

FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

USING PYTHON

A Project report submitted in partial fulfilment of 7th semester in degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

TEAM ID: PNT2022TMID41050

ARUN KUMAR G: (612719104010)

ANAND K : (612719104004)

ABIMANYU R : (612719104002)

DINESH P : (612719104020)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
THE KAVERY ENGINEERING COLLEGE, MECHARI.**

THE KAVERY ENGINEERING COLLEGE, MECHARI.



BONAFIDE CERTIFICATE

Certificate that this project report “**FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION**” is the bonafide record work done by **MR ARUN KUMAR. G** (612719104010), **MR ANAND.K** (612719104004), **MR ABIMANYU.R** (612710104002), **MR DINESH.P** (612719104020) for **IBM-NALAIYATHIRAN** in VII semester of **B.E**, degree course in **computer science and engineering** branch during the academic year of 2022-2023.

Head Of The Department

MR M. BALAMURUGAN

Staff-in charge

Mrs G. SUDHA(Assi Prof)

Evaluator

Mrs G. SUDHA(Assi Prof)

Industry Mentor

Mrs DURGA PRASAD

Faculty Mentor

Mrs V.VIJAYAKUMARI(Assi Prof)

ACKNOELEDGEMENT

We are highly grateful to thank our Project Coordinator **MS SUDHA.G** and our Project Evaluator **MS SUDHA.G** Department Of Computer Science And Engineering, The Kavery Engineering College , Mechari for the Coordinating us throughout this Project .

We are very much indebted to thank all the faculty members of Department Of Computer Science And Engineering in institute, for their excellent moral support and suggestions to complete our Project Work Successfully.

ARUN KUMAR.G (612719104010)

ANAND.K (612719104004)

ABIMANYU.R (612719104002)

DINESH.P (612719104020)

ABSTRACT

Agriculture is a most important in human life and country development. Most of the plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and major threat to food security. Hence, early and accurate identification of plant diseases are essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods and inadequate plant protection techniques. An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

Farmers encounter great difficulties in detecting and controlling plant diseases. Thus, it is of great importance to diagnose the plant diseases at early stages so that appropriate and timely action can be taken by the farmers to avoid further losses. The project focuses on the approach based on image processing for detection of diseases of plants. In this paper, we propose an Android application that helps farmers for identifying plant disease by uploading a leaf image to the system. The system has a set of algorithms which can identify the type of disease. Input image given by the user undergoes several processing steps to detect the disease and results are returned back to the user via android application. In agricultural aspects, if the plant is affected by leaf disease, then it reduces the growth of the agricultural level. Finding the leaf disease is an important role of agriculture preservation. After pre-processing using a median filter, segmentation is done by Guided Active Contour method and finally, the

leaf disease is identified by using Support Vector Machine. The disease-based similarity measure is used for fertilizer recommendation.

Project Report Format

1.INTRODUCTION

- 1.1 Project overview
- 1.2 Purpose

2.LITERATURE SURVEY

- 2.1 Existing problem
- 2.2 Reference
- 2.3 Problem Statement Definition

3.IDEATION &PROPOSED SOLUTION

- 3.1 Empathy map Canvas
- 3.2 Ideation & Brainstroming
- 3.3 Probosed pollution
- 3.4 Problem solution fit

4.REQUIREMENT ANALYSIS

- 4.1 Functional Statement
- 4.2 Non Fuctional Requirements

5.PROJECT DESIGN

- 5.1 Data Flow Diagrams
- 5.2 Solution &Technical Architure
- 5.3 User stories

6.PROJECT PLANNING & SCHEDULING

- 6.1 Sprint Planning Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports From JIRA

7.CODING &SOLUTIONING(Explain the features added in the project along with code)

- 7.1 Feature 1

7.2 Feature 2

7.3 Database schema (if Applicable)

8.TESTING

8.1 Test cases

8.2 User Acceptence Testing

9.RESULTS

9.1 Performance Metrics

10.ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE WORK

13. APPENTIX

Source Code

Gitub&Project Demo Link

CHAPTER 1

INTRODUCTION

PROJECT OVERVIEW

Machine learning is particularly effective in identifying and detecting plant illnesses early on by providing symptoms of the diseases. For the purpose of diagnosing plant diseases, plant pathologists can analyse digital photos using digital image processing. Simply said, the use of computer vision and image processing techniques benefits farmers across all areas of agriculture. The yield of crops is impacted by variables such as the climate, soil, and illnesses. The agricultural business is seeking for new and better ways to enhance food production as a result of the growing population, political unrest, and changing climatic circumstances. Many farmers are leaving agriculture for new occupations as a result of industrialisation and low productivity. By utilising the modern technology, which enables farmers to boost agricultural output, productivity, and quality at a very low cost, we must expand the use of agriculture. We must keep an eye on and regulate conditions like temperature, humidity, and light in order to grow crops with improved yield and quality. In most cases, aberrant physiological functioning of plants is what causes plant diseases. Therefore, the difference between the plants' regular physiological capabilities and aberrant physiological functionalities leads to the generation of the specific symptoms. The pathogens that often infect plant leaves are found on the stems of the plants. Different image processing techniques can forecast these various leaf signs and illnesses. These many approaches make use of several core techniques including segmentation, feature extraction, and classification, among others. Most often, segmentation is used to distinguish between healthy and diseased tissues of leaves in order to forecast and diagnose leaf diseases.

PURPOSE

Different image processing techniques can forecast these various leaf signs and illnesses. These many approaches make use of several core techniques including segmentation, feature extraction, and classification, among others. Most often, segmentation is used to distinguish between healthy and diseased tissues of leaves in order to forecast and diagnose leaf diseases.



Figure 1.1 Plant leaf disease prediction

CHAPTER 2

LITERATURE SURVEY

EXISTING PROBLEM

- proposed a deep learning approach in which the complete system was learned without hand-engineered components. We pre-trained a convolutional neural network using 1.8 million images and used a fine-tuning strategy to transfer learned recognition capabilities from general domains to the specific challenge of Plant Identification task. The classification accuracy obtained by our method outperformed the best result was obtain.
- difficulties in detecting and controlling plant diseases. Thus, it is of great proposed the android application in agriculture to encounter great importance to diagnose the plant diseases at early stages so that appropriate and timely action can be taken by the farmers to avoid further losses. The project focuses on the approach based on image processing for detection of diseases of plants.

REFERENCES

- Angie K. Reyes, Juan C. Caicedo, and Jorge E. Camargo, “Fine-tuning Deep Convolutional Networks for Plant Recognition”, In CLEF (Working Notes), 2015.
- Mr. Ashish Nage, “Plant Leaf Diseases Detection Using Python V3.5.2”, International Engineering Journal for Research and Development, Vol. 4 No. 6, 2019.
- Surleen Kaur and Prabhpreet Kaur, “Plant Species Identification based on Plant Leaf Using Computer Vision and Machine Learning

Techniques”, Journal of Multimedia Information System, Vol. 6, No. 2, Page. No. 49-60, ISSN 2383-7632, 2019.

- Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino, “Deep-Plant: Plant Identification with Convolutional Neural Network”, Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Page. No. 452-456, 2015.
- Jana Waldchen, Michael Rzanny, Marco Seeland, and Patrick Mader, “Automated plant species identification-Trends and future directions”, PLoS Computational Biology, Vol. 14, No. 4, 2018.
- SrdjanSladojevic, Marko Arsenovic, Andras Anderla, DubravkoCulibrk, Darko Stefanovic, “Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification”, Computational Intelligence and Neuroscience, Vol. 6, Page. No. 1-11, 2016.
- N.Valliammal and Dr.S.N.Geethalakshmi, “Automatic Recognition System Using Preferential Image Segmentation For Leaf And Flower Images”, Computer Science & Engineering: An International Journal (CSEIJ), Vol.1, No.4, Page. No. 13-25, 2011.
- C. Arun Priya, T. Balasaravanan, Antony SelvadossThanamani, “An efficient leaf recognition algorithm for plant classification using support vector machine”, In Proceedings of International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012), Page. No. 428-432, 2012.
- Satnam Singh and Dr. Manjit Singh Bhamrah, “Leaf Identification Using Feature Extraction and Neural Network”, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), ISSN: 2278-2834, Volume 10, Issue 5, Ver. I, Page. No. 134-140, 2015.

- Vijayashree. T and Gopal. A, "Authentication of Leaf Image Using Image Processing Technique", ARPN Journal of Engineering and Applied Sciences, Vol. 10, No.9, Page. No. 4287-4291, 2015

PROBLEM STATE DEFINITION

The above literatures says that plant leaf disease prediction using image processing of MATLAB and neural network. Drawback of above topics only it has predicted the plant leaf disease but we have to predict the plant leaf disease and also recommended for particular disease.

CHAPTER 3

IDEATION & PROPOSED SOLUTION

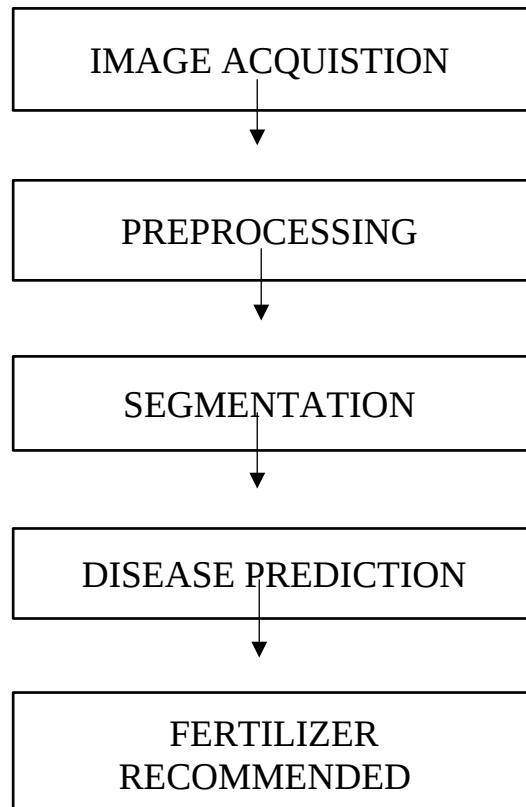
IDEATION & PLANT LEAF DISEASE

Convolutional Neural Networks are a class of feed forward neural networks that have the ability to process multidimensional data. In feed forward neural networks, perceptron's form layers. The first layer takes inputs and the last layer produces outputs. The external world is not connected with the middle layers called hidden layers. A perceptron of one layer is connected to every perceptron of the next layer but not the same layer. From one layer to another, information is fed forward constantly. CNN reduces images into an easy-to-process form, without compromising the quality of features that are required to get good prediction. Images are provided as input in order to classify the leaf diseases in a precise manner. From those images, the convolution layer (image filter) extracts the features. From the extracted features, the pooling layer calculates the feature values. The convolution and max pooling layer can further be added to obtain more information, depending on the complexity of images. Fully connected layer converts the output of previous layer into a single vector which is used as input for the next layer. In image classification, convolution and other layers extract feature maps repetitively and the network outputs a label indicating an estimated class eventually.

EMPATHY MAPS



PROPOSED SYSTEM DESIGN



PROPOSED SOLUTION FIT



CHAPTER 4

REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS

A digital camera or similar devices are used to take images of different types, and then those are used to identify the affected area in leaves. Then different types of image-processing techniques are applied to them, the process those images, to get different and useful features needed for the purpose of analysing later. Plant leaf disease identification is especially needed to predict both the quality and quantity of the. First segmentation step primarily based on a mild polygonal leaf model is first achieved and later used to guide the evolution of an energetic contour. Combining global shape descriptors given by the polygonal model with local curvature-based features, the leaves are then classified over leaf datasets. In this research work introduce a method designed to deal with the obstacles raised by such complex images, for simple and plant leaves. A first segmentation step based on graph-cut approach is first performed and later used to guide the evolution of leaf boundaries, and implement classification algorithm to classify the diseases and recommend the fertilizers to affected leaves.

NON-FUNCTIONAL REQUIREMENTS

Hardware Requirements

□ CPU type	:	Intel Pentium 4
□ Clock speed	:	3.0 GHz
□ Ram size	:	8 GB

- Hard disk capacity : 500 GB
- Monitor type : 15 Inch color monitor

Software Requirement

- Operating System : Windows 10, Linux
- Language : Python
- Tool : Anaconda

PYTHON

- Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.
- Python is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python:

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 is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner's Language – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Characteristics of Python

Following are important characteristics of Python Programming

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python Features

Python provides lots of features that are listed below.

Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high-level programming language.

Expressive Language

Python language is more expressive means that it is more understandable and readable.

Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

GUI Programming Support

Graphical user interfaces can be developed using Python.

Integrated

It can be easily integrated with languages like C, C++, and JAVA etc.

PYTHON APPLICATIONS

Python is known for its general-purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.

Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautiful Soup, Feed parser etc. It also provides Frameworks such as Django, Pyramid, Flask etc. to design and develop web-based applications.

Desktop GUI Applications

Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wx Widgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multi touch applications.

Software Development

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

Scientific and Numeric

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.

Business Applications

Python is used to build Business applications like ERP and e-commerce systems.

Console Based Application

We can use Python to develop console based applications. For example: IPython.

Audio or Video based Applications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

3D CAD Applications

To create CAD application Fandango is a real application which provides full features of CAD.

Enterprise Applications

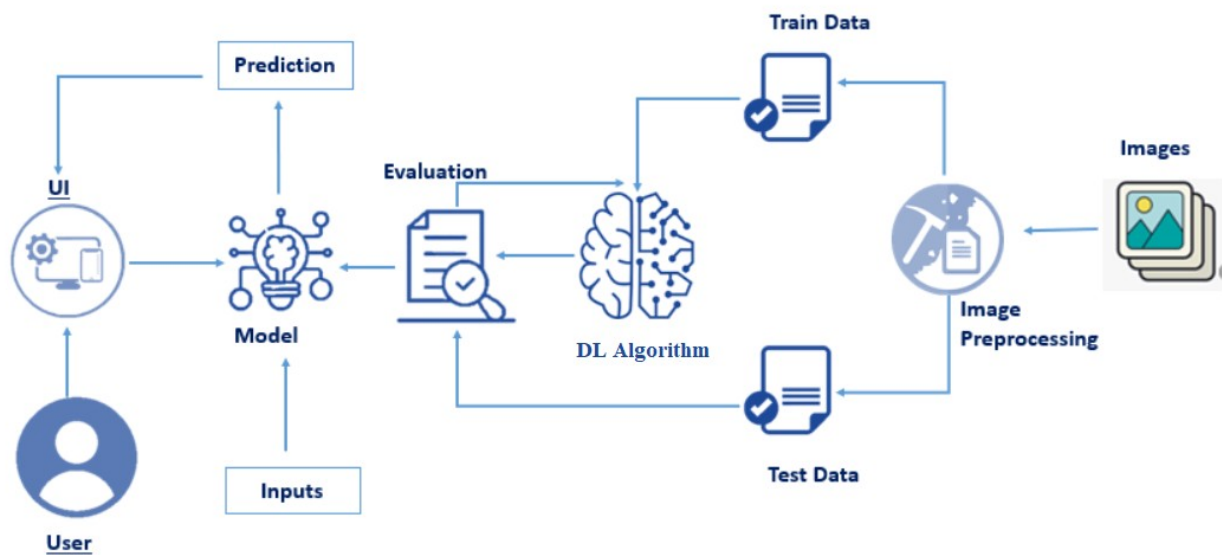
Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

Applications for Images

Using Python several applications can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc.

PROJECT DESIGN

Data flow diagrams



USER STORES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer (Mobile user)	Download the database	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	I can access my account / dashboard	High
	Register	USN-2	As a user, I can register for the application by entering my email, password, and confirming my password	I can receive confirmation email & click confirm	High
	Login	USN-3	As a user, I will receive confirmation email once I have registered for the application	I can register & access the dashboard with Facebook Login	Low
	Upload the image	USN-4	As a user, I must upload the image to identify the disease		Medium
Customer (Web user)	The functional requirements are same as mobile user	Same as mobile user	Same as mobile user	Same as mobile user	High when compared to mobile users

PROJECT PLANNING & SCHEDULING

Sprint planning & Estimation

Milestone

In modern Technology are increassing and optomizing the performance of the Artificial intelligence (AI) model.

In based crob yield Disease prediction system, it will be helpful for formers to prevent the crop from the various Disease which can be identify the Disease with in a process of capturing the image at the plant and machine Learing Algorithm will give affected Disease Name.

In this project Milestone will be given the Best solution for the former using complete friendly and simple user interface web application to fetching the solution by own.

In addition process we are planned to add a valid module that is fertilizer recommendation for the specific Disease ,it can give both artificial and Natural Fertilizerin suggestion manager.



Activity List

In project Management Planning is an important task to scheduling the phrase of the project to the Team Members.

In this Activity can shows the various Activity are allocated and done by the Team Members.

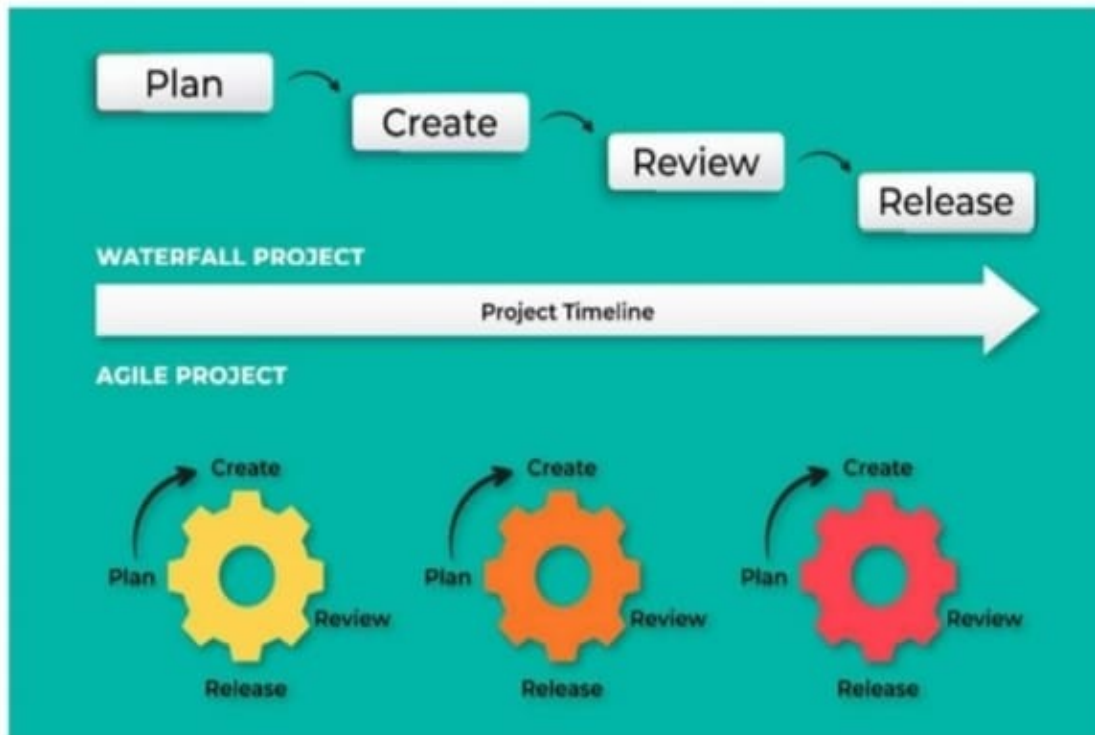
In project we can Split into the Four Step of Phrases are,

Phase 1: Informatin Collection and Requirement Analysis.

Phase 2: Project Planning and Developing Modules.

Phase 3: Implementing the high Accuracy Deep Learning Algorithm to Perform.

Phase 4: Deploying the model on Cloud and Testing the Model and UI Performance.



Agile Methodology for Activity Planning

Sprint Delivery Schedule

The delivery plan of project deliverables is a strategic element for every Project Manager. The goal of every project is, in fact, to produce a result that serves a specific purpose. With the word „purpose“, we can mean the most disparate goals: a software program, a chair, a building, a translation, etc....

In Project Spirit Delivery Planning is one of the processes of Completing the project and Show Casing the Time Line of the Project Planning. This Delivery plan help to understanding the process and Work Flow of the Project working by the Team Mates.

Every Single Modules are assigned to the team mates to show case their work and contribution of developing the Project

```
graph TD; A[Spirit Model Planning] --> B[Spirit One]; A --> C[Spirit Two]; A --> D[Spirit Three]; A --> E[Spirit Four];
```

Spirit One	Spirit Two	Spirit Three	Spirit Four
<ul style="list-style-type: none">✓ The team should conduct a proof of concept.	<ul style="list-style-type: none">✓ The team make any final adjustments to the process.	<ul style="list-style-type: none">✓ The team mates should understand the project Flow	<ul style="list-style-type: none">✓ The team will show the Demo process to team mentor
<ul style="list-style-type: none">✓ The team should start by outlining the milestones.	<ul style="list-style-type: none">✓ The team should make the prototype for their project	<ul style="list-style-type: none">✓ The team should monitor the process of work	<ul style="list-style-type: none">✓ The team mates will updating the project modules in cloud
<ul style="list-style-type: none">✓ The team should monitor the process efficiency.	<ul style="list-style-type: none">✓ The team should start by analysis the project Specification	<ul style="list-style-type: none">✓ The team should follow the project Management System	<ul style="list-style-type: none">✓ The team make to deploy the final Model for the Testing Process



TESTING MODELS

Unit Testing

Unit testing focuses on verification errors on the smallest unit of software design-the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

The module interface is tested to ensure that the information properly flows into and out of the program unit under test. Boundaries conditions are tested to ensure that the module operates properly at the boundaries established to limit of restrict processing.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associated with interfacing. The

objective is to take unit tested modules and build a program structure that have been dictated by design.

White Box Testing

White box testing is some time is called glass box testing, is a test case design that uses a control structure of the procedural design to drive the test cases. Using white-box testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions are on the true and false sides
- Exercise all logical decisions are on the true and false sides
- Execute all loops at their boundaries and within their operational bounds
- Exercise internal data structure to assure the validity

Acceptance Testing

Finally when the software is completely built, a series of acceptance tests are conducted to enable the client to validate all requirements. The user conducts these tests rather than the system developer, which can range from informal test drive to a planned and systematical executed series of tests. These acceptance tests are conducted over a period of weeks or months, there by uncovering cumulative errors that might degrade the system order time. In this process alpha testing and beta testing are used to uncover the errors that only the end user seems able to find.

Alpha Testing

The customer conducts the alpha test at the developer's site. The client notes the errors and usage problems and gives report to the developer. Alpha tests are conducted in a control environment.

Beta Testing

The beta testing is conducted at one or more customer's sites by the end users of the software. Unlike the alpha testing, the developer is not present. Therefore a beta test is a "live" application of the software in the environment that cannot be developed by the developer. The customer records all the problems encountered during the beta testing and reports these to the developers at regular intervals.

Black Box Testing

Black box testing focuses on the functional requirements of the software. That is black box testing enables the software engineer to drive a set of input conditions that will fully exercise the requirements for a program.

Black box testing is not an alternative for white box testing techniques. Rather, it is a complementary approach that is likely to uncover different class of errors. Black box testing attempts to find errors in the following categories:

- Interface errors.
- Performances in data structures or external database access.
- Performance errors.
- Initialization and termination errors.
- Incorrect or missing functions.

All the above-mentioned errors were checked in the process of black box testing and the bugs Found Were Fixed.

TESTING

Test Cases

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to your customer. Your goal is to design a series of test cases that have a high likelihood of finding errors. To do so we have techniques provide systematic guidance for designing tests that: (1) exercise the internal logic of software components, and (2) exercise the input and output domains of the program to uncover errors in program function, behavior, and performance. Resource presented in this section address the following topic categories.

Software Testing is the process of confirming the functionality and correctness of software by running it. Software testing is usually performed for one of two reasons:

- Defect detection
- Reliability estimation.

The problem of applying software testing to defect detection is that software can only suggest the presence of flaws, not their absence (unless the testing is exhaustive). The problem of applying software testing to reliability estimation is that the input distribution used for selecting test cases may be flawed. In both of these cases, the mechanism used to determine whether program output is correct is often impossible to develop. Obviously the benefit of the entire software testing process is highly dependent on many different pieces. If any of these parts is faulty, the entire process is compromised.

Software is now unique unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical systems fail in a fixed (and reasonably small) set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible.

The key to software testing is trying to find the myriad of failure modes – something that requires exhaustively testing the code on all possible inputs. For most programs, this is computationally infeasible. It is commonplace to attempt to test as many of the syntactic features of the code as possible (within some set of resource constraints) are called *white box* software testing technique. Techniques that do not consider the code's structure when test cases are selected are called black box technique.

Functional testing is a testing process that is black box in nature. It is aimed at examine the overall functionality of the product. It usually includes testing of all the interfaces and should therefore involve the clients in the process. Final stage of the testing process should be System Testing. This type of test involves examination of the whole computer system, all the software components, all the hard ware components and any interfaces. The whole computer-based system is checked not only for validity but also to meet the objectives.

System Implementation

Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system or it may be major modification to the system currently put into use. This system “Access Point Selection for Improving the Voice Quality and Overall Throughput in Wireless LANs” is a new system. Implementation as a whole involves all those tasks that we do for successfully replacing the existing or introduce new software to satisfy the requirement. The test case has performed in all aspect and the system has given correct result in all the cases.

The System implementation phase consists of the following steps:

- Testing the developed software with sample data.
- Correction of any errors if identified.
- Creating the files of the system with actual data.

- Making necessary changes to the system to find out errors.
- Training of user personnel.

The system has been tested with sample data, changes are made to the user requirements and run in parallel with the existing system to find out the discrepancies. The user has also been appraised how to run the system during the training period.

This phase is primarily concerned with user training, site preparation and file conversions. During the final testing, user acceptance is tested, followed by user training. Depending in the nature of the extensive user training may be required.

After development and testing has been completed, implementation of the information system can begin. During system implementation, the project team should be brought back to full strength. During software development stage, project teams tend to play passive role as the technical steps of program development and testing evolve. However, broad organizational representation, accomplished through the project team, is required to complete the system development cycle.NET Framework has offer very efficient yet simple implementation techniques for development of the project.

Implementation plan

Implementation is the stage, which is crucial in the life cycle of the new system designed. Implementation means converting a new or revised system design into an operational one. This is the stage of the project where the theoretical design is turned into a working system. In this project “**Access Point Selection**” implementation includes all those activities that take place to convert from the old system to the new one. The important phase of implementation plan is change over.

The implementation phase’s construction, installation and operations lie on the new system. The most crucial and very important stage in achieving a new

successful system and in giving confidence on the new system for the user that it will work efficiently and effectively.

There are several activities involved while implementing a project:

- Careful planning
- Investigation current system and its constraints on implementation
- Design of methods to achieve the change over
- Training of the staff in the changeover procedure and evaluation of change over method

The implementation is the final stage and it is an important phase. It involves the individual programming system testing, user training and the operational running of developed proposed system that constitutes the application subsystems. On major task of preparing for implementation is education of users, which would really have taken place much earlier in the project when we're being involved in the investigation and design work. The implementation phase of software development is concerned with translating design specifications into source code. The user tests the developed system and changes are made according to their needs.

Change over

The implementation is to be done step by step since testing with dummy data will not always reveal the faults. The system will be subjected to the employees to work. If such error or failure is found, the system can be corrected before it is implemented in full stretch.

The trail should be done as long as the system is made sure to function without any failure or errors. Precautions should be taken so that any error if occurred should not totally make the process to a halt. Such a care should be taken. The system can be fully established if it does not create any error during the testing period.

Education and user training

Well-designed and technically elegant systems can succeed or fail because of the way they are operated and used. Therefore the quality of the training received by the personnel involved with the systems help or hinder, and may even prevent, the successful completion of the system.

An analysis of user training focuses on user capabilities and the nature of the system being installed. Those users are verifying type and nature. Some of them may not have any knowledge about the computers and the others may be very intelligent. The requirements of the system also range from simple to complex tasks. So the training has to be generated to the specific user based on his/her capabilities and system's complexity.

User training must instruct individuals in trouble shooting the system, determining whether a problem that arises is caused by hardware or software. A good or perfect documentation which instructs the user on how to start the system and the various functions and meanings of various codes must be prepared and that will help the user to understand the system in a better manner. Through the training demonstration with personnel contact also, the user can be trained. This training demonstration will help the users to understand the system in many ways.

By this the user receives encouragement and attention. Another rapid way of training the user is by resident experts. Several user training aids are provided such as user manual.

CODING AND SIMULATION

#Import necessary libraries

```
from flask import Flask, render_template, request

import numpy as np

import os

from tensorflow.keras.preprocessing.image import load_img

from tensorflow.keras.preprocessing.image import img_to_array

from tensorflow.keras.models import load_model

filepath =
'C:/Users/Madhuri/AppData/Local/Programs/Python/Python38/Tomato_Leaf_D
isease_Prediction/model.h5'

model = load_model(filepath)

print(model)

print("Model Loaded Successfully")

def pred_tomato_dieas(tomato_plant):

test_image = load_img(tomato_plant, target_size = (128, 128)) # load image

print("@@ Got Image for prediction")

test_image = img_to_array(test_image)/255 # convert image to np array and
normalize

test_image = np.expand_dims(test_image, axis = 0) # change dimention 3D to
4D

result = model.predict(test_image) # predict diseased plant or not
```

```
print('@@ Raw result = ', result)

pred = np.argmax(result, axis=1)

print(pred)

if pred==0:

    return "Tomato - Bacteria Spot Disease", 'Tomato-Bacteria Spot.html'

elif pred==1:

    return "Tomato - Early Blight Disease", 'Tomato-Early_Blight.html'

elif pred==2:

    return "Tomato - Healthy and Fresh", 'Tomato-Healthy.html'

elif pred==3:

    return "Tomato - Late Blight Disease", 'Tomato - Late_blight.html'

elif pred==4:

    return "Tomato - Leaf Mold Disease", 'Tomato - Leaf_Mold.html'

elif pred==5:

    return "Tomato - Septoria Leaf Spot Disease", 'Tomato - Septoria_leaf_spot.html'

elif pred==6:

    return "Tomato - Target Spot Disease", 'Tomato - Target_Spot.html'

elif pred==7:

    return "Tomato - Tomoato Yellow Leaf Curl Virus Disease", 'Tomato - Tomato_Yellow_Leaf_Curl_Virus.html'

elif pred==8:
```

```
        return "Tomato - Tomato Mosaic Virus Disease", 'Tomato -  
Tomato_mosaic_virus.html'
```

```
elif pred==9:
```

```
        return "Tomato - Two Spotted Spider Mite Disease", 'Tomato - Two-  
spotted_spider_mite.html'
```

```
# Create flask instance
```

```
app = Flask(__name__)
```

```
# render index.html page
```

```
@app.route("/", methods=['GET', 'POST'])
```

```
def home():
```

```
    return render_template('index.html')
```

```
# get input image from client then predict class and render respective .html page  
for solution
```

```
@app.route("/predict", methods = ['GET','POST'])
```

```
def predict():
```

```
    if request.method == 'POST':
```

```
        file = request.files['image'] # fet input
```

```
        filename = file.filename
```

```
print("@@ Input posted = ", filename)
```

```
file_path=
```

```
os.path.join('C:/Users/Arunkumar/AppData/Local/Programs/Python/Python38/  
Tomato_Leaf_Disease_Prediction/static/upload/', filename)
```

```
file.save(file_path)

print("@@ Predicting class.....")

    pred, output_page = pred_tomato_dieas(tomato_plant=file_path)

    return render_template(output_page, pred_output = pred, user_image =
file_path)

# For local system & cloud

if __name__ == "__main__":

app.run(threaded=False,port=8080)
```


RESULT AND DISCUSSION

To compare the performance of the proposed SVM method with the existing CNN (Convolutional Neural Network) method. Metrics such as True Positive, False Positive, True Negative, False Negative are used. The proposed method is implemented using .NET. The code existing CNN method was written in Python was downloaded from the web [<https://github.com/cs-chan/Deep-Plant>].





ADVANTAGES & DISADVANTAGES

List of advantage

- The proposed model here produces very high accuracy of classification.
- Very large Datasets can also be trained and tested.
- Images of very can be resized within the proposed itself.

List of Disadvantages

- For training and testing, the proposed model requires very high computational time.
- The neural network architecture used in this project work has high complexity.

CONCLUSION

The major objective is to effectively identify and diagnose leaf diseases while keeping in mind the advantages for farmers. A model that resembles the human brain is created with the use of neural networks. There were relatively few models that were really trained in this manner prior to the invention of neural networks. The problem of pricey domain experts is resolved by the suggested solution. Following accurate diagnosis of the leaf illness, it also offers the treatment that must be used as a cure to restore the health of the plant in a short amount of time. This model may be mounted on drones to provide aerial surveillance and real-time coverage of expansive agricultural fields, which will cut down on the amount of time and human labour needed. The drone's high-resolution camera takes pictures of the leaves that the model uses as input. The budget is relatively high for small-scale farming, but it is valuable

for large-scale farming. The plant leaf prediction and fertilizer recommended system implements the real time application in agriculture field.

FUTURE WORK

It is possible to create a smartphone application that uses voice commands for navigation so that farmers with little literacy may use it with ease. The model may be expanded by also integrating the diagnostic for an increased number of leaf diseases, which will be advantageous. Extensions can also be implemented, such as displaying the portion of the leaf that is impacted in terms of percentage.

APPLINDIX

Github link - <https://github.com/IBM-EPBL/IBM-Project-44983-1660727689>

Demo video link

https://drive.google.com/file/d/1DMfRnQN-antRcs5fBLQi_SohGMKybhOf/view