based on spectral detection to identify different diseases. Different types of diseases and insect pests cause different leaf damage, which leads to different spectral absorption and reflection of leaves eroded by diseases and healthy crops. The other one is to use computer vision technology to identify images. That is to say, the characteristics of disease images are extracted by using computer related technology, and the recognition is carried out through the different characteristics of diseased plants and healthy plants.

In recent years, the rapid development of artificial intelligence has made life more convenient, and AI has become a well-known technology. For example, AlphaGo defeated the world champion of Go. Siri and Alexa as voice assistants of Apple and Amazon are all applications of artificial intelligence technology represented by deep learning in various fields. As the key research object of computer vision and artificial intelligence, image recognition has been greatly developed in recent years. In agricultural applications, the goal of image recognition is to identify and classify different types of pictures, and analyze the types of crops, disease types, severity and so on. Then we can formulate corresponding countermeasures to solve various problems in agricultural production in a timely and efficient manner. To further ensure and improve the yield of crops and help the better development of agriculture.

With the rapid development of deep learning [2], especially in image recognition [3], speech analysis, natural language processing and other fields, it shows the uniqueness and efficiency of deep learning. Compared with the traditional methods, deep learning is more efficient in the diagnosis of crop diseases in the field of agricultural production. The deep learning model can monitor, diagnose, and prevent the growth of crops in time. Image recognition of crop diseases and insect pests can reduce the dependence on plant protection technicians in agricultural production, so that farmers can solve the problem in time. Compared with artificial identification, the speed of intelligent network identification is much faster than that of manual detection. And the recognition accuracy is getting higher and higher in the continuous development. The establishment of a sound agricultural network and the combination of Internet and agricultural industry can not only solve the problems related to crop yield affected by diseases and

insect pests, but also be conducive to the development of agricultural informatization [4].

However, due to the rugged terrain of the mountain environment, the surrounding interference factors are greater. Therefore, the image acquisition is more difficult than the general environment. In addition, the camera and network transmission needed for image recognition and processing will also have a certain impact. Therefore, it is more difficult to carry out intelligent recognition in mountainous areas. This paper tries to build the Internet of Things platform in the complex environment of mountainous areas and carry out the research on the identification model of crop diseases and insect pests. The purpose of this model is to improve agricultural informatization, deal with the harm of pests and diseases to crops and improve crop yield.

2. LITERATURE SURVEY

López et al. proposed an autonomous monitoring system based on a low-cost image sensor that it can capture and send images of the trap contents to a remote-control station with the periodicity demanded by the trapping application. This autonomous monitoring system will be able to cover large areas with very low energy consumption.This issue would be the main key point in this study; since the operational live of the overall monitoring system should be extended to months of continuous operation without any kind of maintenance (*i.e.*, battery replacement). The images delivered by image sensors would be time-stamped and processed in the control station to get the number of individuals found at each trap. All the information would be conveniently stored at the control station, and accessible via Internet by means of available networkservices at control station (WiFi, WiMax, 3G/4G, *etc.*).

Srivastav et al. focused on a pest control and monitoring system for efficient sugarcane crop production, which is a staple crop grown in Pune. The main pests that affect sugarcane are top shoot borer, stalk borer, rood borer and sugarcane wooly aphid. Apart from this, the main diseases that affect sugarcane crop are red rot, Smut, Grassy Shoot and Wilt. The system uses an acoustic device sensor which monitors the noise level of the pests and gives an indication to the farmer through an alarm when the noise crosses a threshold. The dissemination of is done via a network of wireless sensors connected to a control room computer. Transmission and reception of field data is through ZigBee 802.15.4 digital communication device standard. The system covers large areas with very low energy consumption.

Athanikar et al. described a neural network-based detection and classification of Potato leaf samples using Segmentation of K-Means Clustering. Algorithms are developed to acquire and process colour images of single leaf samples. Different leaves like healthy and diseased are considered for the study. The developed algorithms are used to extract over 24 (colour, texture, and area) features. The texture features are extracted from the gray level co-occurrence matrix (GLCM). A back Propagation Neural Network (BPNN)-based classifier is used to identify and classify the unknown leaf that is the leaf is healthy or diseased, if leaf is diseased, one then classifies the disease by giving description (name, cause, pesticides). The colour, texture and area features are presented to the neural network for training purposes. The trained network is then used

to identify and classify the unknown leaf samples. The classification is carried out using different types of features sets, viz., colour, texture, and area. Classification accuracies of over 92% are obtained for all the leaves samples (healthy and diseased) using all the three feature sets.

Wang et al. recognized method to realize plant image diseases, four kinds of neural networks including backpropagation (BP) networks, radial basis function (RBF) neural networks, generalized regression networks (GRNNs) and probabilistic neural networks (PNNs) were used to distinguish wheat stripe rust from wheat leaf rust and to distinguish grape downy mildew from grape powdery mildew based on color features, shape features and texture features extracted from the disease images. The results showed that identification and diagnosis of the plant diseases could be effectively achieved using BP networks, RBF neural networks, GRNNs and PNNs based on image processing.

Samantha et al. proposed image processing methodology to detect scab disease of potato. In this paper first, the captured images are collected from different potato field and are processed for enhancement. Then image segmentation is carried out to get target regions (disease spots). Finally, analysis of the target regions (disease spots) based on histogram approach to finding the phase of the disease and then the treatment consultative module can be prepared by on the lookout for agricultural experts, so plateful the farmers.

Too et al. focused on fine-tuning and evaluation of state-of-the-art deep convolutional neural network for image-based plant disease classification. An empirical comparison of the deep learning architecture is done. The architectures evaluated include VGG 16, Inception V4, ResNet with 50, 101 and 152 layers and DenseNets with 121 layers. The data used for the experiment is 38 different classes including diseased and healthy images of leafs of 14 plants from plant Village. Fast andaccurate models for plant disease identification are desired so that accurate measures can be applied early. Thus, alleviating the problem of food security. In this experiment,DenseNets has tendency's to consistently improve in accuracy with growing number ofepochs, with no signs of overfitting and performance deterioration.

Mohanty et al. used a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions, in this work trained a deep convolutional

neural network to identify 14 crop species and 26 diseases (or absence thereof). The trained model achieves an accuracy of 99.35% on a held-out test set, demonstrating the feasibility of this approach. Overall, the approach of training deep learning models on increasingly large and publicly available image datasets presents a clear path toward smartphone-assisted crop disease diagnosis on a massive global scale.

Dyrmann et al. presented a method that is capable of recognising plant species in colour images by using a convolutional neural network. The network is built from scratch trained and tested on a total of 10,413 images containing 22 weed and crop species at early growth stages. These images originate from six different data sets, which have variations with respect to lighting, resolution, and soil type. This includes images taken under controlled conditions about camera stabilisation and illumination, and images shot with hand-held mobile phones in fields with changing lighting conditions and different soil types. For these 22 species, the network can achieve a classification accuracy of 86.2%.

Sa et al. presented a novel approach to fruit detection using deep convolutional neural networks. The system builded an accurate, fast and reliable fruit detection system, which is a vital element of an autonomous agricultural robotic platform; it is a key element for fruit yield estimation and automated harvesting. Recent work in deep neural networks has led to the development of a state-of-the-art object detector termed Faster Region-based CNN (Faster R-CNN). We adapt this model, through transfer learning, for the task of fruit detection using imagery obtained from two modalities: colour (RGB) and Near-Infrared (NIR). Early and late fusion methods are explored for combining the multi-modal (RGB and NIR) information.

Sladojevic et al. studied the plant disease recognition has been proposed for the first time. All essential steps required for implementing this disease recognition model are fully described throughout the paper, starting from gathering images to create a database, assessed by agricultural experts. Caffe, a deep learning framework developed by Berkley Vision and Learning Centre, was used to perform the deep CNN training. The experimental results on the developed model achieved precision between 91% and 98%, for separate class tests, on average 96.3%.

Ahmeb and Wang proposed a crop disease and pest identification model based on deep learning from the perspective of ecological and environmental protection to solve the

problems of many kinds of crop diseases and pests, fast diffusion speed, and long-time of manual identification of diseases and pests. Firstly, crop images are collected by field sampling to collect data set, and image preprocessing is completed by using nearest neighbor interpolation. Then, the network structure of the AlexNet model is improved. By optimizing the full connection layer, different neuron nodes and experimental parameters are set. Finally, the improved AlexNet model is used to identify crop diseases and pests. And the recognition accuracy of this method on other data sets is not less than 91% which has good portability.

Tao and Cuicu studied the aking leaf black spot, anthracnose, and leaf blight of Ophiopogon japonicus as the research objects, lesions were separated by K-Means clustering segmentation technology. PCA (principal component analysis) was carried out on the 46-dimensional eigenvectors composed of color, shape, and texture features, and then the multi-level classifier designed by SVM (support vector machine) was used to identify lesions. The recognition rate of the developed leaf disease recognition system of O. japonicus achieved 93.3%. The results indicated that the system is of great significance to the prevention and control of O. japonicus diseases and the modernization of O. japonicus industry.

Ranjith et al. studied an outline on the major insect pests and the natural enemies associated with wheat. Insect pests and natural enemies in wheat vary from place to place and there are many other insect pests and natural enemies which are associated with wheat ecosystem than those documented during this study. So, further research with in-depth study is recommended so that the role of natural enemies in suppressing the pest population will be helpful for integrated pest management in the wheat ecosystem.

Srinivasan et al. studied to reduce the over-reliance on chemical insecticides, evaluated the effectiveness of microbial pesticides (Bacillus thuringiensis and Metarhizium anisopliae formulations), and neem leaf extract alone and in combination (as an IPM package) against aphids, thrips, and pod borer on yard-long bean in three different provinces of Cambodia from 2015 to early 2018. The bio-pesticides reduced the incidences of thrips (Megalurothrips *usitatus*), and the infestation by the aphid (Aphis craccivora) and the pod-borer (Maruca *vitrata*) to the levels equivalent to chemical pesticide (abamectin) during trials in 2015 and 2016. Hence, the IPM package

can be a better alternative to chemical pesticides in managing the key insect pests on yard-long bean in Cambodia.

Toyinbo et al. studied the genetic variability for thrips resistance, estimate heritability of yield and other traits and investigate inter-trait relationships under thrips infestation. One hundred and fifty-six cowpea lines, including one resistant and one susceptible check, were screened for resistance under natural infestation at two locations in Nigeria, in 2016. Test lines were scored for thrips damage weekly for three consecutive weeks, after removal of spreader plants, to obtain damage scores (DS) 1, 2 and 3 while data were collected on agronomic traits. The data were subjected to analysis of variance from which genetic components of the phenotypic variance were computed. Genetic variability among the lines suggests the possibility of genetic control of thrips while number of pods per peduncle, number of peduncles per plant and DS3 would serve as useful selection criteria for thrips resistance.

Authors of [14] developed a mobile software of plant hospital, which could assist users to diagnose lots of kinds plant disease through deep learning technical. An image dataset consisting of 54,306 pictures of healthy or infected plant leaves is used to train a CNN model, to identify 14 kinds of crop species and 26 types of diseases. Authors of [15] did the similar work. They change CNN model to Deep CNN, to increase the ability of plant disease diagnosis and extend the ability of distinguishing plants from their surroundings.

Plants species classification was also attempted to solve by deep learning method. In [16], authors tried to recognize weeds and plant species by CNN model trained with colourful images. A dataset consisting of 10,413 images with 22 weeds and crop species was tested, and the network failed to classify some plant species due to absence of training sample of corresponding species.

A system called DeepFruits was developed to detecting fruit in [17]. The authors use imagery data to detect fruit by CNN approach. To build an accurate, fast, and reliable fruit detection system, they choose the faster-RNN model and made some adjustment [18]. The trained model was able to achieve an improvement of 0.838 precision and recall rate in sweet pepper detection task. They claimed they could complete the entire process of training an annotating a new model per fruit in four hours [17].

At present, the typical convolutional neural networks widely used are as follows.

1) LeNet-5 [19], [20]: Although proposed very early, but LeNet-5 is a complete and successful neural network, especially in handwritten numeral recognition system applications. The LeNet-5 network has seven layers, including two convolution layers, two convergence layers (also called pooling layers), and three full connection layers. The input image size is $32 * 32$, and the output corresponds to 10 categories.

2) ALexNet [21]: AlexNet consists of five convolution layers, three convergence layers and three full connection layers. ALexNet absorbs the idea and principle of LeNet-5 network and makes many innovations. These include using the ReLU function instead of the Sigmoid function to solve the gradient dispersion problem. Dropout is used at the fully connected level to avoid overfitting.

3) Inception Network [22]: Inception is different from the general convolution neural network in that it contains multiple convolution kernels of different sizes in its convolution layer, and the output of Inception is the depth stitching of the feature map. GoogLeNet, the winner of the 2014 ImageNet Image Classification Competition, is the earliest version of Inception v1 used.

4) Residual network [23]: The core idea of residual network is to make a non-linear element composed of neural networks infinitely approximate the original objective function or residual function by using the general approximation theorem. Many nonlinear elements form a very deep network, which is called residual network.

3. SYSTEM ANALYSIS

3.1 INTRODUCTION

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analysis plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system id done. A key question considered here is, “what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

3.1.1 SYSTEM MODEL

SOFTWARE DEVELOPMENT LIFE CYCLE

There is various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as"Software Development Process Models". Each process model follows a particular life cycle

in order to ensure success in process of software development.

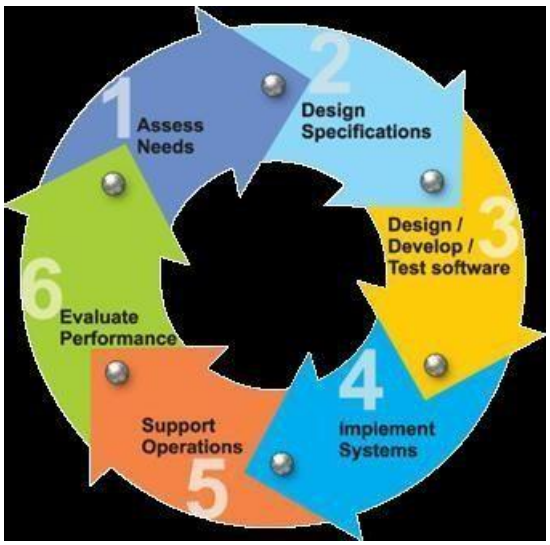


Fig 3.1.1 SDLC

3.1.2 REQUIREMENTS

Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements. Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. This produces a nice big list of functionality that the system should provide, which describes functions the system should perform, business logic that processes data, what data is stored and used by the system, and how the user interface should work. The overall result is the system as a whole and how it performs, not how it is actually going to do it.

3.1.3 DESIGN

The software system design is produced from the results of the requirements phase. Architects have the ball in their court during this phase and this is the phase in which their focus lies. This is where the details on how the system will work is produced. Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.

3.1.4 IMPLEMENTATION

Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle. For a developer, this is the main focus of the life cycle because this is where the code is produced. Implementation my overlap with both the design and testing phases. Many tools exists (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

3.1.5 TESTING

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole. So in a nutshell, that is a very basic overview of the general software development life cycle model.

3.2 EXISTING SYSTEM

3.2.1 Existing Environment

Fig. 3.1 shows the block diagram of existing system. All the images are trained using the SVM network model with basic features, statistical and texture features. And random unknown test sample is applied to the system for detection and classification, respectively.

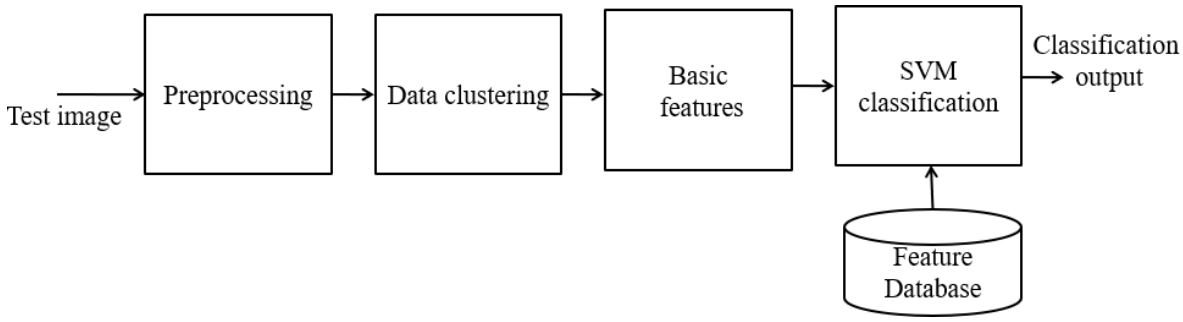


Fig. 3.2: Existing block diagram.

3.2.2 Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

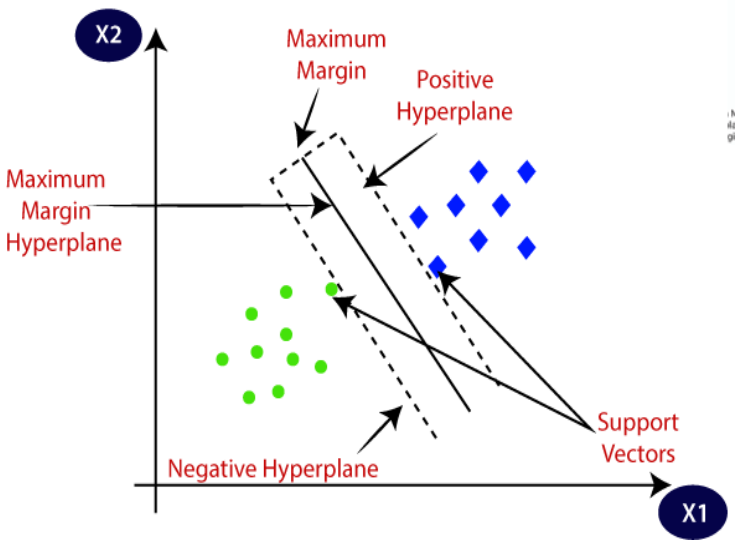


Fig. 3.2.2: Analysis of SVM

Example: SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. Because of the support vectors, it will classify it as a cat. Consider the below diagram:

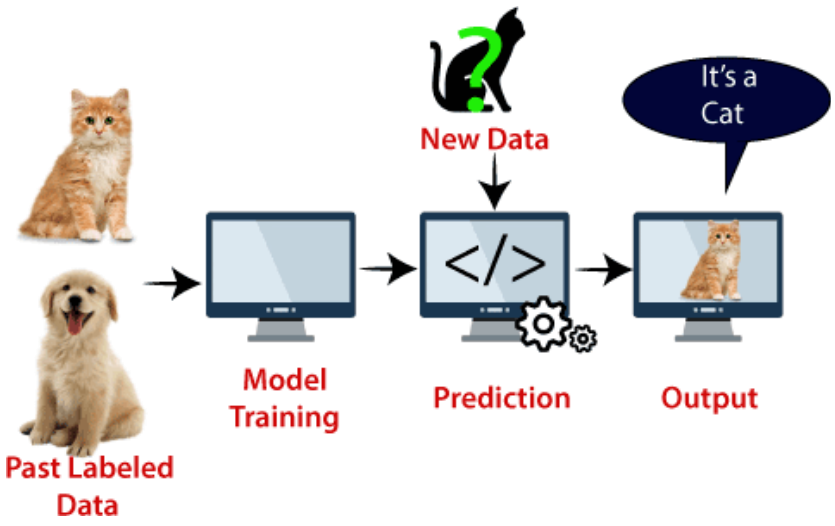


Fig. 3.2.3: Basic classification using SVM.

Types of SVM: SVM can be of two types:

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier

3.2.3 SVM working

Linear SVM: The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair (x1, x2) of coordinates in either green or blue. Consider the below image:

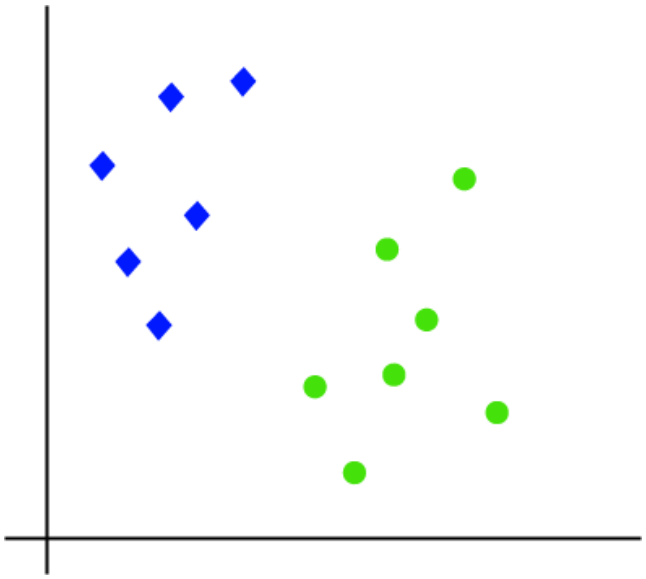


Fig. 3.2.4: Linear SVM.

So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

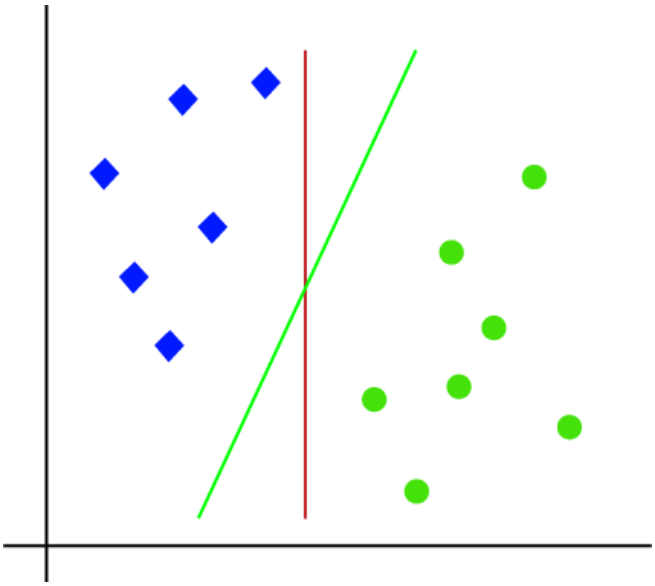


Fig. 3.2.5: Test-Vector in SVM.

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

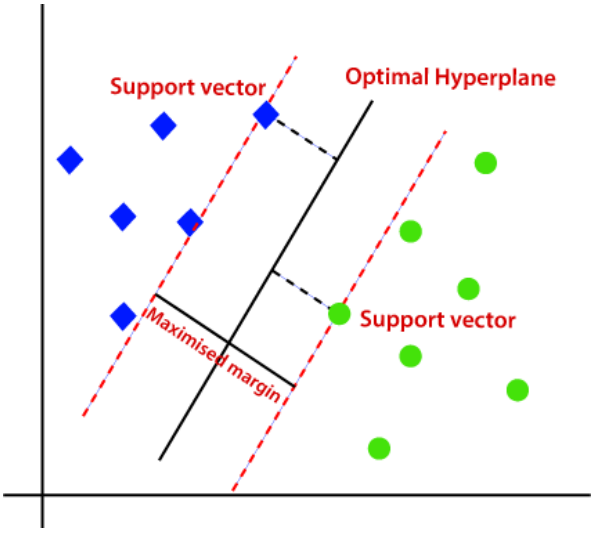


Fig. 3.2.6: Classification in SVM.

Non-Linear SVM: If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:

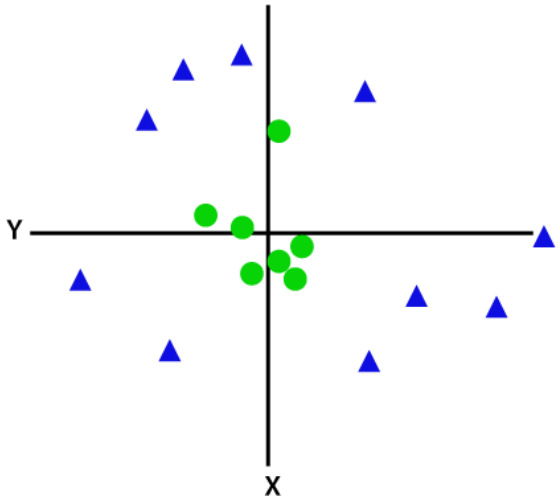


Fig. 3.2.7: Non-linear SVM.

So, to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third-dimension z. It can be calculated as:

$$z=x^2 +y^2$$

By adding the third dimension, the sample space will become as below image:

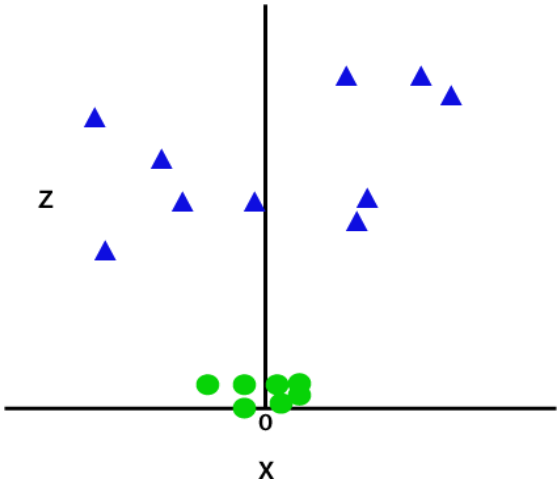


Fig. 3.2.8: Non-linear SVM data seperation.

So now, SVM will divide the datasets into classes in the following way. Consider the below image:

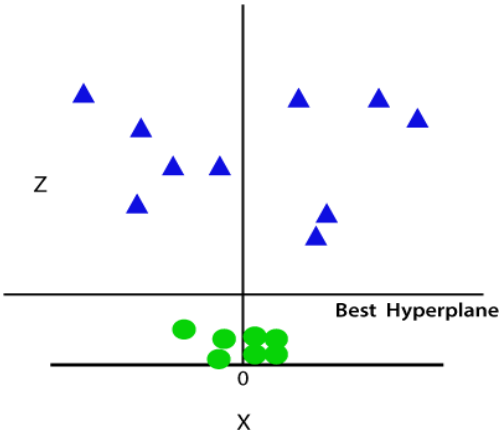


Fig. 3.2.9: Non-Linear SVM best hyperplane.

Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:

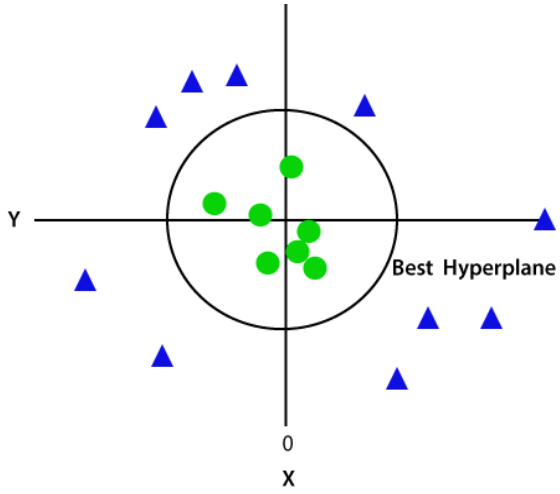


Fig. 3.2.10: Non-Linear SVM with ROC.

Hence, we get a circumference of radius 1 in case of non-linear data.

3.2.4 Disadvantages of support vector machine:

- Support vector machine algorithm is not acceptable for large data sets.
- It does not execute very well when the data set has more sound i.e. target classes are overlapping.

- In cases where the number of properties for each data point outstrips the number of training data specimens, the support vector machine will underperform.
- As the support vector classifier works by placing data points, above and below the classifying hyperplane there is no probabilistic clarification for the classification.

3.3 PROPOSED SYSTEM

3.3.1 Crop Disease Recognition Model

Agriculture is one of the most important sources for human sustenance on Earth. Not only does it provide the much necessary food for human existence and consumption but also plays a major vital role in the economy of the country. But Plant diseases have turned into a dilemma as it can cause significant reduction in both quality and quantity of agricultural products. Nowadays farmers are facing many crucial problems for getting better yield cause of rapid change in climate and unexpected level of insects, in order to get better yield, need to reduce the level of pest insect. Several millions of dollars are spent worldwide for the safety of crops, agricultural produce and good, healthy yield. It is a matter of concern to safeguard crops from Bio-aggressors such as pests and insects, which otherwise lead to widespread damage and loss of crops. In a country such as India, approximately 18% of crop yield is lost due to pest attacks every year which is valued around 90,000 million rupees. Conventionally, manual pest monitoring techniques, sticky traps, black light traps are being utilized for pest monitoring and detection in farms.

Manual pest monitoring techniques are time consuming and subjective to the availability of a human expert to detect the same. 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, stems 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. In general, there are two types of factors which can bring death and destruction to plants; living(biotic) and nonliving (abiotic) agents. Living agent's including insects, bacteria, fungi and viruses. Nonliving agents include extremes of temperature, excess moisture, poor light, insufficient nutrients, and poor soil pH and air pollutants.

In recent years, deep learning has made breakthroughs in the field of digital image processing, far superior to traditional methods. How to use deep learning technology to study plant diseases and pests' identification has become a research issue of great concern to researchers.

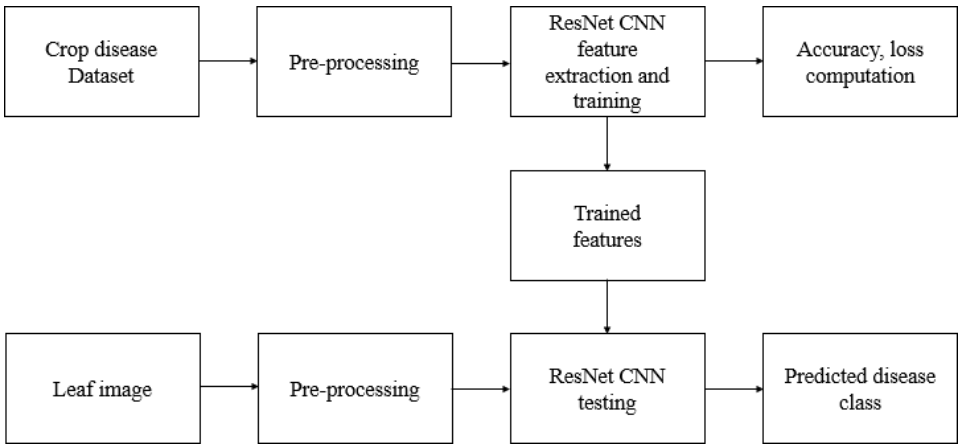


Fig. 3.3.1: Block diagram of proposed system.

Crop disease datasets are pre-processed and uploaded to Residual Network-CNN ((ResNet-CNN) for feature extraction. On the other hand, leaf images are also pre-processed and uploaded to ResNet CNN for testing. The leaf images and the crop disease datasets are compared to the trained features which are already trained with the plant diseases. The extracted features have some loss computation and accuracy. The comparison graph could predict the classes of the plant disease.

3.3.2 Crop disease dataset

The dataset totally contains 15 classes of crop diseases, such as pepper__bell___Bacterial_spot', 'Pepper__bell___healthy', 'Potato___Early_blight', 'Potato___healthy', 'Potato___Late_blight', 'Tomato_Target_Spot', 'Tomato__Tomato_mosaic_virus', 'Tomato_Tomato_YellowLeaf_Curl_Virus', 'Tomato_Bacterial_spot', 'Tomato_Early_blight', 'Tomato_healthy', 'Tomato_Late_blight', 'Tomato_Leaf_Mold','Tomato_Septoria_leaf_spot','Tomato_Spider_mites_Two_spotted_spider_mite. Here, Pepper, Potato, and Tomato are the major crop classes with different disease sub-types.

3.3.3 Image pre-processing

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analogue image processing. It allows a much wider range of algorithms to be applied to the input data — the aim of digital image processing is to

improve the image data (features) by suppressing unwanted distortions and/or enhancement of some important image features so that our AI-Computer Vision models can benefit from this improved data to work on. To train a network and make predictions on new data, our images must match the input size of the network. If we need to adjust the size of images to match the network, then we can rescale or crop data to the required size.

we can effectively increase the amount of training data by applying randomized augmentation to data. Augmentation also enables to train networks to be invariant to distortions in image data. For example, we can add randomized rotations to input images so that a network is invariant to the presence of rotation in input images. An augmented Image Datastore provides a convenient way to apply a limited set of augmentations to 2-D images for classification problems.

we can store image data as a numeric array, an ImageDatastore object, or a table. An ImageDatastore enables to import data in batches from image collections that are too large to fit in memory. we can use an augmented image datastore or a resized 4-D array for training, prediction, and classification. We can use a resized 3-D array for prediction and classification only.

There are two ways to resize image data to match the input size of a network. Rescaling multiplies the height and width of the image by a scaling factor. If the scaling factor is not identical in the vertical and horizontal directions, then rescaling changes the spatial extents of the pixels and the aspect ratio.

Cropping extracts a subregion of the image and preserves the spatial extent of each pixel. We can crop images from the center or from random positions in the image. An image is nothing more than a two-dimensional array of numbers (or pixels) ranging between 0 and 255. It is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically.

Resize image: In this step-in order to visualize the change, we are going to create two functions to display the images the first being a one to display one image and the second for two images. After that, we then create a function called processing that just receives the images as a parameter.

Need of resize image during the pre-processing phase, some images captured by a camera and fed to our AI algorithm vary in size, therefore, we should establish a base size for all images fed into our AI algorithms.

3.3.4 Proposed ResNet-CNN

Deep neural network is gradually applied to the identification of crop diseases and insect pests. Deep neural network is designed by imitating the structure of biological neural

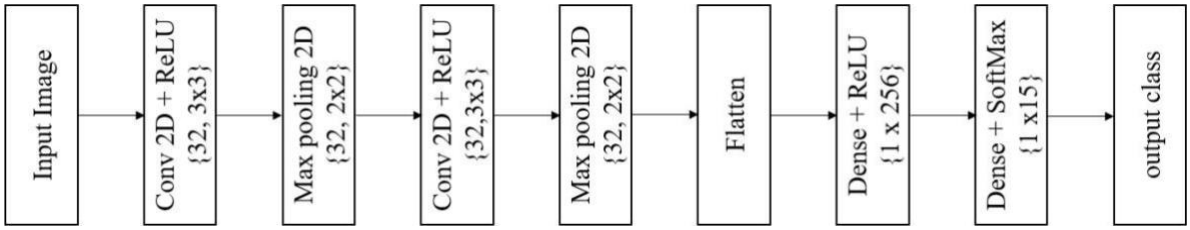


Fig. 3.3.4: Proposed ResNet-CNN.

Table. 3.3.4.1: Layers description.

Layer Names	No. of filters	Kernel size	Feature size
Conv 2D +ReLU	32	3 x 3	62x62x32
Max pooling 2D	-	3 x 3	31x31x32
Conv 2D+ReLU	32	3 x 3	29x29x32
Max pooling 2D	-	3 x 3	14x14x32
Flatten	-	1x6272	1x6272
Dense +ReLU		1 x 256	1 x 256
Dense + SoftMax		1 x 15	1 x 15

network, an artificial neural network to imitate the brain, using learnable parameters to replace the links between neurons. Convolutional neural network is one of the most widely used deep neural network structures, which is a branch of feed forward neural network. The success of AlexNet network model also confirms the importance of convolutional neural network model. Since then, convolutional neural networks have developed vigorously and have been widely used in financial supervision, text and speech recognition, smart home, medical diagnosis, and other fields.

Convolutional neural networks are generally composed of three parts. Convolution layer for feature extraction. The convergence layer, also known as the pooling layer, is mainly used for feature selection. The number of parameters is reduced by reducing the number of features. The full connection layer carries out the summary and output of the characteristics. A convolution layer is consisting of a convolution process and a nonlinear activation function ReLU. A typical architecture of CNN model for crop disease recognition is shown in Fig. 4.2.

The leftmost image is the input layer, which the computer understands as the input of several matrices. Next is the convolution layer, the activation function of which uses ReLU. The pooling layer has no activation function. The combination of convolution and pooling layers can be constructed many times. The combination of convolution layer and convolution layer or convolution layer and pool layer can be very flexibly, which is not limited when constructing the model. But the most common CNN is a combination of several convolution layers and pooling layers. Finally, there is a full connection layer, which acts as a classifier and maps the learned feature representation to the sample label space.

Convolutional neural network mainly solves the following two problems.

1) Problem of too many parameters: It is assumed that the size of the input picture is $50 * 50 * 3$. If placed in a fully connected feedforward network, there are 7500 mutually independent links to the hidden layer. And each link also corresponds to its unique weight parameter. With the increase of the number of layers, the size of the parameters also increases significantly. On the one hand, it will easily lead to the occurrence of over-fitting phenomenon. On the other hand, the neural network is too complex, which will seriously affect the training efficiency. In convolutional neural networks, the parameter sharing mechanism makes the same parameters used in multiple functions of

a model, and each element of the convolutional kernel will act on a specific position of each local input. The neural network only needs to learn a set of parameters and does not need to optimize learning for each parameter of each position.

2) Image stability: Image stability is the local invariant feature, which means that the natural image will not be affected by the scaling, translation, and rotation of the image size. Because in deep learning, data enhancement is generally needed to improve performance, and fully connected feedforward neural is difficult to ensure the local invariance of the image. This problem can be solved by convolution operation in convolutional neural network.

3.4 ResNet-CNN

According to the facts, training and testing of CNN involves in allowing every source data via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from.

Convolution layer is the primary layer to extract the features from a source image and maintains the relationship between pixels by learning the features of image by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source image $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of an image (here d=3 since the source image is RGB) and a filter or kernel with similar size of input image and can be denoted as $F(k_x, k_y, d)$..

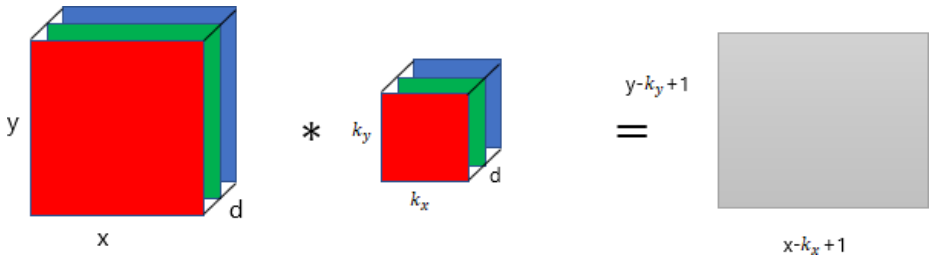
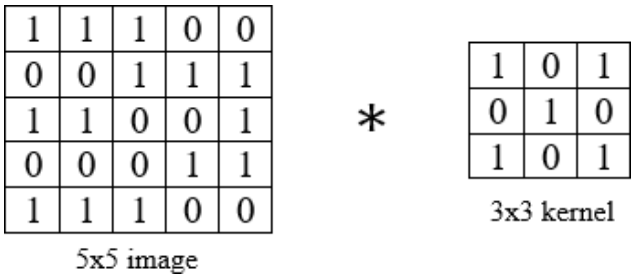


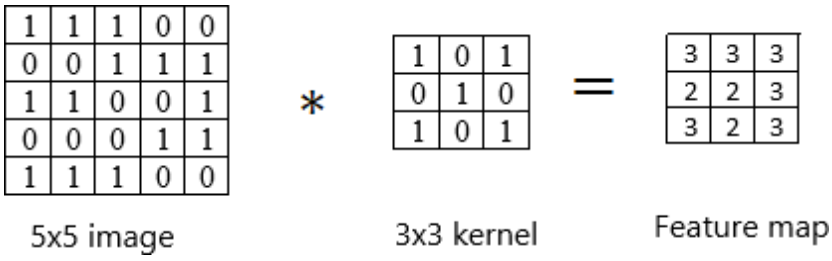
Fig. 4.3: Representation of convolution layer process.

The output obtained from convolution process of input image and filter has a size of $C\,((x - k_x + 1), (y - k_y + 1), 1)$, which is referred as feature map. Let us assume an

input image with a size of 5×5 and the filter having the size of 3×3. The feature map of input image is obtained by multiplying the input image values with the filter values.



(a)



(b)

Fig. 4.4: Example of convolution layer process (a) an image with size 5×5 is convolving with 3×3 kernel (b) Convolved feature map.

3.4.1 ReLU layer

Networks those utilizes the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows:

$$\mathcal{G}(x) = \max\{0, x\}$$

3.4.2 Max pooing layer

This layer mitigates the number of parameters when there are larger size images. This can be called as subsampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.

3.4.3 Software classifier

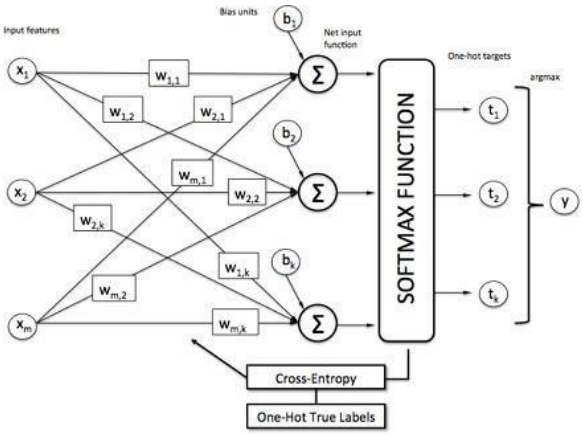


Fig. 4.5: Crop disease prediction using SoftMax classifier.

Generally, as seen in the above picture softmax function is added at the end of the output since it is the place where the nodes are meet finally and thus, they can be classified. Here, X is the input of all the models and the layers between X and Y are the hidden layers and the data is passed from X to all the layers and Received by Y. Suppose, we have 10 classes, and we predict for which class the given input belongs to. So, for this what we do is allot each class with a particular predicted output. Which means that we have 10 outputs corresponding to 10 different class and predict the class by the highest probability it has.

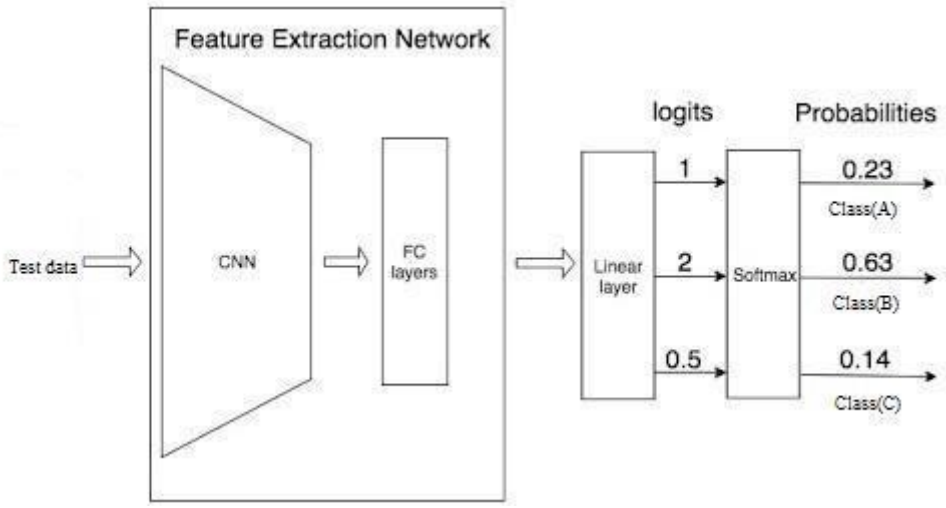


Fig. 4.6: Example of SoftMax classifier.

In Fig. 4.6, and we must predict what is the object that is present in the picture. In the normal case, we predict whether the crop is A. But in this case, we must predict what is the object that is present in the picture. This is the place where softmax comes in handy. As the model is already trained on some data. So, as soon as the picture is given, the model processes the pictures, send it to the hidden layers and then finally send to softmax for classifying the picture. The softmax uses a One-Hot encoding Technique to calculate the cross-entropy loss and get the max. One-Hot Encoding is the technique that is used to categorize the data. In the previous example, if softmax predicts that the object is class A then the One-Hot Encoding for:

Class A will be [1 0 0]

Class B will be [0 1 0]

Class C will be [0 0 1]

From the diagram, we see that the predictions are occurred. But generally, we don't know the predictions. But the machine must choose the correct predicted object. So, for machine to identify an object correctly, it uses a function called cross-entropy function.

So, we choose more similar value by using the below cross-entropy formula.

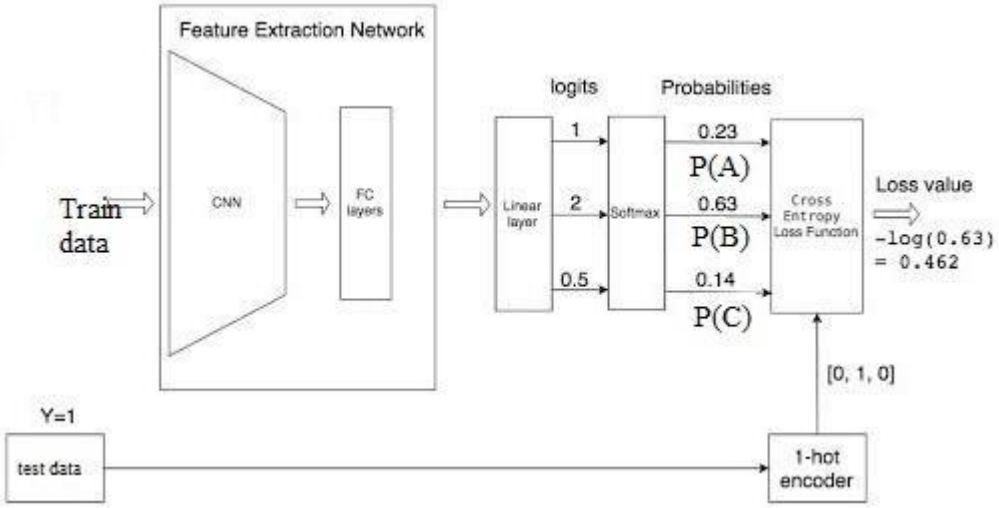


Fig. 4.7: Example of SoftMax classifier with test data.

In the above example we see that 0.462 is the loss of the function for class specific classifier. In the same way, we find loss for remaining classifiers. The lowest the loss function, the better the prediction is. The mathematical representation for loss function can be represented as: -

$$LOSS = np.sum(-Y * np.log(Y\_pred))$$

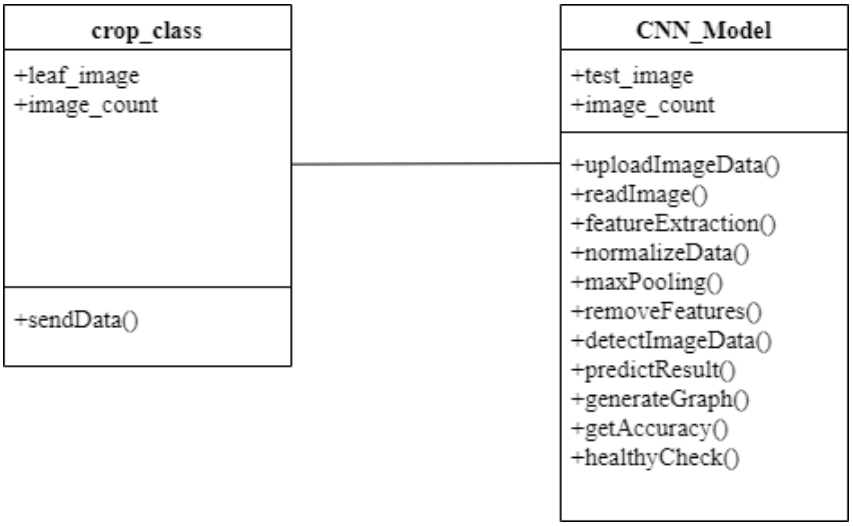
3.4.4 Advantages of proposed system

- CNNs do not require human supervision for the task of identifying important features.
- They are very accurate at image recognition and classification.
- Weight sharing is another major advantage of CNNs.
- Convolutional neural networks also minimize computation in comparison with a regular neural network.
- CNNs make use of the same knowledge across all image locations.

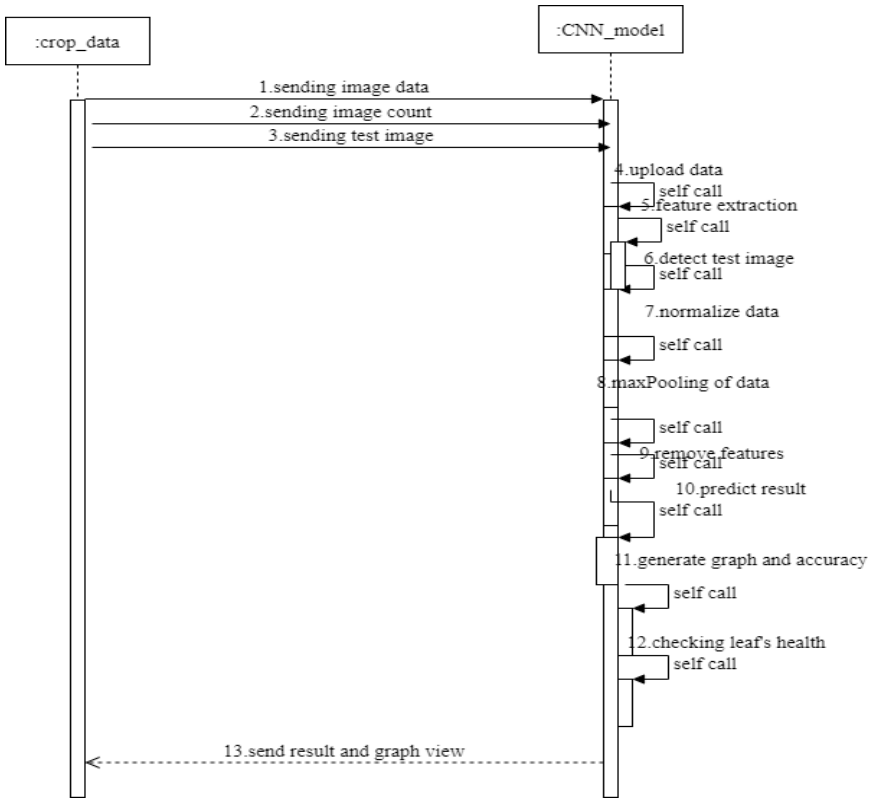
4. SYSTEM DESIGN

4.1 UML DIAGRAMS

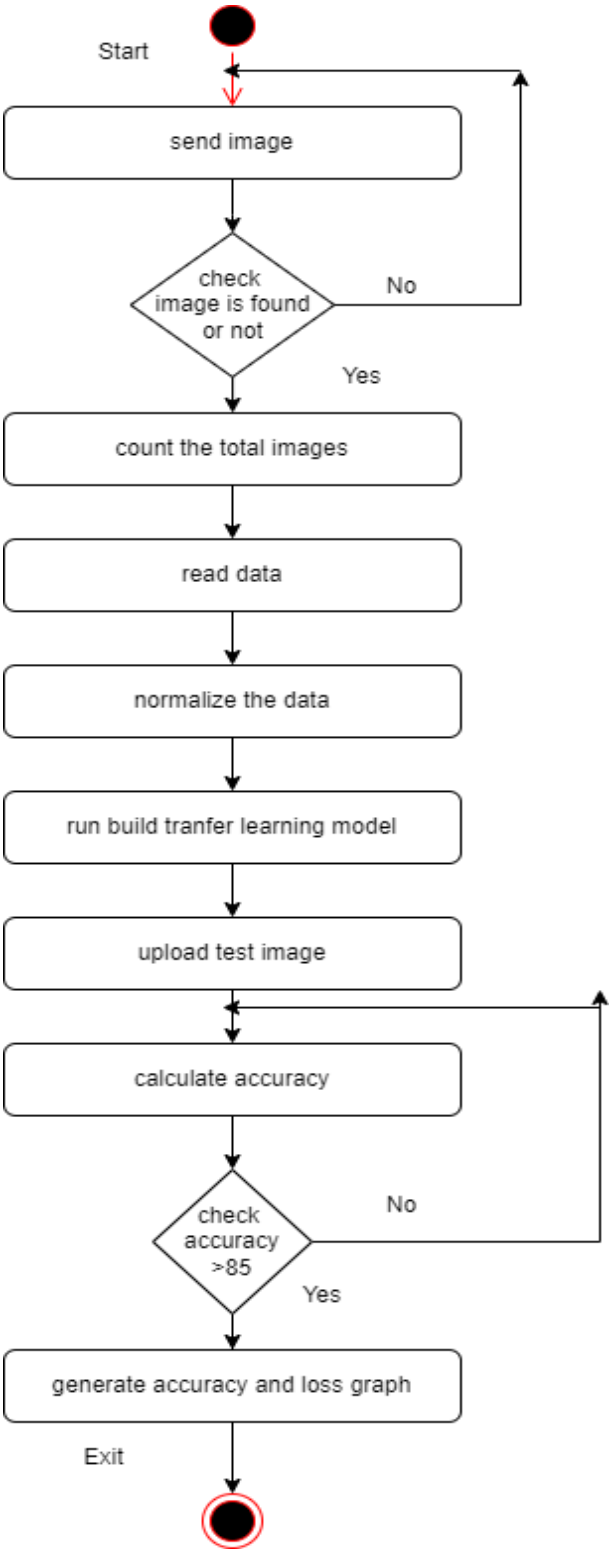
4.1.1 Class Diagram: Class diagram is a static diagram. It represents the static view of an application.



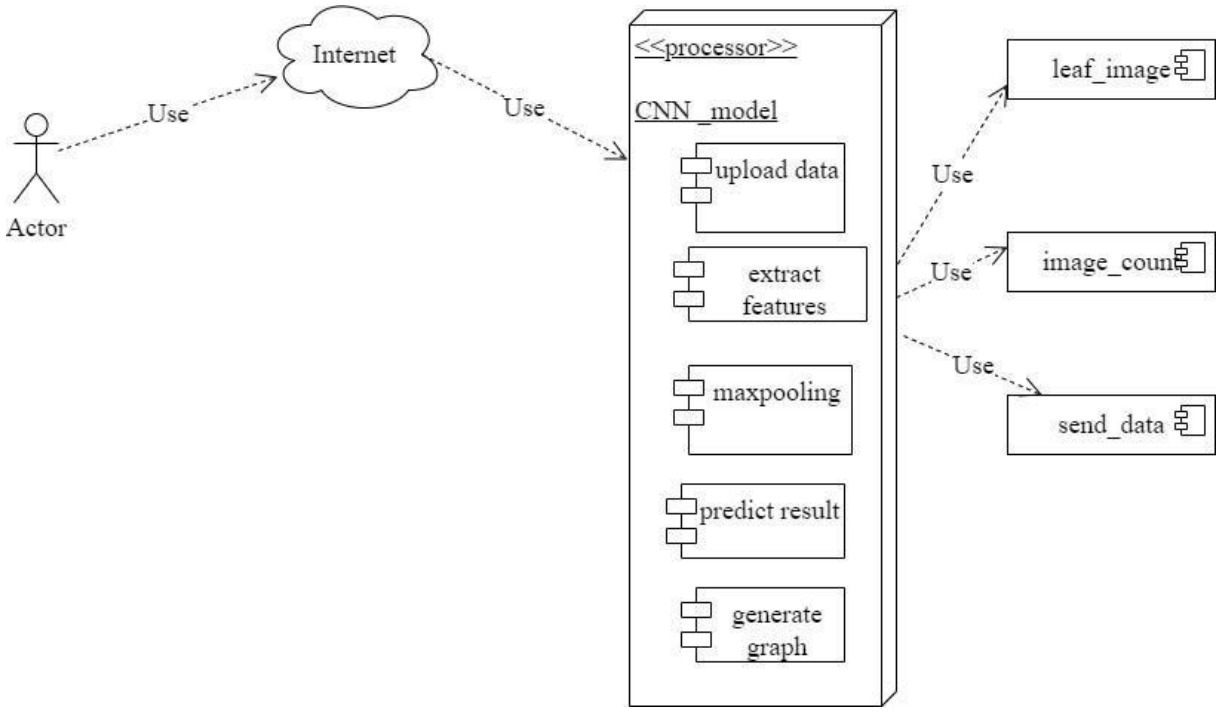
4.1.2 Sequence Diagram: Sequence diagrams are used to capture the order of messages flowing from one object to another.



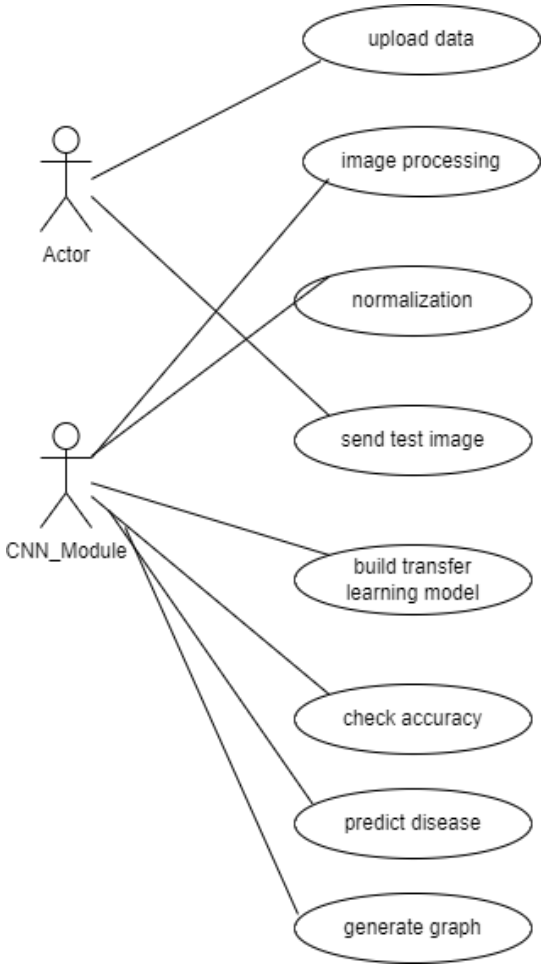
4.1.3 Activity diagram: Activity diagram is basically a flowchart to represent the flow from one activity to another activity.



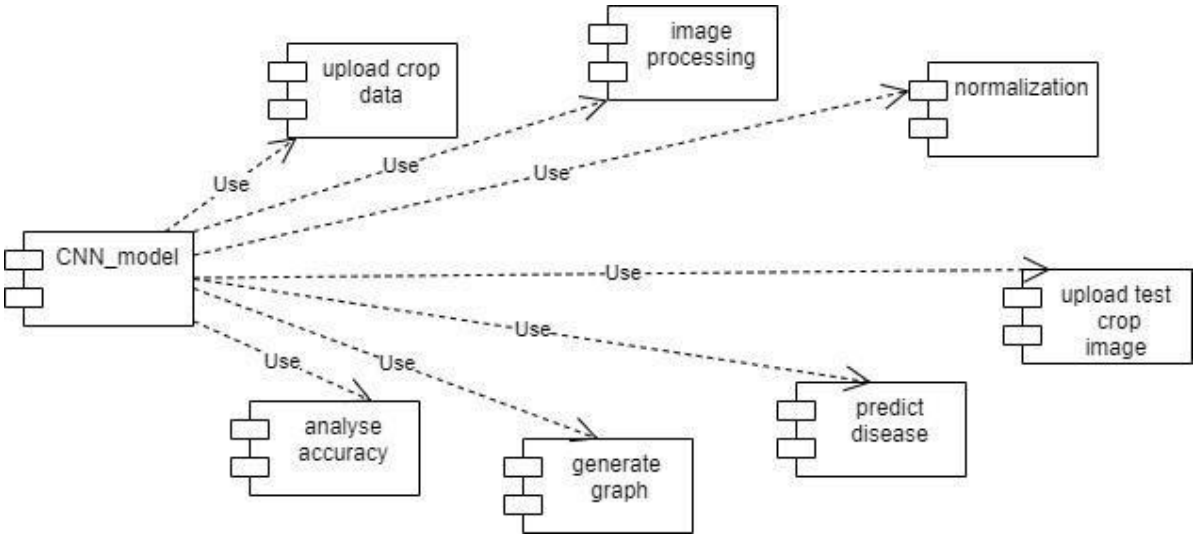
4.1.4 Deployment diagram: Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.



4.1.5 Use Case Diagram: Use case diagrams are used to gather the requirements of a system including internal and external influences.



4.1.6 Component diagram: Component diagrams are used to describe the physical artifacts of a system.



5. SYSTEM ENVIRONMENT

5.1 MACHINE LEARNING

What is Machine Learning

Before we look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Categories of Machine Leaning

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label and is often described as "letting the dataset speak for itself." These models

include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate, and solve complex problems. On the other side, AI is still in its initial stage and have not surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Challenges in Machines Learning

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

1. Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

2. Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

3. Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

4. No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.
5. Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.
6. Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.
7. Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML

- Emotion analysis
- Sentiment analysis
- Error detection and prevention
- Weather forecasting and prediction
- Stock market analysis and forecasting
- Speech synthesis
- Speech recognition
- Customer segmentation
- Object recognition
- Fraud detection
- Fraud prevention
- Recommendation of products to customer in online shopping

How to Start Learning Machine Learning?

Arthur Samuel coined the term "Machine Learning" in 1959 and defined it as a "Field of study that gives computers the capability to learn without being explicitly programmed".

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of $146,085 per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let's get started!!!

How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don't know these, never fear! You don't need a Ph.D. degree in these topics to get started but you do need a basic understanding.

(a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

(b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as Fork Python available Free on GeeksforGeeks.

Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

(a) Terminologies of Machine Learning

- Model – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
- Feature – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
- Target (Label) – A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
- Training – The idea is to give a set of inputs(features) and it's expected outputs(labels),

so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

- Prediction – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

(b) Types of Machine Learning

- Supervised Learning – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
- Unsupervised Learning – This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- Semi-supervised Learning – This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- Reinforcement Learning – This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of Machine Learning

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed.

5.2 Python

Below are some facts about Python.

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python

Let's see how Python dominates over other languages.

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

8. Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

9. Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you withyour tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.

The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatestachievement of ABC was to influence the design of Python. Python was conceptualizedin the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido vanRossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC partsthat I liked. I created a basic syntax, used indentation for statement grouping instead ofcurly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Python Development Steps

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

- Print is now a function.
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g., a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e., int. long is int as well.
- The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

Purpose

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

6. System implementations

6.1 Modules Used in Project

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load,

prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the

interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how

to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

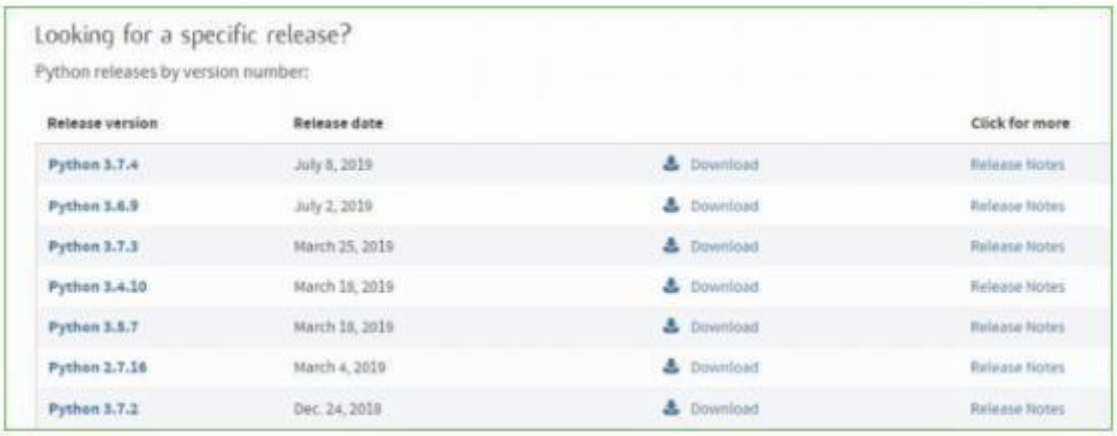
Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7f99ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c1ecaa5ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b243f	28082845	SIG
Windows help file	Windows		d6399957a2c068b2ac58cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ee9b9abe03184a43f28a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76deb6d6304a1a583e563400	26885368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c9088bd71ae8e513a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab38d1f8041879fda9413357413fd8	6741628	SIG
Windows x86 executable installer	Windows		33cc802f942a54446a3d8451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa6d117df82c30983ea371d87c	1324608	SIG

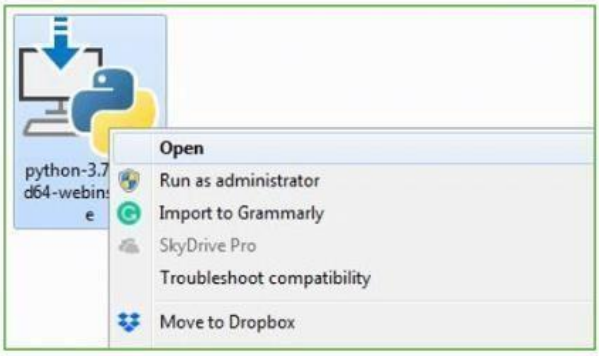
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

Step 3: Click on Install NOW After the installation is successful. Click on Close.

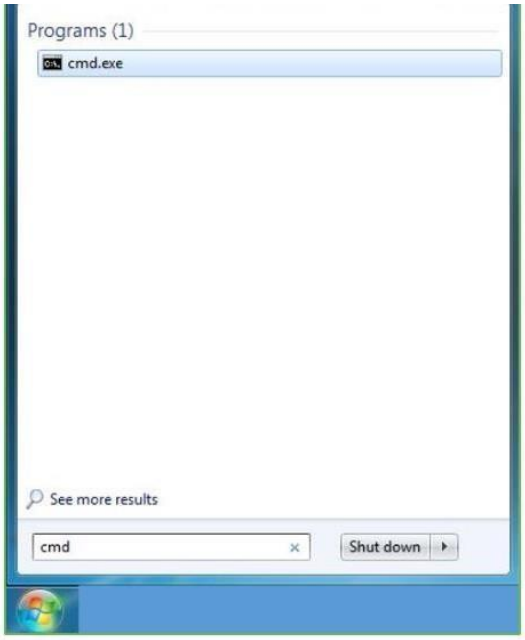
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

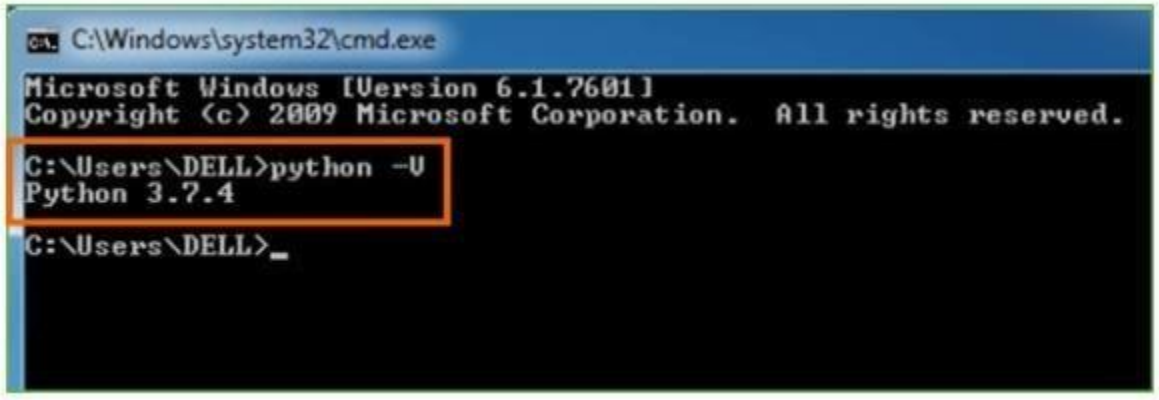
Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.



Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

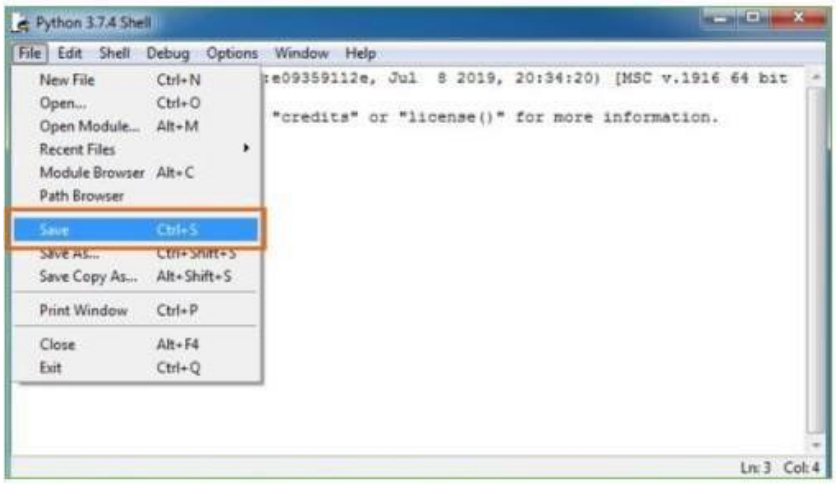
Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.

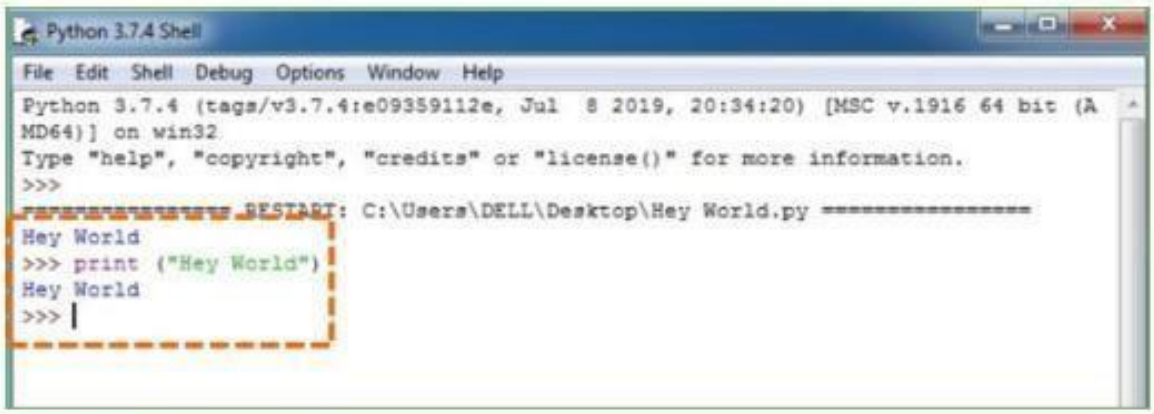
Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print ("Hey World") and Press Enter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python does not need semicolons at the end of the statements otherwise it won't work.

6.2 CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
import matplotlib.pyplot as plt
import numpy as np
from tkinter import ttk
from tkinter import filedialog
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.layers.core import Dense,Activation,Dropout, Flatten
from sklearn.metrics import accuracy_score
import os
import cv2
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
import pickle
from keras.models import model_from_json
main = Tk()
main.title("Plant Disease Detection and Classification with Pesticide Suggestion for e-Agriculture Applications")
```

```
main.geometry("1300x1200")

global filename

global X, Y

global model

global accuracy

plants = ['Pepper_bell_Bacterial_spot', 'Pepperbell_healthy',
'Potato_Early_blight', 'Potato_healthy', 'Potato_Late_blight',
'Tomato_Target_Spot',

    'Tomato_Tomato_mosaic_virus',
'TomatoTomato_YellowLeaf_Curl_Virus', 'Tomato_Bacterial_spot',
'Tomato_Early_blight', 'Tomato_healthy',

    'Tomato_Late_blight', 'Tomato_Leaf_Mold',
'Tomato_Septoria_leaf_spot',
'Tomato_Spider_mites_Two_spotted_spider_mite']

pesticide = ['Seed Treatment with Hot Water', 'No Pesticide is required',
'Proxanil', 'No Pesticide is required', 'Proxanil', 'Insecticidial Soap and
Horticultural Oil',

    'Insecticidial Soap and Horticultural Oil', 'Mancozeb', 'Insecticidial
Soap and Horticultural Oil', ' Bonide tomato', 'No Pesticide is required',

    'Copper based fungicide', 'Applying fungicides', 'Mancozeb',
'Insecticidial Soap and Horticultural Oil']

def uploadDataset():

   global X, Y

   global filename
```

```
    text.delete('1.0', END)
    filename = filedialog.askdirectory(initialdir=".")
    text.insert(END,'dataset loaded\n')
def imageProcessing():
    text.delete('1.0', END)
    global X, Y
    X =  np.load("model/myimg_data.txt.npy")
    Y = np.load("model/myimg_label.txt.npy")
    Y = to_categorical(Y)
    X = np.asarray(X)
    Y = np.asarray(Y)
    X = X.astype('float32')
    X = X/255
    indices = np.arange(X.shape[0])
    np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    text.insert(END,'image processing completed\n')
    img = X[20].reshape(64,64,3)
    cv2.imshow('ff',cv2.resize(img,(250,250)))
    cv2.waitKey(0)
```

```
def cnnModel():
    global model
    global accuracy
    text.delete('1.0', END)
    if  os.path.exists('model/model.json'):
        with open('model/model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            model = model_from_json(loaded_model_json)
        json_file.close()
        model.load_weights("model/model_weights.h5")
        model._make_predict_function()
        print(model.summary())
        f = open('model/history.pckl', 'rb')
        accuracy  = pickle.load(f)
        f.close()
        acc = accuracy['accuracy']
        acc = acc[9] * 100
        text.insert(END,"CNN Crop Disease Recognition Model Prediction Accuracy = "+str(acc))
    else:
        model = Sequential() #resnet transfer learning code here
```

```
model.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))

model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Convolution2D(32, 3, 3, activation = 'relu'))

model.add(MaxPooling2D(pool_size = (2, 2)))

model.add(Flatten())

model.add(Dense(output_dim = 256, activation = 'relu'))

model.add(Dense(output_dim = 15, activation = 'softmax'))

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])

print(model.summary())

hist = model.fit(X, Y, batch_size=16, epochs=10, validation_split=0.2, shuffle=True, verbose=2)

model.save_weights('model/model_weights.h5')

model_json = model.to_json()

with open("model/model.json", "w") as json_file:

   json_file.write(model_json)

json_file.close()

f = open('model/history.pckl', 'wb')

pickle.dump(hist.history, f)

f.close()

f = open('model/history.pckl', 'rb')
```

```
        accuracy = pickle.load(f)
        f.close()
        acc = accuracy['accuracy']
        acc = acc[9] * 100
        text.insert(END,"Transfer Learning Model Accuracy = "+str(acc))
def predict():
    global model
    filename = filedialog.askopenfilename(initialdir="testImages")
    img = cv2.imread(filename)
    img = cv2.resize(img, (64,64))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1,64,64,3)
    test = np.asarray(im2arr)
    test = test.astype('float32')
    test = test/255
    preds  =  model.predict(test)
    predict  =  np.argmax(preds)
    img = cv2.imread(filename)
    img = cv2.resize(img, (800,400))
    cv2.putText(img, 'Suggested pesticide is : '+pesticide[predict], (10,
250),  cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 0, 255), 2)
```

```
    cv2.putText(img, 'Crop Disease Recognize as : '+plants[predict], (10, 25),  cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 255, 0), 2)
    cv2.imshow('Crop Disease Recognize as : '+plants[predict], img)
    cv2.waitKey(0)
def graph():
    acc = accuracy['accuracy']
    loss = accuracy['loss']
    plt.figure(figsize=(10,6))
    plt.grid(True)
    plt.xlabel('Iterations')
    plt.ylabel('Accuracy/Loss')
    plt.plot(acc, 'ro-', color = 'green')
    plt.plot(loss, 'ro-', color = 'blue')
    plt.legend(['Accuracy', 'Loss'], loc='upper left')
    #plt.xticks(wordloss.index)
    plt.title('Iteration Wise Accuracy & Loss Graph')
    plt.show()
def close():
    main.destroy()
    #text.delete('1.0', END)
font = ('times', 15, 'bold')
```

```
title = Label(main, text='Plant Disease Detection and Classification with Pesticide Suggestion for e-Agriculture Applications')

title.config(bg='powder blue', fg='olive drab')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

ff = ('times', 12, 'bold')

uploadButton = Button(main, text="Upload Crop Disease Dataset", command=uploadDataset)

uploadButton.place(x=20,y=100)

uploadButton.config(font=ff)

processButton = Button(main, text="Image Processing & Normalization", command=imageProcessing)

processButton.place(x=20,y=150)

processButton.config(font=ff)

modelButton = Button(main, text="Build Transfer Learning Model", command=cnnModel)

modelButton.place(x=20,y=200)

modelButton.config(font=ff)

predictButton = Button(main, text="Upload Test Image & Predict Disease", command=predict)

predictButton.place(x=20,y=250)
```

```
predictButton.config(font=ff)

graphButton = Button(main, text="Accuracy & Loss Graph", command=graph)

graphButton.place(x=20,y=300)

graphButton.config(font=ff)

exitButton = Button(main, text="Exit", command=close)

exitButton.place(x=20,y=350)

exitButton.config(font=ff)

font1 = ('times', 12, 'bold')

text=Text(main,height=30,width=85)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=450,y=100)

text.config(font=font1)

main.config(bg='light blue')

main.mainloop()
```

7. SYSTEM REQUIREMENTS

7.1 Hardware Requirements:

System	: Pentium IV 2.4 GHz.
Hard Disk	: 40 GB.
Floppy Drive	: 1.44 Mb.
Monitor	: 15 VGA Colour.
Mouse	: Logitech.
Ram	: 512 Mb.

7.2 Software Requirements:

Operating System	: Windows
Coding Language	: Python 3.7

7.3 FUNCTIONAL REQUIREMENTS

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

OUTPUT DEFINITION

The outputs should be defined in terms of the following points:

1. Type of the output
2. Content of the output
3. Format of the output
4. Location of the output
5. Frequency of the output
6. Volume of the output
7. Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES:

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation

- Data correction

INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

INPUT MEDIA:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

ERROR AVOIDANCE

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

ERROR DETECTION

Even though every effort is make to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

DATA VALIDATION

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

USER INTERFACE DESIGN

It is essential to consult the system users and discuss their needs while designing the user interface:

USER INTERFACE SYSTEMS CAN BE BROADLY CLASIFIED AS:

1. User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
2. Computer initiated interfaces

In the computer initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

USER_INITIATED INTERGFACES

User initiated interfaces fall into tow approximate classes:

1. Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
2. Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms oriented interface is chosen because it is the best choice.

COMPUTER-INITIATED INTERFACES

The following computer – initiated interfaces were used:

1. The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
2. Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

ERROR MESSAGE DESIGN:

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed.

This application must be able to produce output at different modules for different inputs.

PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application.

Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system
- The existing system is completely dependent on the user to perform all the duties.

8. SYSTEM TESTING

8.1 TESTING METHODOLOGIES

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, sub assemblies, assemblies 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 test. Each test type addresses a specific testing requirement.

8.2 TYPES OF TESTING

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.

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.

Functional test

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.

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.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests…

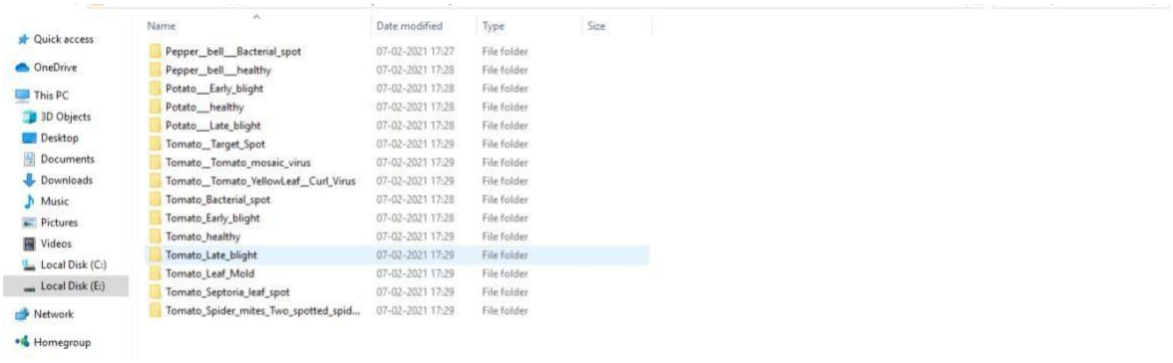
8. RESULTS

Modules

1) Upload Crop Disease Dataset: This module is used to select the dataset.
2) Image Processing & Normalization: The image preprocessing and normalization of dataset is achieved by this module.
3) Build Crop Disease Recognition Model: Either selection of trained model or retraining of module is achieved by this module.
4) Upload Test Image & Predict Disease: This module is used to identify the disease class from the test image.
5) Accuracy & Loss Graph: This module is used to plot the accuracy and loss comparison graph various iterations (epochs).

Discussions

To implement this project, we have used crop disease recognition dataset and this dataset saved inside 'CropDiseaseDataset' folder and below screen shots showing various type of crop disease images.



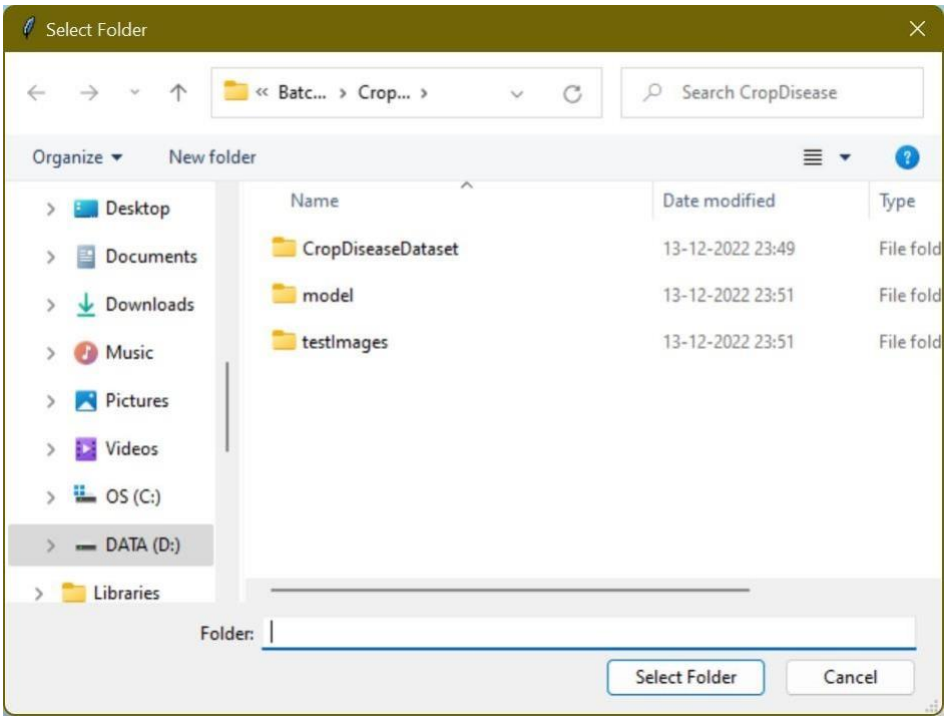
In above screen we are using 15 different type of crop images and each folder contains images of own leaf and in below screen you can see those image. You too just go inside any above folder to see images.


0a3f2927-4410-46a3-bfda-5f4769a5aaf8__JR_HL 8275.JPG
0ade14b6-8937-43ea-93eb-98343af6bae7__JR_HL 8026.JPG
0b76f650-27cf-4b62-b3ad-c97d81e0db0c__JR_HL 8554.JPG
0b796b7a-dfc8-47f7-9970-8373263c6408__JR_HL 5948.JPG
0ba474dd-0cfd-4fd2-a58c-8e3d18dbe7c3__JR_HL 8395.JPG
0bb97c36-159d-4ee2-8b06-1fbf3f533af5__JR_HL 8345.JPG
0c93ef1a-ccc8-4a62-9353-308e7da3ac3e__JR_HL 8245.JPG
0c102dad-d31f-469b-89d1-6fe7bf93708c__JR_HL 7689.JPG
0c002246-262c-46d7-9995-bd1e1b2737d9__JR_HL 7969.JPG
0cc984a1-ea7d-4f97-9967-5ca1be71c932__JR_HL 8803.JPG
0d4e35c0-f6d6-4810-9522-14e1d491a05d__JR_HL 8673.JPG
0d4f19fe-9e85-4689-9c99-0b01a58db9f4__JR_HL 8032.JPG
0dd09705-5593-4d01-9a9b-c1d5857daf92__JR_HL 8212.JPG
0de13a24-e542-45a2-be5a-8db7d8d858db__JR_HL 5982.JPG
0e2afa6e-568c-4694-babd-dc15057b53b3__JR_HL 8475.JPG
0e05c3e0-7766-4049-b7a8-563b5c028c33__JR_HL 8828.JPG
0e5d9ac1-4de8-491e-ab95-809850866805__JR_HL 8534.JPG
0e043e9a-7652-4e7a-aef1-0caa0c085f55__JR_HL 8479.JPG
0e69c47d-72c6-4fc6-9437-910c95b183dc__JR_HL 8113.JPG
0e526c7b-72ee-4547-916b-5cba013097b6__JR_HL 7735.JPG
0e1942a4-91d7-460a-8189-1a4e693cfc84__JR_HL 8896.JPG
0eabc3c2-d492-4227-90d8-14dab9fd4a9a__JR_HL 8699.JPG
0eb476b9-8b21-4b24-adef-7db813abbca3__JR_HL 7926.JPG
0eeb924f-88db-44e6-a278-e016fa2d25a4__JR_HL 8016.JPG
0f56ffda-5d57-4a61-8f7f-37f22a02520a__JR_HL 8589.JPG
0f4823db-878f-4d55-8d0f-ef011243dfda__JR_HL 8436.JPG
0fbb1d8d-63ff-406f-9abe-d555e4fb2de9__JR_HL 5877.JPG
01d03cb9-4505-458c-a6dc-648dedccd71a__JR_HL 8474.JPG
01dd93b0-0e34-447b-87ea-ccc9f2b62d03__JR_HL 8005.JPG
01fbd010-0cc1-4c48-98bc-49e328bf9bbc__JR_HL 8584.JPG
1a1a389d-f186-4
1a93f310-cd75-4
1a482d0c-d3a9-4
1a629a1c-1000-4
1aa25bab-008c-4
1aaaa11e-ec45-4
1b2c7e1e-2f0a-4
1b4b5b1f-9390-4
1b09df7a-eb13-4
1b71a13e-9f6d-4

SCREEN SHOTS

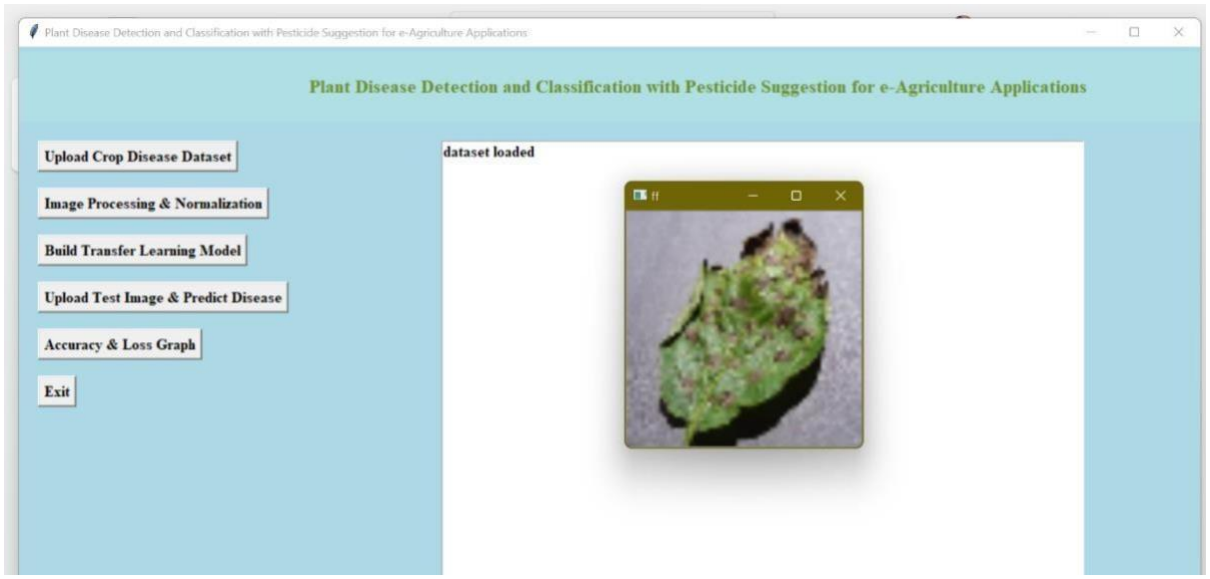
To run project double click on ‘run.bat’ file to get below screen

In above screen click on ‘Upload Crop Disease Dataset’ button to upload dataset images



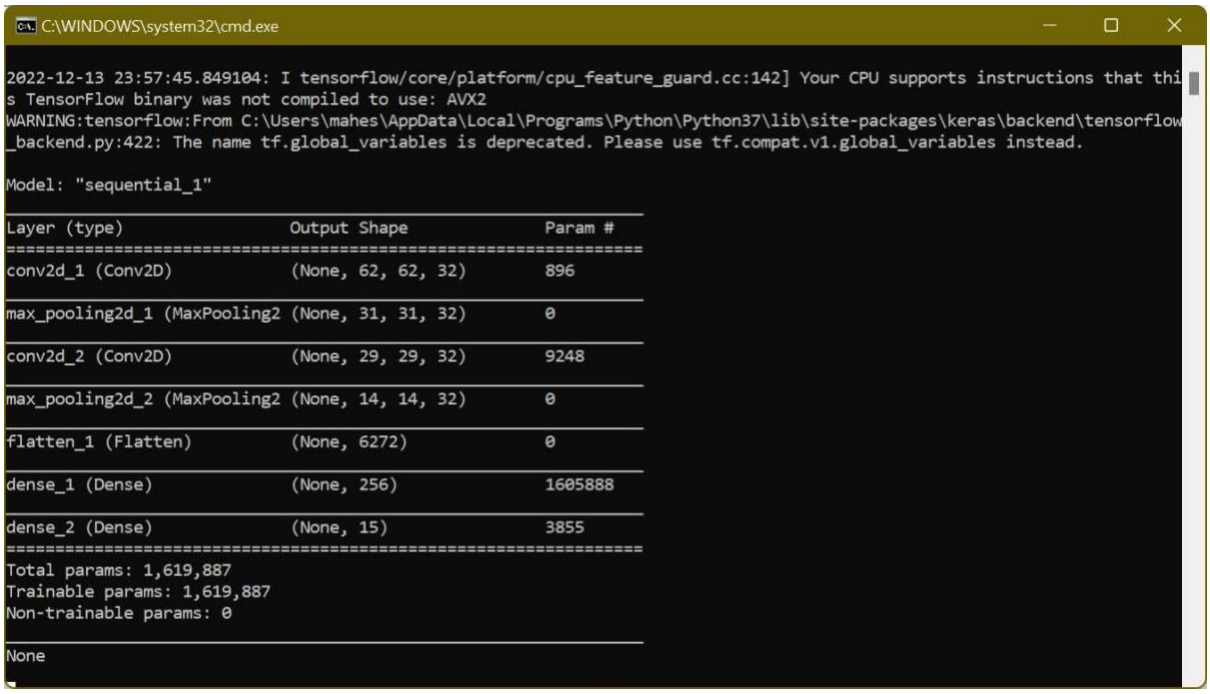
In above screen selecting and uploading ‘CropDiseaseDataset’ folder and then click on ‘Select Folder’ button to load dataset and to get below screen

In above screen dataset loaded and now click on 'Image Processing & Normalization' button to read all images and then process images to normalize by converting each image pixel value between 0 and 1 and for that normalization we will divide image pixels with 255 and then get value as 0 or 1 as all image's pixel value will be between 0 to 255.

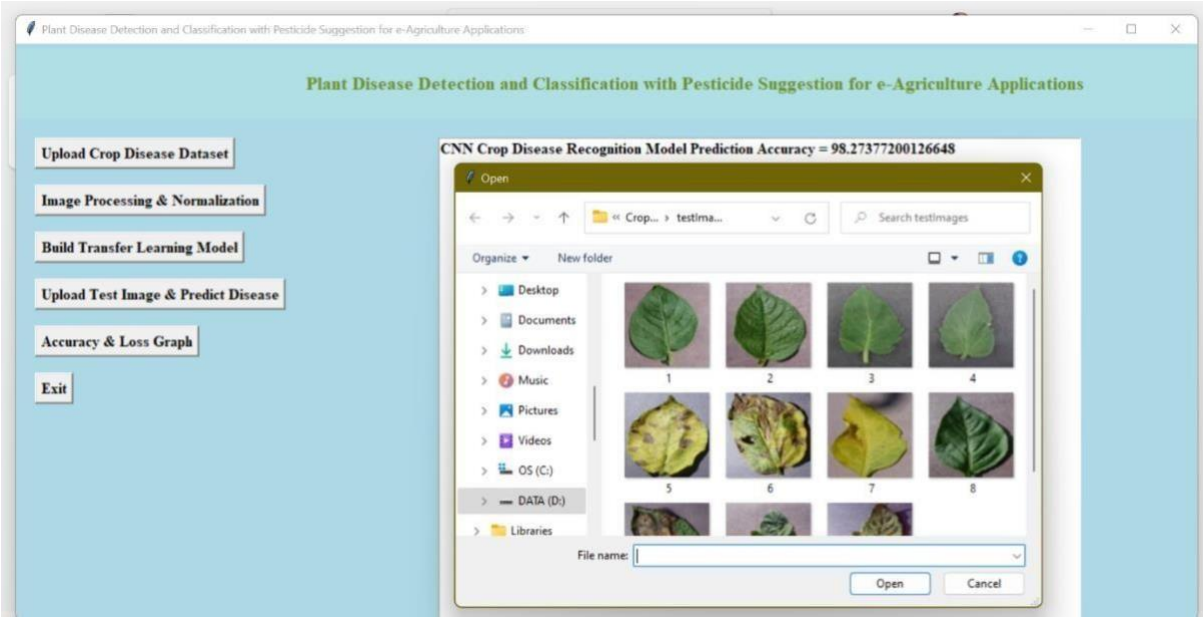


In above screen after applying normalization, we are just displaying one random image from dataset to check whether images loaded and process properly. Now dataset images are ready and now click on 'Build Transfer Learning Model' button to build CNN model

In above screen CNN model generated and its prediction accuracy is 98.27% and in below console screen we can see all CNN layers details



In above screen we can see we have used CONV2D, MAXPOOLING, FLATTEN and DENSE layer to build crop disease recognition model and RELU details you can see in code. Now model is ready and now click on 'Upload Test Image & Predict Disease' button to upload any test image and then application will predict disease or healthy from that image



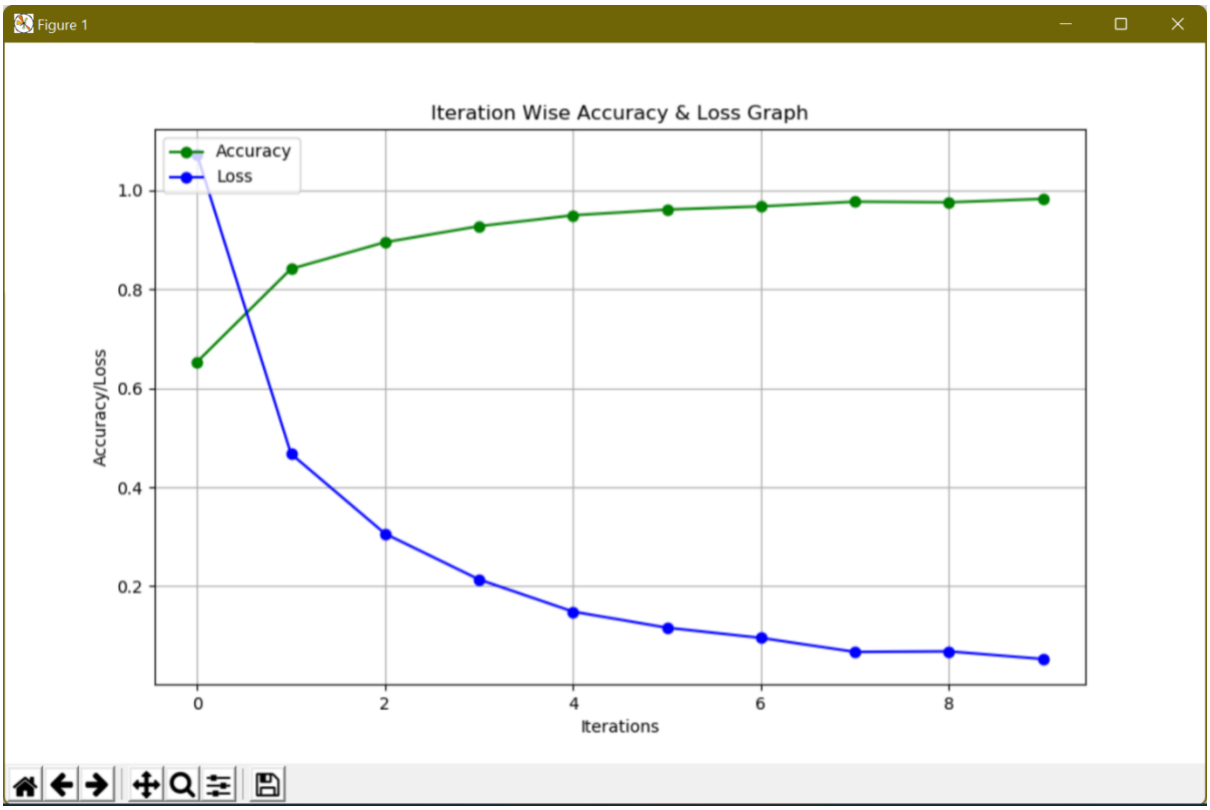
In above screen selecting and uploading '1.JPG' image file and then click on 'Open' button to get below prediction result

In above screen potato leaf predicted as healthy and pesticide suggested as no pesticide is required now try with other images.


Crop Disease Recognize as : Potato__Early_blight
Crop Disease Recognize as : Potato___Early_blight
Suggested pesticide is : Proxanil


Crop Disease Recognize as : Tomato_Bacterial_spot
Crop Disease Recognize as : Tomato_Bacterial_spot
Suggested pesticide is : Insecticidial Soap and Horticultural Oil

Now click on ‘Accuracy & Loss Graph’ button to get below graph.



In above graph x-axis represents epoch/iterations and y-axis represents accuracy/loss and green line represents accuracy and blue line represents loss and from above graph we can see with each increasing iteration accuracy is getting better and better and loss getting decrease.

9. CONCLUSION

In this work 15 kinds of crop diseases were studied. The model is constructed by using deep learning theory and ResNet-CNN technology. Experiments show that the model can effectively identify the data set, and the overall recognition accuracy is as high as 98.23%. The results show that the recognition accuracy of this hybrid network model is relatively higher than the traditional model, and it can be effectively applied to the identification and detection of plant diseases.

10. FUTURE SCOPE

In the future work, there are two directions should be improved, they are extended data set and optimized model. There are 27 diseases with 10 crop species dataset is available, and other species and diseases were not involved, such as rice and wheat, and their related diseases. Therefore, the next step is to obtain more crop species and disease images for research. This model has achieved good recognition accuracy and is worthy of further study and optimization. At the same time, we should design a network model which can classify crop images with higher accuracy.

REFERENCES

[1] J. Schmidhuber, ‘‘Deep learning in neural networks: An overview,’’ Neural Netw., vol. 61, pp. 85–117, Jan. 2015.

[2] A.-R. Mohamed, G. E. Dahl, and G. Hinton, ‘‘Acoustic modeling using deep belief networks,’’ IEEE Trans. Audio, Speech, Lang. Process., vol. 20, no. 1, pp. 14–22.

[3] S. Datir and S. Wagh, ‘‘Monitoring and detection of agricultural disease using wireless sensor network,’’ Int. J. Comput. Appl., vol. 87, no. 4, pp. 1–5, Feb. 2014.

[4] O. López, M. Rach, H. Migallon, M. Malumbres, A. Bonastre, and J. Serrano, ‘‘Monitoring pest insect traps by means of low-power image sensor technologies,’’ Sensors, vol. 12, no. 11, pp. 15801–15819, Nov. 2020.

[5] G. Athanikar and M. P. Badar, ‘‘Potato leaf diseases detection and classification system,’’ Int. J. Comput. Sci. Mobile Comput., vol. 5, no. 2, pp. 76–88, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classification with deep convolutional neural networks,’’ in Proc. Int. Conf. Neural Inf. Process. Syst. RedHook, NY, USA: Curran Associates, pp. 1097–1105.

[7] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 770–778.

[8] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, ‘‘A comparative study of fine-tuning deep learning models for plant disease identification,’’ Comput. Electron.Agricult., vol. 161, pp. 272–279, Jun. 2019.

[9] S. P. Mohanty, D. P. Hughes, and M. Salathé, ‘‘Using deep learning for image-based plant disease detection,’’ Frontiers Plant Sci., vol. 7, p. 1419, Sep. 2016.

[10] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, ‘‘Deep neural networks-based recognition of plant diseases by leaf image classification,’’ in Proc. Comput. Intell. Neurosci., May 2016, Art. no. 3289801.

[11] Y. Lecun, L. Bottou, and Y. Bengio, and P. Haffner, ‘‘Gradient-based learning applied to document recognition,’’ Proc. IEEE, vol. 86, no. 11, pp. 288–319, Nov. 2018.

[12] Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, ‘‘The history began from AlexNet: A comprehensive survey on deep learning approaches,’’ in Proc. Comput. Vis. Pattern Recognit., 2018, pp. 1–39.

[13] F Ahmed, B Wang. “Identification of Crop Diseases and Insect Pests Based on Deep Learning”, Scientific Programming, Hindawi, 9179998,

https://doi.org/10.1155/2022/9179998, 2022.

[14] Tao, Y., & Cuicu, L. (2020). Recognition system for leaf diseases of Ophiopogon japonicus based on PCA-SVM. *Plant Diseases & Pests*, *11*(2).

[15] Ranjith, M., Bajya, D. R., Manoharan, T., & Ramya, R. S. (2018). Biodiversity of insect pests and natural enemies affiliated with wheat (Triticum aestivum) ecosystem in Haryana. *Indian Journal of Agricultural Sciences*, *88*(1), 157-158.

[16] Srinivasan, R., Paola, S., Lin, M. Y., Hy, H. C., Sareth, K., & Sor, S. (2019). Development and validation of an integrated pest management strategy for the control of major insect pests on yard-long bean in Cambodia. *Crop Protection*, *116*, 82-91.

[17] Toyinbo, J. O., Fatokun, C., Boukar, O., & Fakorede, M. A. B. (2021). Genetic variability and trait association under thrips (Megalurothrips sjostedti Trybom) infestation in cowpea (Vigna unguiculata [L.] Walp.). *Euphytica*, *217*(6), 1-9.