KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOY

TEAM ID: PNT2022TMID13660

NALAIYA THIRAN PROJECT REPORT 2022

SUBMITTED BY:

KEERTHIKA S (621319205014)

SUGUNADEVI M (621319205048)

SURUTHIKA S (621319205049)

VINITHA S (621319205057)

TEAM MENTOR:

Mr. J. SATHISHKUMAR, M.Tech, (Ph. D)

CONTENTS

1. INTRODUCTION

Machine learning, as the label intimate, is the learning of machines automatically without being explicitly maneuvered, or learning without direct human intervention. This machine learning process starts by feeding high-quality data and then trains a machine by using the data and different algorithms to create different machine learning models. Choosing an algorithm rely on what kind of data you have and what kind of task you're trying to automate.

As regards the formal characterize of machine learning, a machine learning algorithm is defined by experience E can be said to learn from for example, when playing chess using machine conclusion . Then the experience E is playing many chess games, the task T is playing chess with many players, and the performance measure P is the probability that the algorithm wins the chess game.

Machine Learning is a branch of the broader field of artificial intelligence that makes use of statistical models to develop predictions. It is often described as a form of predictive modelling or predictive analytics and traditionally, has been defined as the ability of a computer to learn without explicitly being programmed to do so. In basic technical terms, machine learning uses algorithms that take empirical or historical data in, analyse it, and generate outputs based on that analysis. In some approaches, the algorithms work with so-called “training data” first and then they learn, predict, and find ways to improve their performance over time.

1.1 Project Overview

Growing plants and maintaining them are becoming a challenging task to farmers now a days due to pathogens and climatic conditions that affect both quality and production of crops. The plant illness is caused due to its abnormal anatomical functionalities of the plant tissues. Leaves of the crop can be affected by fungus, bacteria, viruses and other insects. Machine learning techniques are used to detect the disease and providing fertilizers to the crops. The disease occurs mostly on the leaf of the crop. Capturing the plant leaf and fed them into suitable classification algorithms greatly helps to identify the disease and fertilizer recommendation. The algorithm includes segmentation, classification, feature extraction. Mainly the tissues of the leaf get affected. Healthy tissue and the diseased tissue are classified and suitable fertilizer with proper guidance will be suggested to the customers.

1.1 Purpose

Fertilizers are materials given to plants to increase their productivity. Farmers use them daily to increase their relent. These fertilizers contain essential nutrients that plants need, such as nitrogen, potassium and phosphorus. They are organic or inorganic. Improves plant metabolism. Fertilizers are added to crops to (economically) encounter yield and quality goals in a way that diminishes nutrient misplacing to the environment. Proper guidance in recommending fertilizers to users increases crop productivity and reduces hunger.

2. LITERATURE SURVEY

Authors: Shriya Sahu, Meenu Chawla and Nilay Khare

Title: An Efficient Analysis of Crop Yield Prediction using Hadoop Framework Based on Random Forest Approach

Published in: IEEE (2019)

Methodology:

This paper considers various parameters from soil to atmosphere to predict suitable crops. Soil parameters such as type, pH value, iron, copper, manganese, sulfur, organic carbon, potassium, phosphate and nitrogen are considered. A random forest algorithm is used to classify the dataset, resulting in low error rate and high accuracy. This framework can process large datasets by working with the MapReduce programming model. The proposed work phases are Data Collection, Data Classification (Random Forest Algorithm), Hadoop Framework - MapReduce Programming Model, and Final Prediction. The implementation is on Ubuntu 14.04 LTS with Hadoop 2.6.0 to collect datasets from various online sources to predict good crops. Accuracy: The accuracy achieved with this method is 91.43%. Future work: Future work will be devoted to predicting the required fertilizer and pesticide ratios based on the atmospheric and soil parameters of the cultivated land.

DISADVANATGE:

- It slows down the study process.

Authors: Luca Bencini, Davide Di Palma, Giovanni Collodi, G. Manes and Antonio Manes

Title: Agricultural monitoring based on wireless sensor network technology: Real long life deployments for physiology and pathogens control.

Published on: Sensor Technologies and Applications. IEEE, 2017.

Methodology:

The proposed system is related to increase in net crop yield based on soil and air related parameters. This model provides revenue predictions that can be made using a "Bayesian Algorithm". Data mining is used to extract large amounts of data from datasets and analyze that data to predict and suggest returns. Limitations include the inaccuracy of atmospheric forecasts. Clustering and SGDM are the most commonly used approaches to plant disease identification. Challenges with these approaches include the influence of background data on the final image, optimization of specific crop foliar disease methodologies, and continuous automated monitoring techniques for crop foliar disease under real field conditions. automation etc. Reviews show that this disease detection technology is promising but has some drawbacks such as its ability to identify plant leaf diseases. As a result, existing research leaves room for improvement.

DISADVANATGE:

- Proposed automatic system that ensures the minimum loss function and maximum prediction accuracy.

Authors: Ms. Kiran R. Gavhale, Ujwalla Gawande

Title: Plant Leaves Disease detection using Image Processing Techniques**.**

Year of Publication: 2020

Methodology**:**

This study explores and describes image processing strategies for plant disease identification in a large number of plant species. BPNN, SVM, K-means. and classify. Color information is widely used in the study of plant leaf diseases. The filters in this model are applied to the three channels according to their RGB components. To train the network, the LVQ was fed the output feature vectors of the convolutional components. Experimental results show that the proposed approach accurately detects its four forms of tomato leaf diseases. Since the proposed approach was not applied in the field of plant disease detection, no comparison was made with similar results obtained by precise methods. This research was extended to develop a complete system consisting of server-side components such as a pre-trained model and a smart mobile device application that can identify diseases photographed by a smartphone camera. The authors anticipate that the extension of this study will have a significant impact on sustainable development and affect the quality of crops for future generations.

DISADVANATGE:

- For automatic hate speech detection, we used multi-faceted text representations.

Authors: S. Yegneshwar Yadhav, T. Senthilkumar, S. Jayanthy, J. Judeson Antony Kovilpillai

Title: Plant Disease Detection and Classification using CNN Model with Optimized Activation Function

Year of publication: 2020

Methodology:

In this study, a Convolutional Neural Network (CNN) algorithm is used for optimal real-time detection of crop diseases and affected areas so that appropriate fertilizers can be applied to prevent further damage to crops by pathogenic viruses. I will explain the use of activation functions, which are at the heart of CNN models as they combine non-linearities to create real artificial intelligence systems for classification. ReLu is one of the best activation functions, but it has the problem that its derivative vanishes to negative values, leading to nerve necrosis. To improve the accuracy and performance of the system, TensorFlow uses two fully connected dense layers and a sigmoid function. The model was trained on his dataset of 3663 images of apple and tomato leaves and provided an accuracy of 87%. Overlap issues are detected and fixed when the dropout value is set to 0.2. The model also runs on a Tesla GPU and is capable of parallel processing, so we evaluate its speed and accuracy.

DISADVANATGE:

- To identify healthy leaf as diseased a list of criteria based on critical race theory has been presented.

Authors: Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic.

Title: Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification

Year of Publication: 2017

Methodology:

In this study, we developed a novel method for automatic plant disease classification and detection from leaf images using deep learning algorithms. The developed computer was able to detect the presence of leaves and differentiate between healthy leaves and 13 visually detectable abnormalities. The overall accuracy of the trained model was finally 96.3%. Since the proposed approach was not applied in the field of plant disease detection, no comparison was made with similar results obtained by precise methods. This research was extended to develop a complete system consisting of server-side components such as a pre-trained model and an application for smart his mobile device that can identify diseases photographed by a smartphone camera. I'm here. The authors anticipate that the extension of this study will have a significant impact on sustainable development and affect the quality of crops for future generations.

DISADVANATGE:

- Should seek out hateful sources of training data without resorting

Authors: Melike Sardogan, Adem Tuncer, Yunus OzenPlant

Title: Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm

Year of Publication: 2018

Methodology:

Early detection of diseases is very important to achieve effective yields in agriculture. Bacterial leaf spot, downy mildew, septoria leaf spot and yellow curl affect tomato quality. Automated methods for classifying plant diseases can also help take action after symptoms of foliar diseases are detected. This paper describes a convolutional neural network (CNN) model and a method based on the learning vector quantization (LVQ) algorithm for tomato leaf disease detection and classification. The dataset contains 500 images of tomato leaves showing symptoms of four diseases. We created a CNN model for an automated feature extraction framework. A new mathematical activation function is constructed and compared with the current activation function. Experimental results on the trained database show that the generated activation function improved the accuracy and performance of the CNN model by 95%. Implementing the proposed optimizer on an ARM processor increases the training speed of the CNN model by 83%. Additional disease affected area is calculated using a K-means clustering approach to fertilizer optimization.

DISADVANATGE:

- Poor accuracy due to less dataset.

Authors: Ishrat Zahan Mukti

Title: Dipayan BiswasTransfer Learning Based Plant Diseases Detection Using ResNet50

Year of Publication: 22th December 2019

Methodology:

Plant diseases are a major threat to food safety. The biggest challenge in agriculture is the detection of plant diseases. State-of-the-art convolutional neural networks (CNNs) provide excellent results in computer vision image classification tasks. Transfer learning allows you to build deep CNNs in the most cost-effective way. In this work, we developed a transfer learning-based CNN model for accurate plant disease identification. The dataset we used consisted of 70,295 training images and 17,572 validation images containing plant leaf images of 38 different classes. We primarily focused on ResNet50 networks, a popular CNN architecture, as pre-trained transfer learning models. In addition, several transfer learning architectures were tested on other popular pre-trained models (VGG16, VGG19, AlexNet) and compared with the proposed model. The proposed model performed best with 99.80% training accuracy.

DISADVANATGE:

- The Recommended fertilizer is suitable to use for specific crop.

Authors: Duan Yan-e

Title: Design of Intelligent Agriculture Management Information System Based on IOT, IEEE,4th, Fourth International reference on Intelligent Computation Technology and Automation

Year of Publication: 2011

Methodology:

In his article, the author suggests ways to help predict yields by suggesting the best crops. It also looks at soil type to determine which crops should be planted in the field to increase productivity. Soil type is important when it comes to yield. By including the previous year's weather data in the formula, you can get information about the surface. It can predict which plants are suitable for a particular climate. Harvest quality can also be improved using weather and disease related datasets. Predictive algorithms help classify data based on disease and data extracted from classifiers are used to predict soil and crops. Due to changing climatic conditions, this system cannot predict exact results.

DISADVANATGE:

- Due to changing climatic conditions, the accurate results cannot be predicted by the system.

Authors: Jay GholapAnurag Ingole, Jayesh Gohil, Shailesh Gargade and Vahida Attar,

Title: Soil Data Analysis Using Classification Techniques and Soil Attribute Prediction

Year of Publication: 2012.

Methodology:

The proposed system deals with increasing the net yield of crops based on soil and air related parameters. This model provides revenue predictions that can be made using a "Bayesian Algorithm". Data mining is used to extract large amounts of data from datasets and analyse that data to predict and suggest returns. Limitations include the inaccuracy of atmospheric forecasts. Clustering and SGDM are the most commonly used approaches to plant disease identification. Challenges with these approaches include the influence of background data on the final image, optimization of specific crop foliar disease methodologies, and continuous automated monitoring techniques for crop foliar disease under real field conditions. automation etc. Reviews show that this disease detection technology is promising but has some drawbacks such as its ability to identify plant leaf diseases. As a result, existing research leaves room for improvement.

DISADVANATGE:

- Drawbacks including the capacity to identify plant leaf illnesses. As a result, existing research has room for improvement.

Authors: Dubravko Culibrk, and Darko Stefanovic

Title: AI to detect Plant Diseases by Leaf Image Classification.

Year of Publication: 2016

Methodology:

In this study, we developed a narrative method for automatic plant disease classification and detection from leaf images using deep learning algorithms. The developed computer was able to detect the presence of leaves and differentiate between healthy leaves and 13 visually detectable abnormalities. The overall accuracy of the trained model was finally 96.3%. Since the proposed approach was not applied in the field of plant disease detection, no comparison was made with similar results obtained by precise methods. This research was extended to develop a complete system consisting of server-side components, including a pre-trained model and an application for smart-his mobile-his devices that can identify diseases photographed by a smartphone camera. The authors anticipate that the extension of this study will have a significant impact on sustainable development and affect the quality of crops for future generations.

DISADVANATGE:

- Due to the changing climatic conditions, accurate results cannot be predicted by this system.

2.1 Existing Problem

Existing systems use cameras to take picture of plants and use images to identify affected areas of leaves. Various kinds of image processing methods are then applied to these images and processed to obtain various useful features required for plant leaf disease analysis. Plant disease detection and machine learning detection are highly methodical for providing disease detection symptoms as early as possible. Plant pathologists can use digital image processing to analyse digital images and interpretation of plant diseases. By applying computer vision and the image processing strategies, we can easily help farmers in all agricultural regions. Plant diseases are generally caused by tissue abnormalities in plant physiology. Thus, characteristic symptoms are generated based on the distinction between normal and abnormal plant physiology. In most cases, plant leaf diseases are caused by infectious agents that reside in the plant stem. These various leaf manifestations and leaf illness are predicted by different image processing methods. These different methods involve different basic processes such as segmentation, feature extraction, and classification.

2.2 References

[1] Reyes Angie. K, Juan C. Caicedo, and Jorge E. Camargo, “Fine-tuning Deep Convolutional Networks for Plant Recognition”, In CLEF (Working Notes), 2019.

[2] Dimitrovski, Ivica, GjorgjiMadjarov, DragiKocev, and PetreLameski, “Maestra at LifeCLEF 2014 Plant Task: Plant Identification using Visual Data”, In CLEF (Working Notes), pp. 705-714, 2021.

[3] Uzal, Lucas C., Guillermo L. Grinblat, Rafael Namías, Mónica G. Larese, J. S. Bianchi, E. N. Morandi, and Pablo M. Granitto. “Seed-per-pod estimation for plant breeding using deep learning.” Computers and electronics in agriculture 150 (2018): 196-204.

[4] Lee, Sue Han, CheeSeng Chan, Paul Wilkin, and Paolo Remagnino. “Deep-plant: Plant identification with convolutional neural networks.” In 2015 IEEE International Conference on Image Processing (ICIP), pp. 452-456.IEEE, 2015.

[5] Lee, Sue Han, Chee Seng Chan, Simon Joseph Mayo, and Paolo Remagnino. “How deep learning extracts and learns leaf features for plant classification.” Pattern Recognition 71 (2017)

[6] Pujari, Devashish, Rajesh Yakkundimath, and Abdulmunaf S. Byadgi. “SVM and ANN based classification of plant diseases using feature reduction technique ,2016”

[7] Akbarzadeh, Saman, AriePaap, SelamAhderom, BeniaminApopei, and Kamal Alameh. “Plant discrimination by Support Vector Machine classifier based on spectral reflectance, 2018”

[8] Ramesh, S., and D. Vydeki. “Application of machine learning in the detection of blast disease in South Indian rice crops.” Journal of Phytology (2019).

[9] Mohanty, Sharada P., David P. Hughes, and Marcel Salathé, “Using deep learning for image-based plant disease detection”, Frontiers in plant science 7, pp: 1419, 2016.

[10] Lee, Sue Han, CheeSeng Chan, Simon Joseph Mayo, and Paolo Remagnino, “How deep learning extracts and learns leaf features for plant classification”, Pattern Recognition 71, pp: 1-13, 2017

2.3 Problem statement definition

A fertilizer recommendation system is a system that recommends fertilizers based on plant diseases. Fertilizers are added to crops to achieve yield and quality targets in a manner that minimizes nutrient loss to the plants. Introduction of fertilizers for profuse plant diseases, enhancement of fertilizer proposals through appropriate guidance. Fertilizers replace nutrients that plants remove from the soil. Without additional fertilizer substances, yields and agricultural productivity are greatly reduced. Fertilizers are therefore important today for better yields. Customers are happy when the right fertilizer is recommended with the right guidance. It is a rapid feed system that recommends fertilizer. Connect farmers and retailers with a digital signal. This system can generate more income than the investment in a few years. This system enables drought-tolerant agriculture. Recommendation systems are used in the agricultural sector to provide customers with efficient recommendations for choosing suitable fertilizers for their crops so that they produce more and have less impact on the environment.

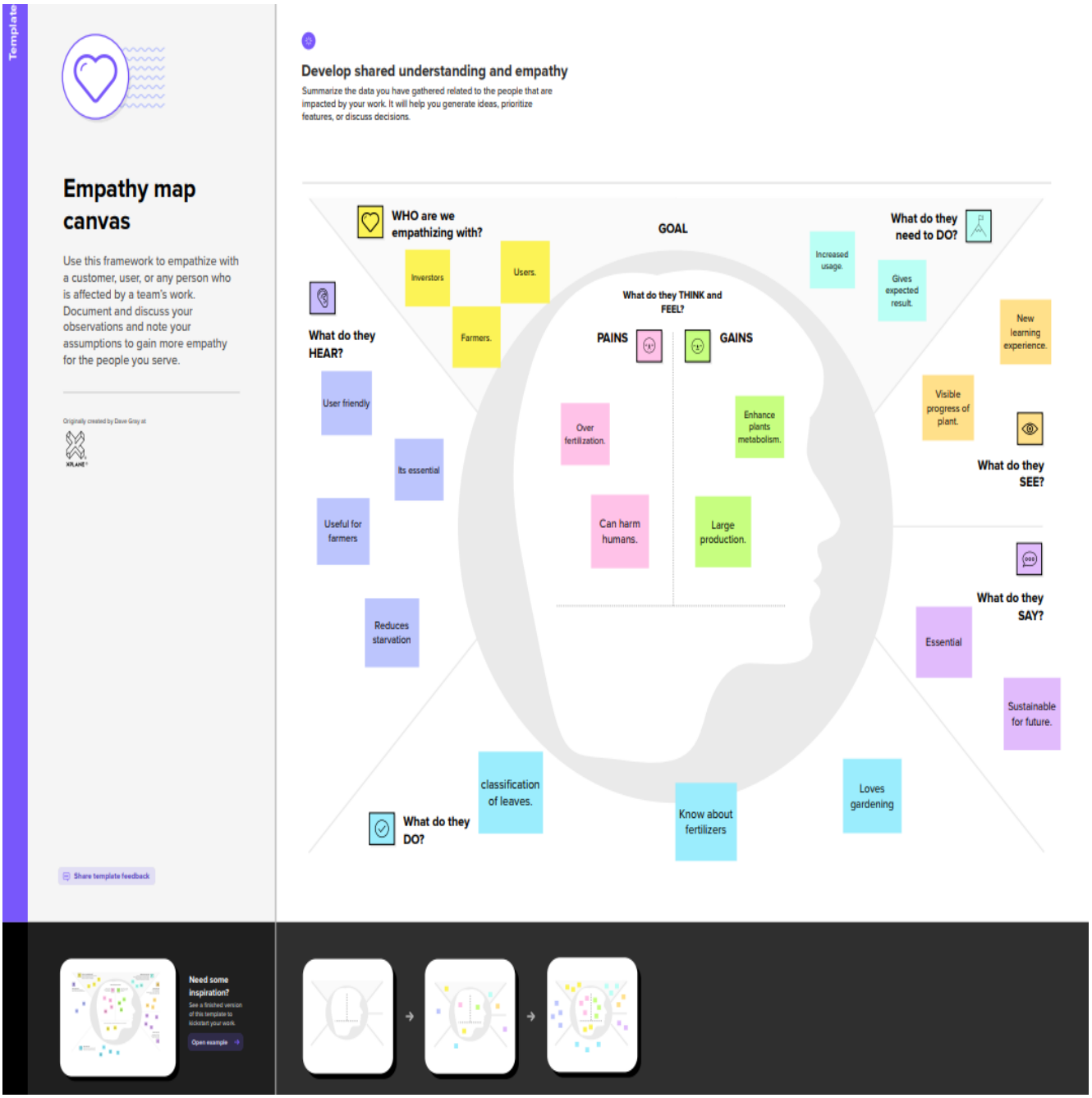
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Empathy Maps is a collaboration tool that enables teams to gain deeper insight into their customers. Similar to user personas, empathy maps can represent user groups such as customer segments. Created by Dave Gray, his Empathy Map has become widely known in the Agile community. Empathy maps can be used whenever you need to delve into a user's environment.

Every people would add at least one sticky to every section. The user can might ask questions, such as:

- What would the user be thinking and/or feeling? What are some of their worries and aspirations?
- What would their friends, colleagues, and boss be likely to say while the user is using our product? What would the user hear in these scenarios?
- What would the user see while using our product in their environment?
- What might the user be saying and/or doing while using our product? How would that change in a public or private setting?
- What are some of the user’s pain points or fears when using our product?
- What gains might the user experience when using our product?



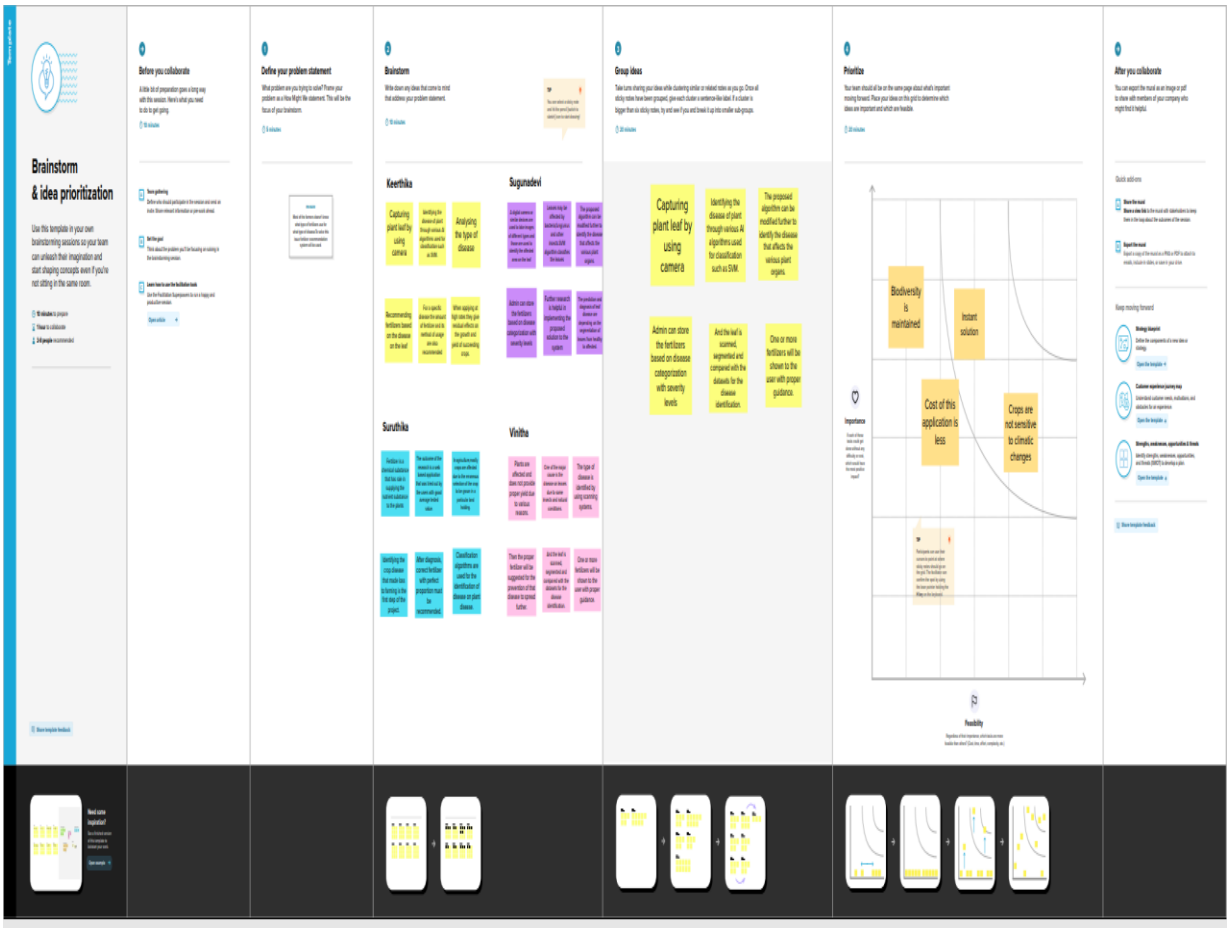
3.1 EMPATHY MAP CANVAS

3.2 Ideation and brainstorming

Idea generation is often closely related to the practice of brainstorming, a specific technique used to generate new ideas. The main difference between conceiving and brainstorming is that conceiving is generally considered an individual activity whereas brainstorming is most often a group activity. Brainstorming is usually done by bringing a group of people together to come up with new ideas in general or to solve a specific problem or situation.

For example, a large company that recently learned it was the subject of a major lawsuit wants to brainstorm executives to publicly respond to the lawsuit being filed. Brainstorming session participants are encouraged to freely express any ideas that come to mind. The reason is that by generating a large number of ideas, brainstorming groups are more likely to find good solutions to the problems they are dealing with.

With the development of several brainstorming software programs such as Bright Idea and Idea Wake, the line between idea generation and brainstorming has become somewhat blurred. These software programs are designed to help the company's employees come up with new ideas to improve their operations and ultimately improve the company's profitability.



3.2 BRAINSTORMING AND IDEA PRIORITIZATION

3.3 Proposed Solution

Proposed solution:

S.No	Paramater	Design
1.	Problem statement	To recommend fertilizers based on the plant leaf disease.
2.	Idea/Solution description	Supplying fertilizer to the crop to achieve yield and quality goals (economic) in a manner that minimizes nutrient losses to the plants.
3.	Novelty/Uniqueness	Introduction of fertilizers for different plant disease and improvement in suggestion of fertilizers with proper guidance.
4.	Social Impact/Customer Satisfaction	Fertilizers replace the nutrients that crops remove from the soil. Without the addition of fertilizers, crop yields and agricultural productivity would be significantly reduced. So, fertilizers are important now-a-days for larger yield. Customers are satisfied when the proper fertilizer with correct guidance is recommended.
5.	Business Model	A fast forward system for recommendation of fertilizers. It connects farmers and retailers through digital signals. The system may reach revenue more than the investment in 2 years.
6.	Scalability of solution	Through this system, we can achieve drought tolerance agriculture. The accuracy of the system is to be increased. So it will never have extinction.

3.3 PROPOSED SOLUTION

3.4 Problem solution fit

Project Design Phase-I - Solution Fit Template **Team ID:** PNT2022TMID13660

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
Farmer, Gardener.

6. CUSTOMER CONSTRAINTS CC
Network connection, Available device, Quality camera.

5. AVAILABLE SOLUTIONS AS
Efficient use of animal manure and organic fertilizers.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P
There are different types of disease so we need to identify them and to suggest fertilizer.

9. PROBLEM ROOT CAUSE RC
Pathogen attacks, improper soil quality, anormal climatic changes.

7. BEHAVIOUR BE
People need to use the recommendation system to address the problem and get suggestions for the problem.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR
Improvement of the system in market levels will trigger the customers to act.

10. YOUR SOLUTION SL
We are going to implement the system with various plant disease and we are going to suggest both organic and inorganic fertilizes that are going to be included in our system.

8. CHANNELS of BEHAVIOUR CH
8.1ONLINE
The system will give result in online mode.

8.1OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

4. EMOTIONS: BEFORE / AFTER EM
lost, worried > Happy, confident.

3.4 PROBLEM SOLUTION FIT

4. REQUIREMENT ANALYSIS

Requirements analysis, also known as requirements engineering, is the process of determining user expectations for new or modified products. These characteristics, called requirements, should be quantifiable, relevant, and detailed. In software development, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis includes frequent communication with system users to determine expectations for specific functionality, resolve conflicts or ambiguities in requirements in response to requests from various users or user groups, to avoid creep and document all aspects of the project development process from start to finish. Instead of trying to match user expectations and requirements, you should focus your energy on ensuring that the final system or product meets your customer's needs. Requirements analysis is a team effort that requires a combination of hardware, software and human factors engineering expertise and interpersonal skills. The purpose of the requirements analysis phase is to translate the needs and high-level requirements specified in the previous phase into clear (measurable and testable), traceable, complete, consistent, and stakeholder-approved requirements.

4.1 Functional Requirements

Functional requirements involves calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is to be accomplished. Behavioural requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	Update native language	The preferred language of the user will be change by the user.
FR-2	Image Capture	Take the image of leaf and check the leaf is captured under given parameters.
FR-3	Image processing	Upload the leaf image by clicking the ok button.
FR-4	Leaf prediction	The kind of leaf and its disease will be predicted and fertilizer will be recommended
FR-5	Recommendation	Both organic and inorganic fertilizer under proper guidance will be recommended to the user.
FR-6	Email notification	The kind of disease and its further prevention will be sent to the user through email.

4.1 FUNCTIONAL REQUIREMENTS

4.2 Non-Functional requirements

In systems engineering and requirements engineering, non-functional requirements (NFRs) are requirements that specify criteria by which the behaviour of a system can be determined, rather than specific behaviour. They face functional requirements that define specific behaviours and functions. Plans for implementing functional requirements are detailed in the system design. Plans for implementing non-functional requirements are detailed in System Architecture in System Architecture. Becausethese are typically architecturally important requirements.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Datasets based on leaves are used for all kinds of leaves. Datasets can be reusable.
NFR-2	**Security**	User information and leaf data are secure. Algorithms used are more secure.
NFR-3	**Reliability**	The image capturing performs consistently well with high quality.
NFR-4	**Performance**	Performance of the system is considerably better and the predictions are good with high accuracy.
NFR-5	**Availability**	It can work in different platforms and easy access of the system will be provided.
NFR-6	**Scalability**	Increase in growth of predicting the results.

4.2 NON-FUNCTIONAL REQUIREMENTS

5. PROJECT DESIGN

A data flow diagram (DFD) is a traditional visual representation of the flow of information in a system. A well-organized DFD can represent a reasonable amount of system requirements graphically. It can be manual, automatic, or a combination of both.

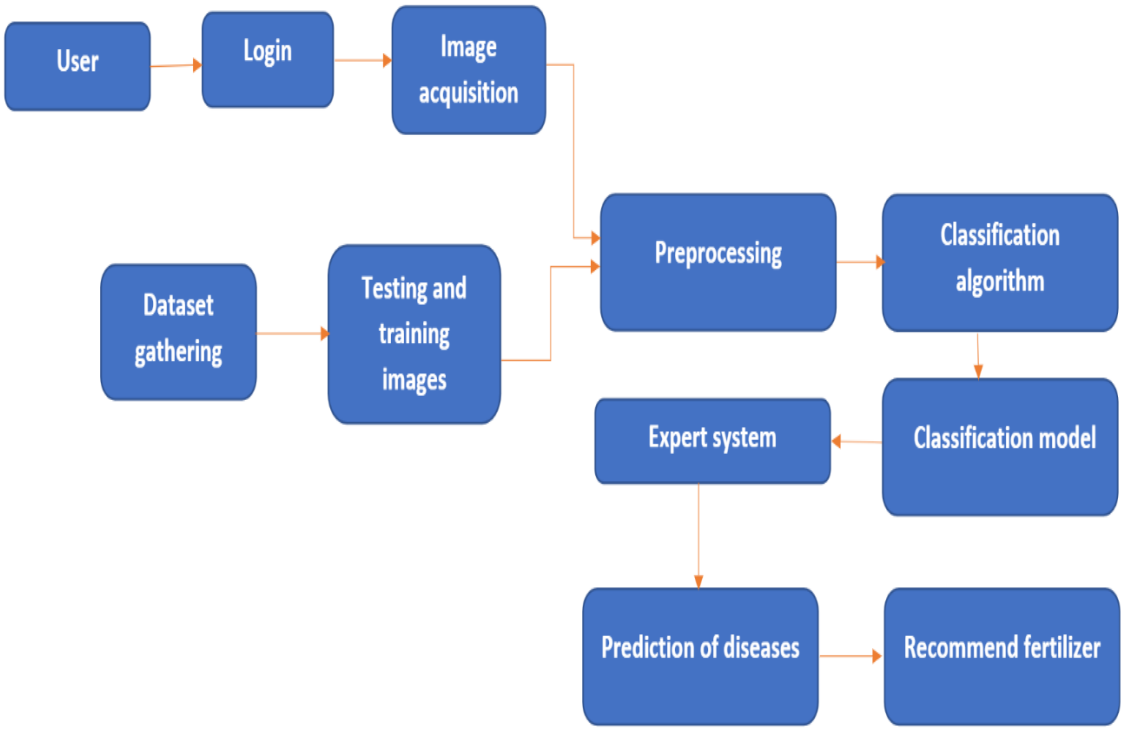
Shows how data enters and exits the system, what changes information, and where data is stored. The purpose of the DFD is to indicate the scope and limits of the overall system. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements. DFDs can be used to run systems or software at any level of abstraction. DFDs can be decomposed into levels that represent increasing information flow and functional detail. The system is then decomposed and described as a multi bubble DFD. The parts of the system represented by each of these bubbles are decomposed and documented as increasingly detailed DFDs.

5.1 Data flow diagram

A data flow diagram (DFD) is a traditional visual representation of the flow of information in a system. A well-organized DFD can represent a reasonable amount of system requirements graphically. It can be manual, automatic, or a combination of both.

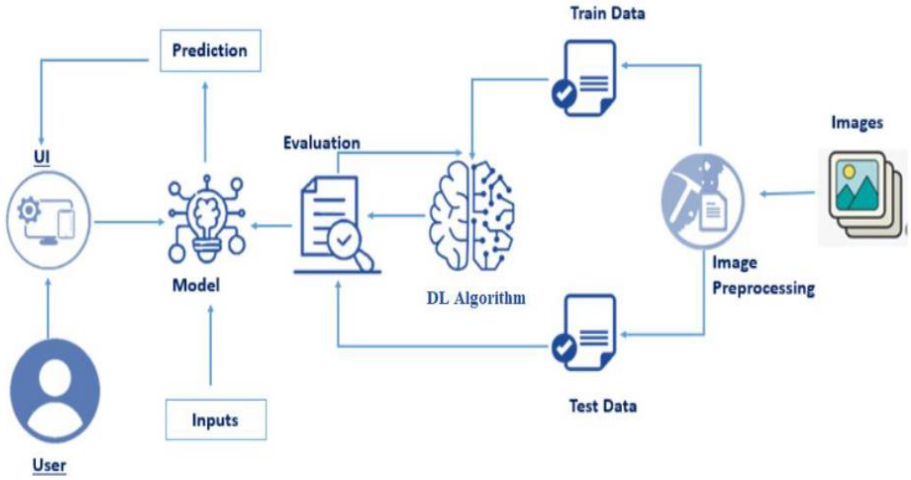
Shows how data enters and exits the system, what changes information, and where data is stored.



5.1 DATA FLOW DIAGRAM

5.2 Solution & Technical Architecture

A solution architect is most like a project manager, ensuring that all stakeholders, including stakeholders, are on the same page and on the right track at every stage. A technical architect leads all activities that lead to a successful implementation of a new application. They suggest combinations of building blocks that provide the best possible solution. This process is very detail oriented and it acts as a link between the enterprise architecture and the technical architecture. Extensive knowledge of the company's technical and commercial internal structure is also required**.**



5.2.1 TECHNOLOGY STACK

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework-PyCharm, anaconda navigator, flask framework.
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	PyCharm
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Web applications to access the system.
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Convolutional Neural Network.

5.2.2 APPLICATION CHARACTERISTICS

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python, CNN
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	Local File system
8.	Machine Learning Model	Purpose of Machine Learning Model	Image Recognition Model, etc.
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

5.2.3 COMPONENTS AND TECHNOLOGIES

5.3 User Stories

A user story is the smallest unit of work in an agile framework. This is an end goal, not a feature expressed from the software user's point of view.

User stories are informal, high-level descriptions of software functionality written from the perspective of an end-user or customer.

The purpose of user stories is to clarify how the work returns a particular value to the customer. Note that a 'customer' is an external end in the traditional sense, he doesn't have to be a user, but could be an internal customer or a colleague within a team-dependent organization.

A user story is a few sentences in plain language outlining a desired outcome. I won't go into details. Requirements are added after the team agrees.

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	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	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-1	As a user, I can see my dashboard and go through the functions provided by the system.	I can access my dashboard	High	Sprint-1
Customer (Web user)	Registration		As a user, I can register for my account through web and login to my web page.			
Customer Care Executive	Login	USN-1	Make a call to the customer care executive and rectify the queries.	Help the user how to access the system.	High	Sprint-1
Administrator	User account control	USN-1	Responsible for carrying out the administration process.	Manage the total team	High	Sprint-1

5.3 USER STORIES

6. PROJECT PLANNING & SCHEDULING

Planning - Planning pertains to the process of creating a plan of which materials and resources will be required to fulfil incoming and forecasted demand. This step is crucial to ensure that you have enough materials and resource capacity available to produce your orders on time. This component pertains to the ‘what’ and ‘how’ of any project: what exactly needs to be achieved and how it will be accomplished.

Scheduling - Scheduling pertains to establishing the timing of the use of specific resources of that organization. In production, scheduling involves developing schedules for workers, equipment, and materials. It reflects on the ‘when’ of a project, by assigning the appropriate resources to get the production plan completed within a period of time. Creating optimized production schedules ensures that your facility is able to reduce costs, increase productivity, and deliver goods to customers on time.

In order to create accurate and realistic production plans that allow manufacturers to react quickly to changes, it is important to have a production plan that is aligned with the resource and material scheduling process. Having any discrepancy or divergence between the planning and scheduling process creates inefficiencies that can be costly for your business. The bigger the divergence, the larger the cost.

6.1 Sprint planning and estimation

Planning:

In sprint planning, the team decides what to build in the next sprint and how to build it. The team decomposes the user's story into tasks, performs task-level estimation, and then commits to the sprint goal. Sprint planning is executed by the Product Owner, Scrum Master, and Team.

In Scrum, each project is divided into time blocks called sprints, which are typically two to four weeks long. At the Sprint Planning Meeting, the team (including the Scrum Master, Scrum Product Managers, and Scrum Team) gets together to decide which backlog items to work on in the next Sprint

Estimation

In Scrum projects, estimates are made by the entire team during the sprint planning meeting. The goal of the estimation is to consider the sprint's user stories by priority and the team's ability to deliver within the sprint's time box.
The Product Owner ensures that prioritized user stories are clear, measurable, and placed at the top of the Product Backlog.
Since the Scrum team has overall responsibility for delivering the product increment, care should be taken to select user stories for the sprint based on the size of the product increment and the effort required to deliver it. be paid. The size of the product increments is estimated in user story her points. Once the size is fixed, the amount of work is estimated based on historical data. That is the amount of work done per user story point called productivity.

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	Collecting plant disease dataset	2	High	Keerthika
Sprint-1		USN-2	Labelling the dataset according to class	1	High	Sugunadevi
Sprint-2	Testing training and creating a model	USN-3	Start initiating the model	2	Low	Suruthika
Sprint-1		USN-4	Adding different layers of CNN (convolution, pooling dense, flatten)	2	Medium	Vinitha
Sprint-1		USN-5	Training the data	1	High	Keerthika,Vinitha
Sprint-3			Testing the data	1	Medium	Sugunadevi
Sprint-3	Flask and framework design		Creating backend framework with flask	2	High	Keerthika,Sugunadevi, Vinitha,Suruthika
Sprint-4			Predicting disease and recommend fertilizer	2	High	Keerthika,Sugunadevi, Vinitha,Suruthika

6.1 PLANNING & ESTIMATION

6.2 Sprint delivery schedule

Sprints span a period of time, so it's important not to waste time during planning and development. And this is where sprint planning comes into play. In case you didn't know, a sprint schedule is a document that outlines the sprint plan from end to end.

This is one of the first steps in the Agile sprint planning process and requires proper research, planning, and communication. Teams often run into problems when creating multiple schedules. This can create conflicts and cause the project to fail in the middle of the cycle. A schedule helps to keep things on track.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	3 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	10 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	17 Nov 2022

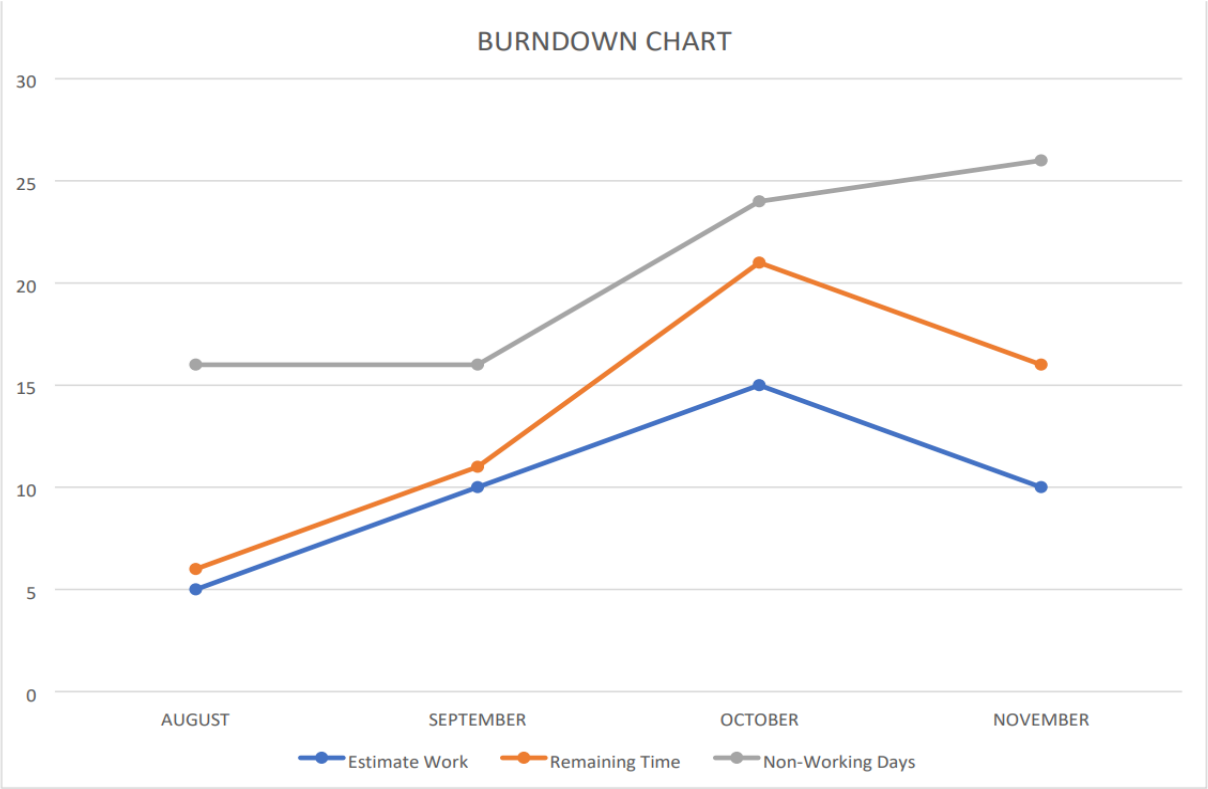
Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.2 SPRINT DELIVERY SCHEDULE

6.3 Reports from JIRA

A burndown chart is a graphical representation of work remaining versus time. It is commonly used in Agile software development methodologies such as Scrum. However, burndown charts can be applied to any project that shows measurable progress over time. Burndown charts often show work on the vertical axis and time on the horizontal axis. Helps predict when all work will be completed. At the daily scrum, the development team updates the sprint burndown to record the rest of the day's work.



6.3 BURNDOWN CHART

7. CODING AND SOLUTIONING

7.1 Feature 1

```
from __future__ import division, print_function

import os

if __name__=='__main__':

    os.environ.setdefault('FLASK_ENV','development')

import numpy as np

import cv2


from tensorflow.keras.models import load_model

from flask import Flask,render_template,request

from werkzeug.utils import secure_filename


app = Flask(__name__)


MODEL_PATH = 'models/my_model.h5'


model = load_model(MODEL_PATH)


print('Model loaded. Start serving...')


print('Model loaded. Check http://127.0.0.1:5000/')
```

```
def model_predict(img_path, model):
    img = cv2.imread(img_path)
    new_arr = cv2.resize(img,(100,100))
    new_arr = np.array(new_arr/255)
    new_arr = new_arr.reshape(-1, 100, 100, 3)
    preds = model.predict(new_arr)
    return preds
@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':

        f = request.files['file']

        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads',f.filename )
        secure_filename(f.filename)
```

```
        f.save(file_path)

    preds = model_predict(file_path, model)

        pred_class = preds.argmax()
```

CATEGORIES = ['Leaf:Pepper bell-->Disease:Bacterial spot-->Suggestions:Washing seeds for 40 minutes in diluted Clorox (two parts Clorox plus eight parts water) is effective in reducing the bacterial population on a seed surface.','Leaf:Pepper bell\nCondition:healthy',

'Leaf:Potato leaf-->Disease:Early blight-->Suggestions:Albina is our highly efficient calcium fertilizer which improves shelf life & firmness of potato tubers and reduces susceptibility to disease & physiological disorders',

'Leaf:Potato-->Condition:healthy',

'Leaf:Tomato-->Disease:Bacterial spot-->Suggestion:Make up a solution of about a teaspoon of Epsom salts per litre (quarter gallon) of water in a spray bottle. Simply wet the foliage on your tomato plants every two weeks using a fine spray setting. It will quickly be absorbed by the leaves. Avoid spraying on hot, sunny days or when rain is imminent' ,

'Leaf:Tomato-->Disease:Early blight-->Suggestion:Use protectant fungicides, mancozeb, chlorothalonil or copper products. Systemic products are available, e.g., strobilurins, although they are expensive and, if used too often, the fungus may develop resistance to them.',

'Leaf:Tomato-->Disease:Late blight-->Suggestion:Use fungicide sprays based on mandipropamid, chlorothalonil, fluazinam, mancozeb to combat late blight. Fungicides are generally needed only if the disease appears during a time of year when rain is likely or overhead irrigation is practiced.',

'Leaf:Tomato-->Disease:Leaf Mold-->Suggestion:Active ingredient chlorothalonil is the most recommended chemical for us on tomato fungus. It can be applied until the day before you pick tomatoes, which is a clear indication of its low toxicity.' ,

'Leaf:Tomato-->Disease:Septoria leaf spot-->Suggestion:Most fungicides registered for use on tomatoes would effectively control Septoria leaf spot. These include maneb, mancozeb, chlorothalonil, and benomyl. Captan is not effective and zineb may be difficult to purchase.',

'Leaf:Tomato-->Disease:Two spotted spider mite-->Suggestion:Apply a miticide spray if mite damage and mites are present on 50 per cent of the leaves and predators are not present.This will effectively control the spread' ,

'Leaf:Tomato-->Disease:Target Spot-->Suggestion:The effective agents are chlorothalonil, copper oxychloride or mancozeb. Treatment should start when the first spots are seen and continue at 10-14-day intervals',

'Leaf:Tomato-->Disease:Yellow Leaf Curl Virus-->Suggestion:Imidacloprid should be sprayed on the entire plant and below the leaves; eggs and flies are often found below the leaves. Spray every 14-21 days and rotate on a monthly basis with Abamectin so that the whiteflies do not build-up resistance to chemicals.',

'Leaf:Tomato-->Disease:Mosaic virus-->Suggestion:To avoid seed-borne mosaic viruses, soak seeds of susceptible plants in a 10% bleach solution before planting.',

'Leaf:Tomato-->Condition:healthy']

return CATEGORIES[pred_class]

```
        return CATEGORIES[pred_class]
    return None

if __name__ == '__main__':
    app.run(debug=False)
```

Output:

```
Serving Flask app 'app'
Debug mode: On
 Running on http://127.0.0.1:5000
```

main.py

```
from fastapi import FastAPI
app = FastAPI(
    title="test",
    description="test",
    version="0.0.1",
)
if __name__ == "__main__":
    import uvicorn
```

```
uvicorn.run(
    "main:app",
    host="0.0.0.0",
    reload=True,
    port=3001,
)
```

Fertilizer code. ipynb

```
import numpy as np
import pickle
import cv2
from os import listdir
import matplotlib.pyplot as plt
EPOCHS = 30
default_image_size = tuple((100,100))
image_size = 0
directory_root = 'plantvillage/'
width=100
height=100
depth=3
```

```
def convert_image_to_array(image_dir):
    try:
        image = cv2.imread(image_dir)
        if image is not None :
            image = cv2.resize(image, default_image_size)
            return img_to_array(image)
        else :return np.array([])
    except Exception as e:
        print(f"Error : {e}")
        return None
import numpy as np
CATEGORIES = ['Pepper__bell___Bacterial_spot','Pepper__bell___healthy',
 'Potato___Early_blight' ,'Potato___Late_blight', 'Potato___healthy',
 'Tomato_Bacterial_spot' ,'Tomato_Early_blight', 'Tomato_Late_blight',
 'Tomato_Leaf_Mold' ,'Tomato_Septoria_leaf_spot',
 'Tomato_Spider_mites_Two_spotted_spider_mite' ,'Tomato__Target_Spot',
 'Tomato__Tomato_YellowLeaf__Curl_Virus', 'Tomato__Tomato_mosaic_virus',
 'Tomato_healthy']
def image(path):
    img = cv2.imread(path)
    new_arr = cv2.resize(img,(100, 100))
    new_arr = np.array(new_arr/255)
```

```
    new_arr = new_arr.reshape(-1, 100, 100, 3)
    return new_arr
model = keras.models.load_model('my_model.h5')
```

Output:

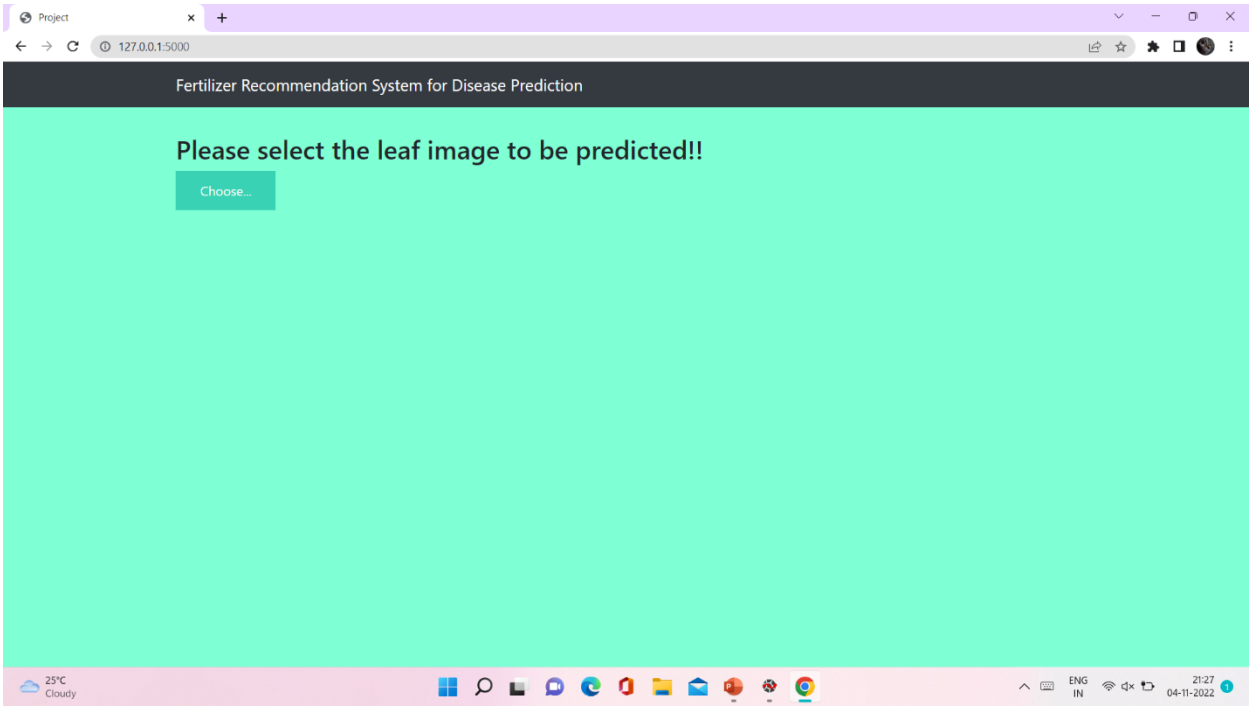
my_model.h5 saved in our local directory.

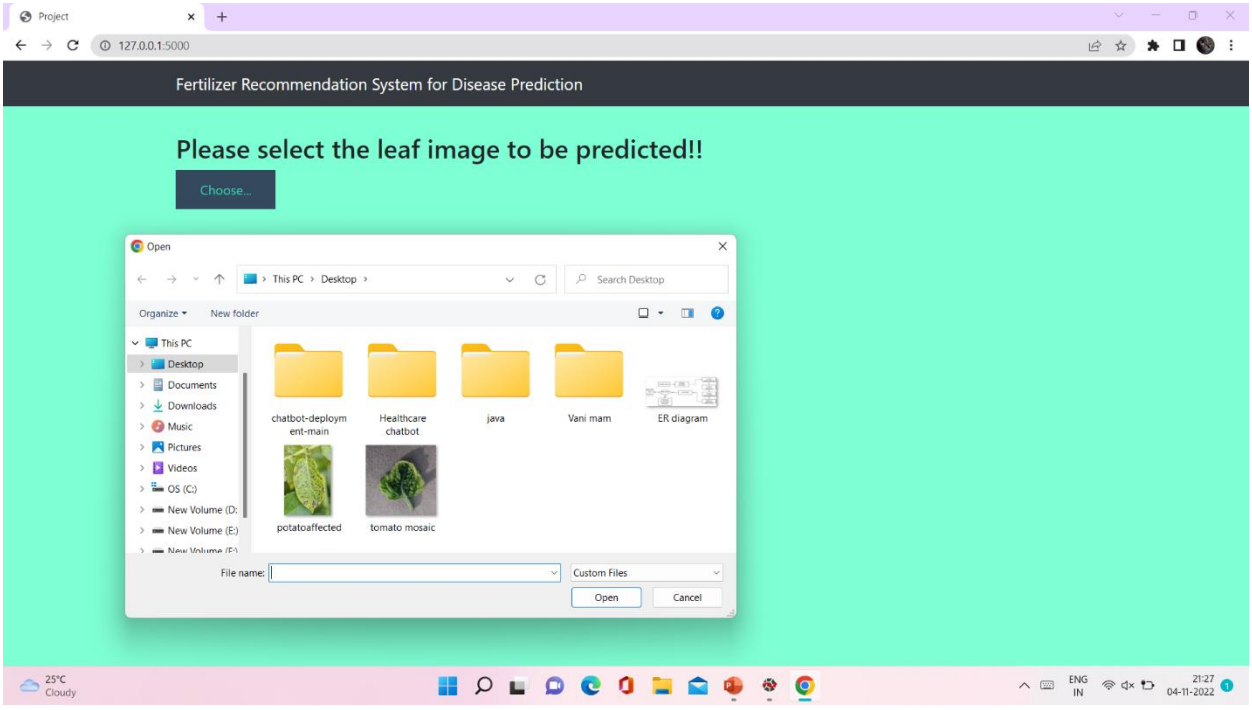
7.2 Feature 2

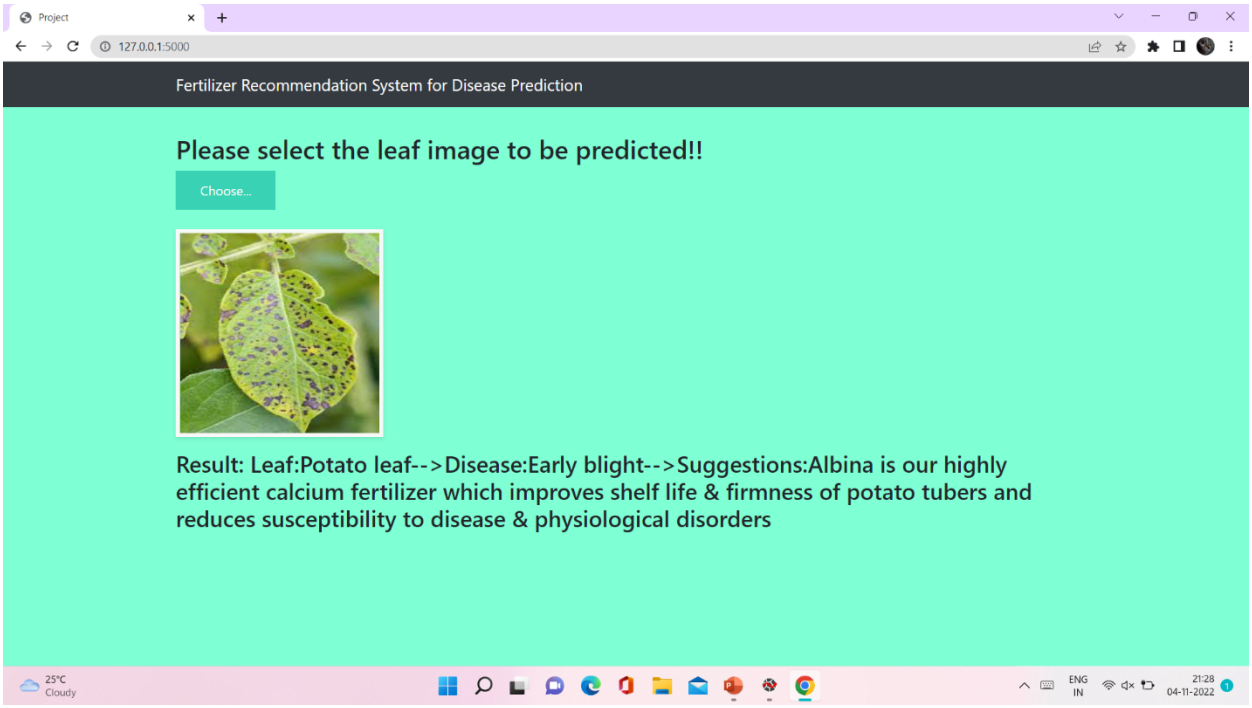
```
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Project</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
    <style>
        body{
            background-color:aquamarine;
```

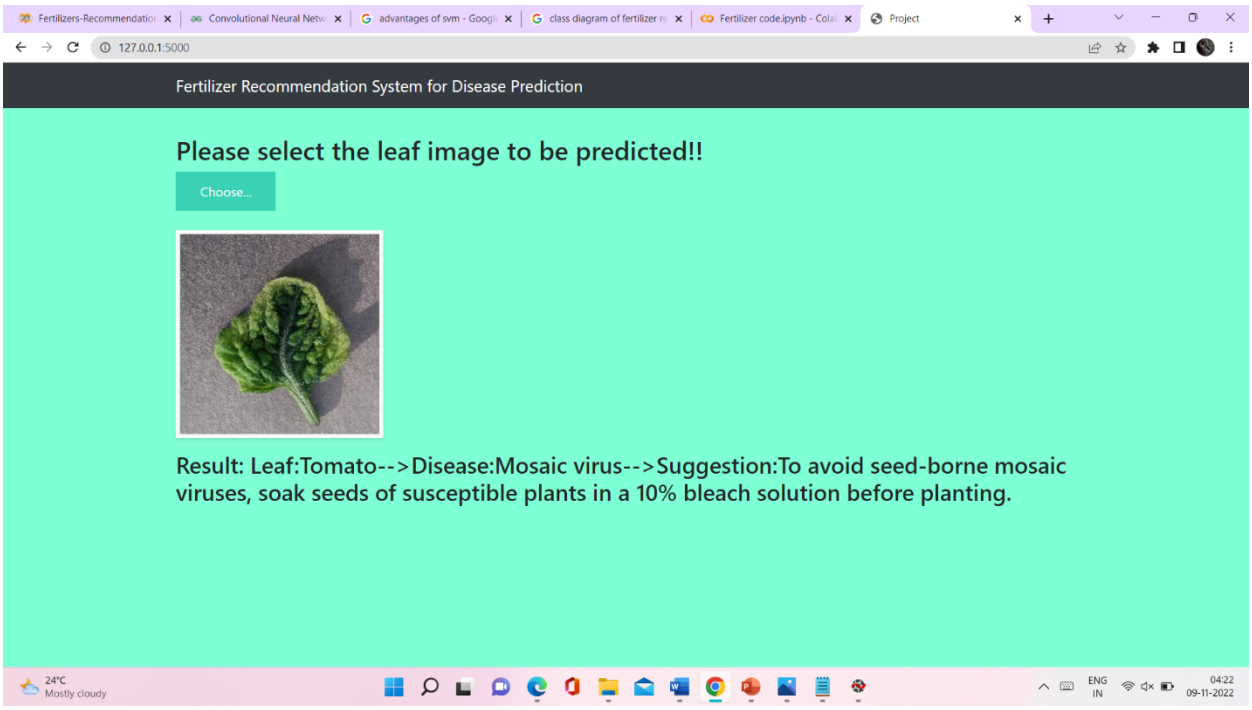
```
        }
    </style>
</head>

<body>
    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">Fertilizer Recommendation System for Disease Prediction</a>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">{% block content %}{% endblock %}</div>
    </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

</html>
```

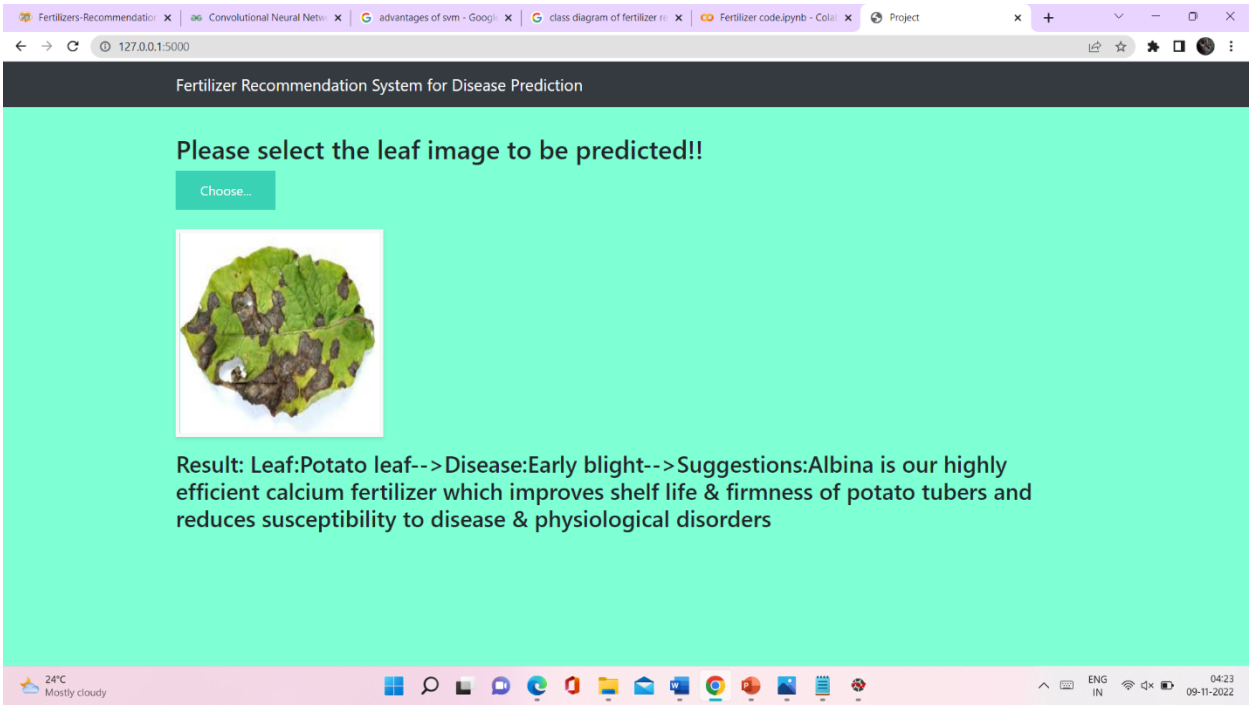
Output:

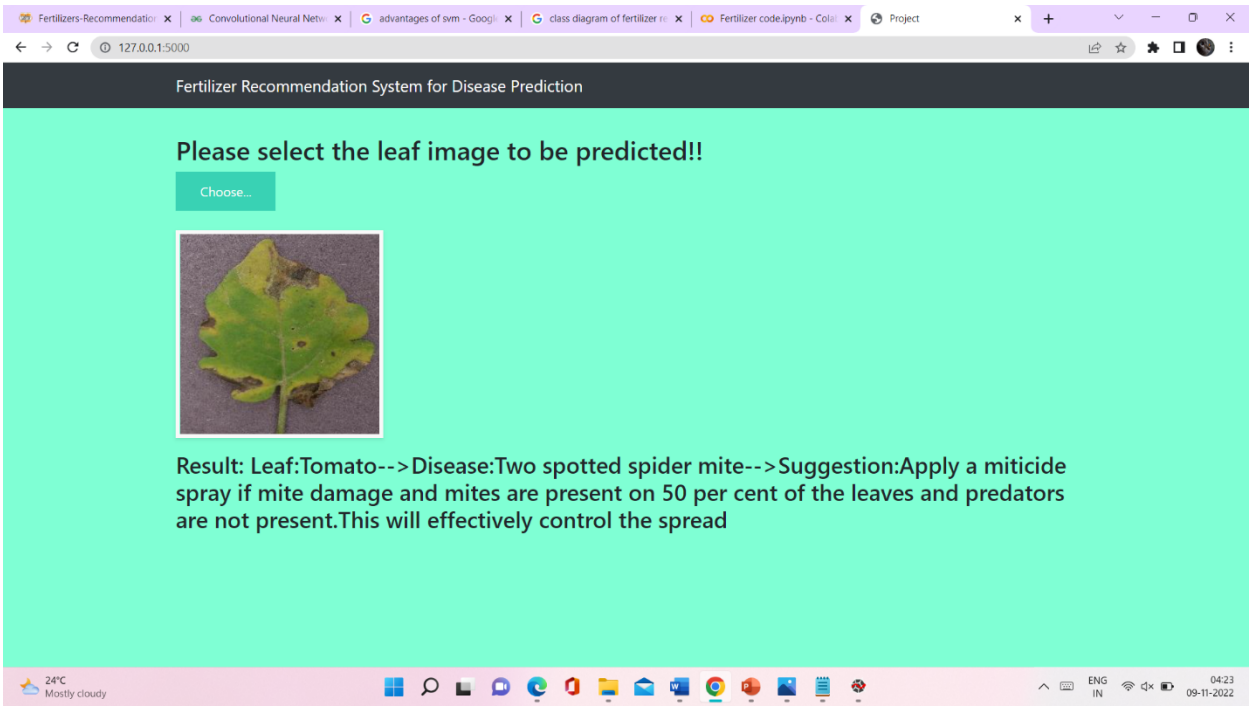

Project
127.0.0.1:5000
Fertilizer Recommendation System for Disease Prediction
Please select the leaf image to be predicted!!
Choose...


Project
127.0.0.1:5000
Fertilizer Recommendation System for Disease Prediction
Please select the leaf image to be predicted!!
Choose...
Open
This PC › Desktop ›
Search Desktop
Organize
New folder
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
New Volume (D:
New Volume (E:)
chatbot-deployment-main
Healthcare chatbot
java
Vani mam
ER diagram
potatoaffected
tomato mosaic
File name:
Custom Files
Open
Cancel


Project
127.0.0.1:5000
Fertilizer Recommendation System for Disease Prediction
Please select the leaf image to be predicted!!
Choose...
Result: Leaf:Potato leaf-->Disease:Early blight-->Suggestions:Albina is our highly efficient calcium fertilizer which improves shelf life & firmness of potato tubers and reduces susceptibility to disease & physiological disorders


Project
127.0.0.1:5000
Fertilizer Recommendation System for Disease Prediction
Please select the leaf image to be predicted!!
Choose...
Result: Leaf:Tomato-->Disease:Mosaic virus-->Suggestion:To avoid seed-borne mosaic viruses, soak seeds of susceptible plants in a 10% bleach solution before planting.


127.0.0.1:5000
Fertilizer Recommendation System for Disease Prediction
Please select the leaf image to be predicted!!
Choose...
Result: Leaf:Potato leaf-->Disease:Early blight-->Suggestions:Albina is our highly efficient calcium fertilizer which improves shelf life & firmness of potato tubers and reduces susceptibility to disease & physiological disorders


127.0.0.1:5000
Fertilizer Recommendation System for Disease Prediction
Please select the leaf image to be predicted!!
Choose...
Result: Leaf:Tomato-->Disease:Two spotted spider mite-->Suggestion:Apply a miticide spray if mite damage and mites are present on 50 per cent of the leaves and predators are not present.This will effectively control the spread

8. TESTING

8.1 Test cases

A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly. Test case designing includes preconditions, case name, input conditions, and expected result. A test case is a first level action and derived from test scenarios.

Input	Actual output	Expected output
Tomato_Septoria_leaf_spot.jpg	Leaf:Tomato-- >Disease:Septoria leaf spot-- >Suggestion:Most fungicides registered for use on tomatoes would effectively control Septoria leaf spot. These include maneb, mancozeb, chlorothalonil, and benomyl. Captan is not effective and zineb may be difficult to purchase	Leaf:Tomato-- >Disease:Septoria leaf spot-- >Suggestion:Most fungicides registered for use on tomatoes would effectively control Septoria leaf spot. These include maneb, mancozeb, chlorothalonil, and benomyl. Captan is not effective and zineb may be difficult to purchase

Pepper_bell___Bacterial_spot.jpg	Leaf:Pepper bell-->Disease:Bacterial spot-->Suggestions:Washing seeds for 40 minutes in diluted Clorox (two parts Clorox plus eight parts water) is effective in reducing the bacterial population on a seed surface	Leaf:Pepper bell-->Disease:Bacterial spot-->Suggestions:Washing seeds for 40 minutes in diluted Clorox (two parts Clorox plus eight parts water) is effective in reducing the bacterial population on a seed surface
Potato___healthy	Leaf:Potato-->Condition:healthy	Leaf:Potato-->Condition:healthy
Potato___Late_blight.jpg	Leaf:Potato leaf-->Disease:Late blight-->Suggestions:Albina is our highly efficient calcium fertilizer which improves shelf life & firmness of potato tubers and reduces susceptibility to disease & physiological disorders'	Leaf:Potato leaf-->Disease:Late blight-->Suggestions:Albina is our highly efficient calcium fertilizer which improves shelf life & firmness of potato tubers and reduces susceptibility to disease & physiological disorders'

8.2 User acceptance testing

User Acceptance Testing (UAT), also known as beta or end-user testing, is defined as testing the software by the user to determine whether it can be accepted or not. This is the final testing performed once the functional, system and regression testing are completed.

The main purpose of this testing is to validate the software against the business requirements. This validation is carried out by the end-users who are familiar with the business requirements.

UAT, alpha and beta testing are different types of acceptance testing.

As the user acceptance test is the last testing that is carried out before the software goes live, obviously this is the last chance for the customer to test the software and measure if it is fit for the purpose.

Need for user acceptance testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

- Developers code software based on requirements document which is their "own" understanding of the requirements and may not actually be what the client needs from the software.
- Requirements changes during the course of the project may not be communicated effectively to the developers.

9. RESULTS

9.1 Performance metrics

Performance indicators are defined as numbers and data that represent an organization's behaviour, capabilities, and overall quality. There are many forms of performance metrics such as sales, profit, return on investment, customer satisfaction, customer ratings, personal ratings, overall quality, and market reputation. Performance metrics can vary significantly across industries. Key Performance Indicators are an integral part of an organization's success. It is important for organizations to choose key performance metrics and focus on these areas. These metrics help guide and measure the success of your organization. Success factors only make sense when they are recognized and tracked.

Acceptance Testing
UAT Execution & Report Submission

Date	18 November 2022
Team ID	PNT2022TMID13660
Project Name	Project – Fertilizer Recommendation System for Disease Prediction
Maximum Marks	4 Marks

1. **Purpose of Document**
   The purpose of this document is to briefly explain the test coverage and open issues of the Handwritten Digit Recognition project at the time of the release to User Acceptance Testing (UAT).

2. **Defect Analysis**
   This reports how the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped					
Won't Fix	1	0	1	0	2
Totals	6	1	4	3	14

IBM project report.pdf - Profile 1 - Microsoft Edge

9.1 PERFORMANCE METRICS

10. ADVANTAGES AND DISADVANTAGES

Advantages

- Fertilizers are quick in providing plant nutrients and restoring the fertility of soil.
- They help to enhance the metabolism of plants.
- Through the recommendation system, the users can be able to get suggestions instantly.
- Fertilizers improve and increase the productivity of many crops.

Disadvantages

- Fertilizers get washed away by water easily and cause pollution.
- They also harm the microbes present in soil.
- They can reduce fertility of soil.
- They are expensive.
- Recommendations are not perfect when the images are blurred and taken with a low intensity light.

11. CONCLUSION

In this project, we have presented a system to recommend fertilizer for the diseased plant leaf. These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes like segmentation, feature extraction and classification and so on. Mostly, the prediction and diagnosis of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves. The proposed method uses SVM to classify tree leaves, identify the disease and suggest the fertilizer. The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result when compared to existing CNN. The accuracy of SVM is similarly larger than CNN. And CNN needs large dataset for more accuracy while SVM deals it with minimum number of datasets.

12. FUTURE ENHANCEMENT

This further research is implementing the proposed algorithm with the existing public datasets. Also, various segmentation algorithms can be implemented to improve accuracy. The proposed algorithm can be modified further to identify the disease that affects the various plant organs such as stems of a leaf and to get a better result in recommending fertilizers.

13. APPENDIX

Source code: https://github.com/IBM-EPBL/IBM-Project-1127-1658375882