TEAM ID	PNT2022TMID30822
PROJECT NAME	FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION
TEAM MEMBERS	620119104060-K.NARMADHA 620119104086-R.SANKARI 620119104099-S.SUSMITHA 620119104096-A.SUBIKSHA

1.INTRODUCTION

1.1. Project Overview

The detection of plant leaf is a very important factor to prevent serious outbreak. Automatic detection of plant disease is essential research topic. The commitment of a plant is very imperative for both human life and condition. Plants do experience the ill effects of ailments, similar to people and creatures. There is the quantity of plant maladies that happen and influences the typical development of a plant. These ailments influence finish plant including leaf, stem, organic product, root, and blossom. More often than not when the illness of a plant has not been dealt with, the plant bites the dust or may cause leaves drop, blossoms and organic products drop and so on. Suitable determination of such illnesses is required for precise ID and treatment of plant sicknesses. Plant pathology is the investigation of plant infections, their causes, methodology for controlling and overseeing them. Yet, the current strategy incorporates human inclusion for order and distinguishing proof of maladies. This strategy is tedious and expensive. Programmed division of illnesses from plant leaf pictures utilizing delicate registering approach can be sensibly valuable than the current one. In this paper, we have presented a strategy named as Bacterial searching improvement based CNN for recognizable proof and characterization of plant leaf illnesses naturally. For doling out ideal weight to CNN we utilize Bacterial searching streamlining (BFO) that further expands the speed and exactness of the system to recognize and arrange the districts tainted of various infections on the plant leafs. The locale developing calculation expands the effectiveness of the system via looking and gathering of seed focuses having regular characteristics for highlight extraction process. Tochipped away at parasitic maladies like basic rust, cedar apple rust, late scourge, leaf twist, leaf spot, and early curse. The proposed strategy achieves higher precision in recognizable proof and characterization of infections. Application of computer vision and image processing strategies simply assist farmers in all of the regions of agriculture. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants. Therefore, the characteristic symptoms are generated based on the differentiation between normal physiological functionalities and abnormal physiological functionalities of the plants. Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants. These different symptoms and diseases of leaves are predicted by different methods in image processing. the prediction and diagnosis of leaves.

1.2. Purpose of Project

PLANT DISEASES PREDICTION:

India is an agricultural country. Farmers have wide range of diversity to select suitable fruit and vegetable crop. Research work develops the advance computing system to identify the diseases using infected images of various leaf spots. Images are captured by digital camera mobile and processed using image growing, then the part of the leaf sport has been used for the classification purpose of the train and test. The technique evolved into the system is both Image processing techniques and advance computing techniques.

Image Analysis Can Be Applied For The Following Purposes:

1. To detect diseased leaf, fruit.
2. To quantify affected area by disease.
3. To find the boundaries of the affected area.
4. To determine the color of the affected area.
5. To identify the Object correctly. Etc.

Disease management is a challenging task. Mostly diseases are seen on the leaves or stems of the plant. Precise quantification of these visually observed diseases, pests, traits has not studied yet because of the complexity of visual patterns. Hence there has been increasing demand for more specific and sophisticated image pattern understanding.

Various Types of Leaf Spot Diseases:

- Bacterial
- Fungal
- Viral

  Most leaf diseases are caused by fungi, bacteria and viruses. Fungi are identified primarily from their morphology, with emphasis placed on their reproductive structures. Bacteria are considered more primitive than fungi and generally have simpler life cycles. It is not only tremendous amount of work but also suffers from two major issues: excessive processing time and subjectiveness rising from different individuals studying leave disease.

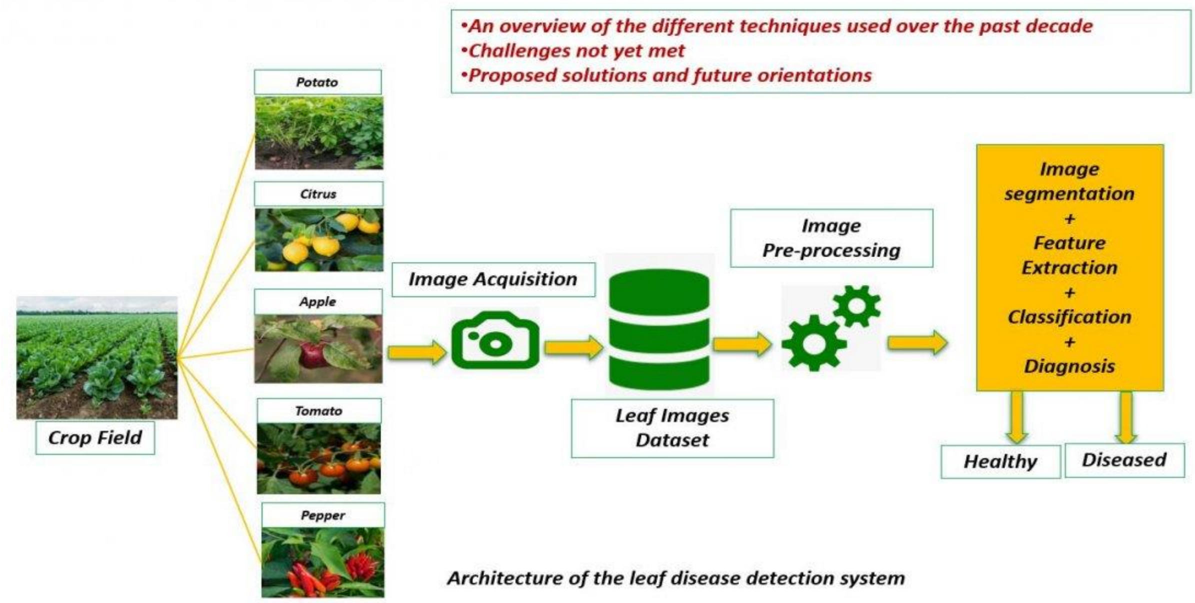


Fig: Various types of diseases

Fungi are identified primarily from their morphology, with emphasis placed on their reproductive structures. Bacteria are considered more primitive than fungi and generally have simpler life cycles. With few exceptions, bacteria exist as single cells and increase in numbers by dividing into two cells during a process called binary fission viruses are extremely tiny particles consisting of protein and genetic material with no associated protein [9]. In biological science, sometimes thousands of images are generated in a single experiment. There images can be required for further studies like classifying lesion, scoring quantitative traits, calculating area eaten by insects, etc. Almost all of these tasks are processed manually or with distinct software packages. It is not only tremendous amount of work but also suffers from two major issues: excessive processing time and subjectiveness rising from different individuals. Hence to conduct high throughput experiments, plant biologist need efficient computer software to automatically extract and analyze significant content. Here image processing plays important role. This project provides image processing techniques used for studding leaf diseases.

It is a mobile app that consists of three modules- crop recommendation, fertilizer recommendation, and plant disease classification. The crop recommendation module recommends the crop based on the values of the different parameters given by the user.

.Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually **Image Processing** system includes treating images as two dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps.

· Importing the image with optical scanner or by digital photography.

· Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.

· Output is the last stage in which result can be altered image or report that is based on image analysis.

Purpose of Image processing

The purpose of image processing is divided into 5 groups. They are:

1. Visualization - Observe the objects that are not visible.
2. Image sharpening and restoration - To create a better image.
3. Image retrieval - Seek for the image of interest.
4. Measurement of pattern – Measures various objects in an image.
5. Image Recognition – Distinguish the objects in an image.

The two types of **methods used for Image Processing** are

Analog image processing

Digital Image Processing.

Analog or visual techniques of image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques.

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information.

2.LITERATURE SURVEY

2.1. Existing System

Precision Botany (PB) refers to the application of new technologies in plant identification. Computer vision can be used in PB to distinguish plants from its species level, so that an identification can be applied on the size and number of plants detected for the classification purpose. Automatic plant identification tasks have gained recent popularity due to its use in quick characterization of plant species without requiring the expertise of botanists. Leaf-based features are preferred over flowers, fruits, etc. due to the seasonal nature of the later and also the abundance of leaves (except may be for the winter season). The current electronic devices for capturing images have been developed to a point where there is little or no difference between the target and its digital counterpart. The success of machine learning for image recognition also suggests applications in the area of identification of plant by herbarium specimens. Once the image of a target is captured digitally, a myriad of image processing algorithms can be used to extract features from it. The use of each of these features will depend on the particular patterns to be highlighted in the image. The automatic classification by computer vision of plants has received increasing attention in the recent past. Historically, identifying an unknown plant species required the consultation of a heavy field guide, where the user had to make sometimes obscure observations of the plant's features and navigate a complex decision tree. The process was nontrivial even for seasoned botanists, and carrying the guides into the field was often impractical – particularly for hikers and other casual users. The advent of highly-portable, computationally-powerful smart phones with large storage capacity presents an opportunity to not only replace and improve the database and decision tree functions of the field guide, but also to create automatic leaf classification applications based on well-known image processing methods currently becoming standardized platforms.

2. LITERATURE REVIEW:

2.1 Plant species identification using Elliptic Fourier leaf shape analysis

Author: Joao Camargo Neto

Plant species can be accurately identified using Elliptic Fourier shape features of whole, extracted leaflets from plant canopies. The EF method combined with principle component analysis and linear discriminate models performed very well. Older plants of the third week had more mature leaves and provided the best leaf images for identifying plant species, demonstrated with an 88.3% classification rate. As leaves develop, their shapes become an important species trade mark. The redroot pigweed plant had the lowest correctly identified rate during both weeks. Redroot pigweed is misclassified with soybean during the third week, because of a similar rounded leaf shape to the trifoliolate soybean leaflet at this growth stage. Some lesser misclassification with sunflower also occurred. By combining the leaflet images for the second and third weeks, an overall species identification accuracy of approximately 88.4% was obtained. Future EF studies should consider improved timing, background lighting, improved camerawork and application to identifying compound leaves. Leaf orientation is important in the EF analysis. To our knowledge, this has not been treated in previous plant species imaging studies. Two angles are needed to describe a leaf plane in three-dimensional space. One of the leaf angles is hard to control or adapt to, but sunlit leaves of many species can present themselves heliotropically toward a light source, such that one could select the best camera angles for full leaf exposure at the top of the canopy. The leaf angle in the plane of the canopy is apparently taken care of by the first EF harmonic, and that is a critical angle for rotationally invariant leaf texture or venation analysis. Additional studies regarding leaf orientation relative to the camera lens might help to resolve classification errors. Future studies are also needed to determine minimal digital image resolutions needed to maintain the highest species discrimination performance.

2.2 First steps toward an electronic field guide for plants

Author: Gaurav Agarwal

This paper aims to construct prototype electronic field guides for the plant species represented in the Smithsonian’s collection in the United States National Herbarium (US). We consider three key components in creating these field guides. First, we are constructing a digital library that is recording image and textual information about each specimen. We initially concentrated our efforts on type specimens (approximately 85,000 specimens of vascular plants at US) because they are critical collections and represent unambiguous species identification. We envision that this digital collection at the Smithsonian can then be integrated and linked digitized type specimen collections at other major world herbaria, such as the New York Botanical Garden, the Missouri Botanical Garden, and the Harvard University Herbaria. However, we soon realized that in many cases the type specimens are not the best variation in a species, so we have broadened use of non-type collections as well in developing the image library. we are developing plant recognition algorithms for comparing and ranking visual similarity in the recorded images of plant species.

2.3 Automatic Plant Leaf Classification for a Mobile Field Guide

Author: David Knight

Historically, identifying an unknown plant species required the consultation of a heavy field guide, where the user had to make sometimes obscure observations of the plant's features and navigate a complex decision tree. The process was nontrivial even for seasoned botanists, and carrying the guides into the field was often impractical – particularly for hikers and other casual users. The advent of highly-portable, computationally-powerful smart phones with large storage capacity presents an opportunity to not only replace and improve the database and decision tree functions of the field guide, but also to create automatic leaf classification applications based on well-known image processing methods currently becoming standardized on mobile platforms. A user friendly application on a popular mobile platform such as Android that is capable of identifying a good number of plant species could achieve widespread adoption, increasing the public's knowledge of and appreciation for their environment. The problem is complicated in this application by complex backgrounds that make image segmentation difficult, but simplified by foreknowledge of one or two desirable crops amongst unwanted weed species. Color and texture information is often sufficient to make this distinction, as opposed to more general applications, where numerous shape features must be acquired. More recently, several groups have approached the problem of automatic leaf classification. Though the groups often use similar digital morphological features, e.g. rectangularity, sphericity, eccentricity, etc

2.4 Leaf shape based plant species recognition

Author: Ji-Xiang Du

Plants can be usually identified according to the shapes, colors, textures and structures of their leaf, bark, flower, seedling and morph. However, it is very difficult for ones to analyze the shapes of flowers, seedling and morph of plants for their complex 3D structures if based on only 2D images. So in our research work, we will identify different plants by leaf features. Leaves are usually firstly clustered so that it is not easy for us to automatically extract features of leaves from the complex background. The leaf image database used in the following experiment is collected and built by ourselves in our lab. The procedure is that we pluck the leaf from plant, put it on the scanner, and then take the digital color image of the leaf directly, or put it on a panel, take digital color image of the leaf with a digital camera. In this way, we can get an image including only one leaf, and the background of the leaf image will be blurred. In this paper, a digital morphological feature based automatic recognition method for plant images was proposed and performed. The data usually contain noises,result in overlapping samples in pattern space, and there may produce some outliers in the training data set. So we need to remove these outliers from the training data set so that a better decision boundary can be easily formed. The fifteen features are used to classify 20 species of plant leaves. In addition, a new moving median centers hyper sphere classifier is adopted to perform the classification. The experimental results demonstrated that the proposed method is effective and efficient. In particular, by comparing with the 1-NN and k-NN classifiers, it can be found that the MMC classifier can not only save the storage space but also reduce the classification time under the case of no sacrificing the classification accuracy.

2.5 Designing a mobile user interface for automated species identification

Author: Sean White

Human interaction is required because the vision algorithms are not perfect. The user can pan and zoom to inspect individual virtual vouchers and compare them with the plant sample. Semantic zooming is accomplished by either tapping on a virtual voucher to zoom in a level and reveal sets of voucher images, identification information, and textual descriptions, or by dragging up or down for continuous zooming. Once the identification has been verified by the botanist, a button press associates the identified species with the sample. A zoom able history of samples can be browsed to recall prior samples and search results. In this paper, we present a new Tablet-PC–based prototype, Leaf-View, which is being field-tested by our botanist colleagues at the Smithsonian Institution, and discuss design decisions based on user feedback and observations. The contribution focuses on interaction and use specific to automated identification in the field. Issues we address include inspection and comparison, immediate and batch processing, feedback from the identification process (via segmentation) as first steps to interacting with a vision algorithm, and the incorporation of identification in the collection process in contrast with post hoc identification.

2.2.REFERENCE:

[1] Neto, J.C., Meyer, G.E., Jones, D.D., Samal, A.K.: Plant species identification using elliptic Fourier leaf shape analysis. Comput. Electron. Agric. 50(2), 121–134 (2006)

[2] Agarwal, G., Belhumeur, P., Feiner, S., Jacobs, D., Kress, J.W.R., Ramamoorthi, N.B., Dixit, N., Ling, H., Russell, D., Mahajan, R., Shirdhonkar, S., Sunkavalli, K., White, S.: First steps toward an electronic field guide for plants. Taxon 55(3), 597–610 (2006)

[3] Knight, D., Painter, J., Potter, M.: Automatic plant leaf classification for a mobile field guide (2010)

[4] Du, J.X.,Wang, X.-F., Zhang, G.-J.: Leaf shape based plant species recognition. Appl. Math. Comput. 185(2), 883–893 (2007)

[5] White, S.M., Marino, D., Feiner, S.: Designing a mobile user interface for automated species identification. In:Conference on Human Factors in Computing Systems, pp. 291–294, San Jose (2007)

[6]. Park, J., Hwang, E., Nam, Y.: Utilizing venation features for efficient leaf image retrieval. J. Syst. Softw. 81(1), 71–82 (2008)

[7]. Wang, X.-F., Huang, D.-S., Du, J.-X., Xu, H., Heutte, L.: Classification of plant leaf images with complicated background. Appl. Math. Comput. 205(2), 916–926 (2008)

[8]. Teng, C.H., Kuo, Y.T., Chen, Y.S.: Leaf segmentation, its 3d position estimation and leaf classification from a few images with very close viewpoints. In: International Conference on Image Analysis and Recognition, pp. 937–946, Halifax (2009)

[9]. Villena-Román, J., Lana-Serrano, S., González-Cristóbal, J.C.: In: Proceeding of CLEF 2011 Labs and Workshop, Notebook Papers. Amsterdam, The Netherlands (2011)

[10]. Paris, S., Halkias, X., Glotin, H.: Participation of LSIS/DYNI to ImageCLEF 2012 plant images classification task. In: Proceeding of CLEF 2012 Labs andWorkshop, Notebook Papers. Rome, Italy (2012)

[11]. Chen, S.Y., Lee, C.L.: Classification of leaf images. Int. J. Imaging Syst. Technol. 16(1), 15–23 (2006)

[12]. Hossain, J., Amin,M.A.: Leaf shape identification based plant biometrics. In: InternationalConference on Computer and Information Technology, pp. 458–463, Dhaka (2010)

[13]. Du, J.X.,Wang, X.-F., Zhang, G.-J.: Leaf shape based plant species recognition. Appl. Math. Comput. 185(2), 883–893 (2007)

[14]. Backes, A.R., Bruno, O.M.: Shape classification using complex network and multi-scale fractal dimension. Pattern Recogn. Lett. 31(1), 44–51 (2010)

[15]. Casanova, D., Florindo, J.B., Gonçalves, W.N., Bruno, O.M.:IFSC/USP at ImageCLEF 2012: plant identification task. In: Proceeding of CLEF 2012 Labs and Workshop, Notebook Papers. Rome, Italy (2012)

[16]. Zhang, S., Lei, Y., Dong, T., Zhang, X.-P.: Label propagation based supervised locality projection analysis for plant leaf classification. Pattern Recogn. 46(7), 1891–1897 (2013)

[17]. Manh, A.G., Rabatel, G., Assemat, L., Aldon, M.J.: Weed leaf image segmentation by deformable templates. J. Agric. Eng. Res. 80(2), 139–146 (2001)
[18]. Tang, X., Liu, M., Zhao, H., Tao, W.: Leaf extraction from complicated background. In: 2nd International Congress on Image and Signal Processing, pp. 1–5, Tianjin (2009)
[19]. Valliammal, N., Geethalakshmi, S.N.: Hybrid image segmentation algorithm for leaf recognition and characterization. In: International Conference on Process Automation, Control and Computing, pp. 1–6, Tamilnadu (2011)
[20]. Kebapci, H., Yanikoglu, B., Unal, G.: Plant image retrieval using color, shape and texture features. Comput. J. 53(1), 1–16 (2010)

2.3. Problem statement Definition

India is agriculture country where in more than 65% population depends on agriculture. The crop losses due to diseases are approximately 10 to 30%. Farmers judge the diseases by their experience but this is not accurate and proper way. Sometimes farmers call the experts for detecting the diseases but this also time consuming way. The diseases mostly on leaves and on stem of plant. The diseases are viral, bacterial, fungal, diseases due to insects, rust, nematodes etc. on plant. It is important task for farmers to find out these diseases as early as possible. Following example shows that how diseases on cotton plant reduces the productivity. There is 20 to 25% of cotton loss due to diseases on plan Accurate, automatic and rapid method for detecting the diseases is required. Diseases decrease the productivity of plant. Which restrict the growth of plant and quality and quantity of plant also reduces. Image processing is best way for detecting and diagnosis the diseases. In which initially the infected region is found then different features are extracted such as color, texture and shape. Finally classification technique is used for detecting the diseases. There are different feature extraction techniques for extracting the color, texture, shape features and classification technique SVM can implement with radial basis function. it could provide with a very inexpensive way of studying and identifying leaves, as no special hardware besides a regular camera and computer processing technology, nowadays ubiquitous, would be required and it could allow for the maintenance of large and possibly specialized leaf databases with reduced effort. On the other hand, the digital approach to leaf classification creates many challenges related with the effectiveness of the available image processing algorithms and with the complexness of the problem itself. Deformed specimens, problems with contour definition, intersection problems arising by either inappropriate digitalization or complex leaf geometry, alterations in leaf aspect, occurring for instance in consequence of contamination by diseases, insect's actions or generic mechanical damage are among some of them. The existing approach starts first by creating device independent color space transformation structure. Thus we create the color transformation structure that defines the color space conversion. The next step is that we apply device-independent color space transformation, which converts the color values in the image to color space specified in the color transformation structure. The color transformation structure specifies various parameters of transformation. A device independent color space is the one where the resultant color depends on the equipment used to produce it. For example the color produced using pixel with a given RGB values will be altered as brightness and contrast on display device used. Thus the RGB system is a color space that is dependent. To improve the precision of the disease detection and classification process, a device independent color space is required. In device independent color space, the coordinates used to specify the color will produce the same color regardless of the device used take the pictures. The mathematical morphology for quasi-flat zone algorithm tries to classify objects (pixels in our case) based on a set of features into number of classes. The classification is done by minimizing the sum of squares of distances between the objects and the corresponding cluster or class centroid.

The network compromises of neurons with "local" or "tuned" receptive fields that can be biologically motivated with somatosensory cells reactive to precise body regions or orientation-selective cells in visual cortex. RBF termed as to be a special class of linear function having a unique feature, of which response decreases or increases monotonically with distance from a center point. The hidden layer is responsible for carrying out non-linear transformation of input and output layer performing linear regression to envision the anticipated outputs. RBF is different from the other networks having multiple hidden layers active at a time. Although there are many radial kernels available to be used for RBF, the Gaussian and Multiquadric are frequently used. A Gaussian RBF having the property of monotonically decreasing with the distance from the center and Multiquadric RBF having the property of monotonically increasing with the distance from the center.

There is the number of reason that could affect the native surroundings and may cause fluctuations in the bacterium population where they exist. The rise in temperature causes a high concentration of nutrient gradients or events due to which all the bacteria in a region are killed or moved to another region. The new replacements are randomly initialized over the search space to handle such situation with some bacteria are liquidated at random with a very small probability. . In other cases, the signs can only be detected in parts of the electromagnetic spectrum that are not visible to humans. A common approach in this case is the use of remote sensing techniques that explore multi and hyper spectral image captures. The methods that adopt this approach often employ digital image processing tools to achieve their goals.Plant recognition has traditionally been done by specialized taxonomists, who use several plant attributes like the general shape of plant leaves, the colour of flowers and the shape and colour of fruits, among other criteria, to distinguish between different species. The development of computer technologies and image and signal processing techniques has been motivating a growing trend to automation and traditional methodologies are being increasingly replaced by novel methods both in industry, applied research and other sectors of society. The problem of plant recognition is not an exception. Currently, there is a shortage on taxonomists and the financial expenditure of this kind of specialized services has been raising [5]. On the other hand, increasingly more sophisticated mobile phones are becoming a commonplace in many people's lives and interest in the development of plant identification applications has been reported. The development of an automatic system for plant recognition through leaf images could provide both specialists and non-specialists with a valuable tool with reduced or no costs. Such a system would have many advantages over the traditional approaches, namely: it could avoid subjective errors done by human operators, as such a system would only use quantitative analysis; it could provide with a very inexpensive way of studying and identifying leaves, as no special hardware besides a regular camera and computer processing technology, nowadays ubiquitous, would be required and it could allow for the maintenance of large and possibly specialized leaf databases with reduced effort.

3.IDEATION AND PROPOSED SOLUTION.

3.1. Empathy Map Canvas

Empathy Map Canvas is a tool to understand customers, provided by XPLANE Company. Using this tool, you can easily obtain customer characteristics. Of course, its uniqueness lies in the fact that it operates beyond demographic characteristics such as living area or income level.

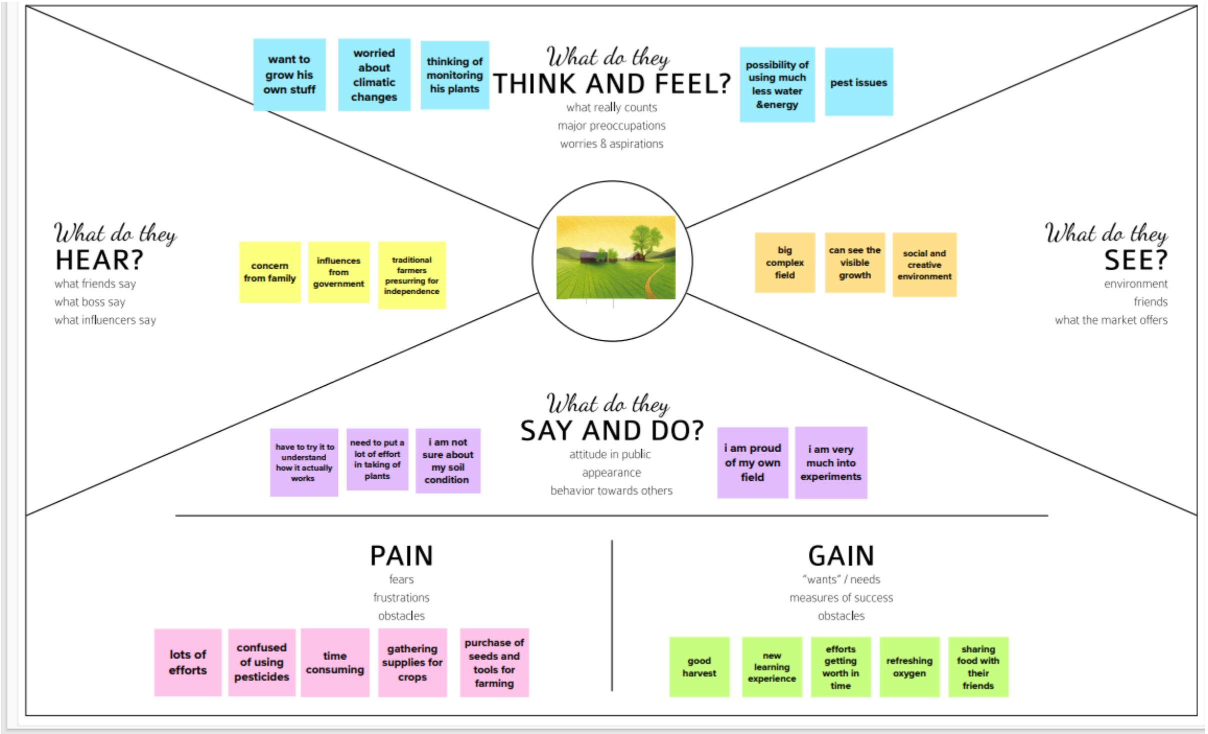
Empathy Map Canvas actually provides a better understanding of the environment, behavior, concerns, and desires of the customer. This map makes it easier for us to identify potential customers.

These three parts include the following:

1. Customer profile

2. Customer attitude based on the surroundings and the environment

3. The customer's personal thoughts and feelings



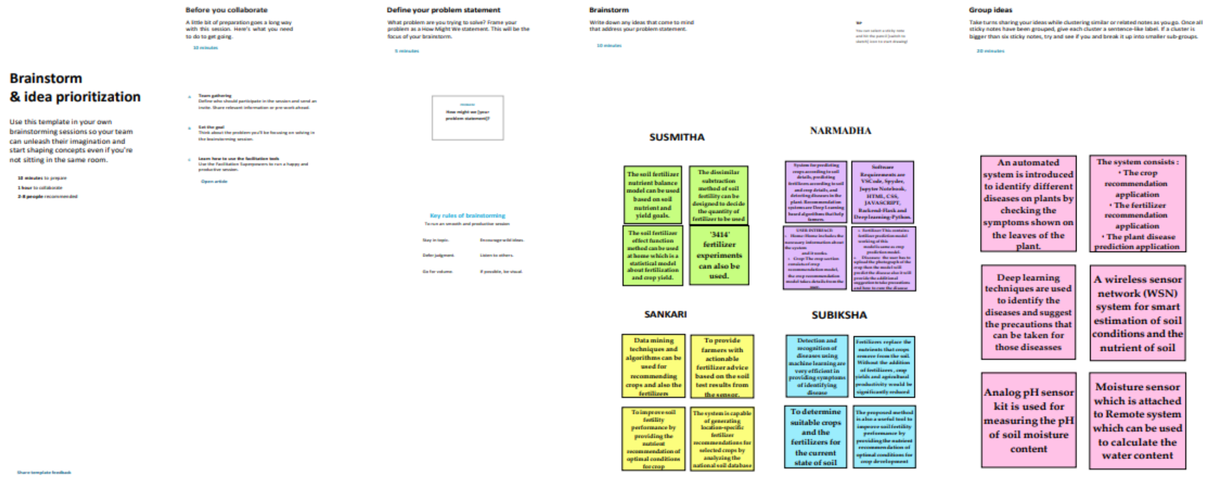
3.2. Ideation and Brainstorming

The Idea Prioritization template is an organizational tool that ranks an idea based on its feasibility and impact. Use this template helps you visually compare the merits of multiple ideas. When you visit any website, it may store or retrieve information on your browser, mostly in the form of cookies.

Idea prioritization is one of the steps of reducing ideas in a Requirements elicitation technique (Brainstorming). To prioritize ideas, **The Hundred Dollar test** is used widely. The Hundred Dollar test: This method works on the strategy of providing fictional money to the participants as their voting tool.

1. Firstly, a goal is defined to understand what the main purpose of brainstorming is.
2. Once we have an end-goal to achieve or a problem to solve, various challenges that come along are explored.

3. Furthermore, different aspects of the problem or situation are explored and we list down ways to overcome the challenges.
4. There is no structure in brainstorming, and no idea is considered wrong. All ideas are noted during the brainstorming sessions, and some can even be clubbed together.



3.3. Proposed Solution

The best fertilizer for every particular crop is also a challenging task. And the other and most important issue is when a plant gets caught by heterogeneous diseases that effect on less amount of agriculture production and compromises with quality as well. The productivity of the Radial Basis Function Neural Network is additionally improved by utilizing district developing strategy hunting down seed focuses and gathering them having comparable properties that assistance in highlight extraction process. BFO with its imitating ability and multi-ideal capacity confirms to be a productive and ground-breaking instrument for instating the heaviness of CNN and preparing the system that can accurately distinguish diverse areas on plant leaf picture with high union speed and exactness. Plant diseases have turned into a big problem as it can cause significant reduction in both quality and quantity of agricultural products. In our proposed work, we center around distinguishing proof and characterization of plant illnesses utilizing some computational knowledge approach. The proposed strategy utilizes CNN that is prepared with the assistance of Bacterial Foraging Optimization (BFO), to locate the influenced district by means of various illnesses present on plant clears out. CNN is the extraordinary direct capacity having a novel ability of which increments or reductions monotonically with separation from the middle point fit for taking care of the multifaceted nature of the influenced district exists on the plant leaf disease. A fertilizer recommendation is the research-based set of guidelines, or management practices, for supplying fertilizer to the crop to achieve yield and quality goals (economic) in a manner that minimizes nutrient losses to the environment. Fertilizers most commonly enter water sources by surface runoff and leaching from agricultural lands. Large amounts of nitrogen and phosphorous are present in the runover.

Plants have become an important source of energy, and are a fundamental piece in the puzzle to solve the problem of global warming. There are several diseases that affect plants with the potential to cause devastating economical, social and ecological losses. In this context, diagnosing diseases in an accurate and timely way is of the utmost importance. There are several ways to detect plant pathologies. Some diseases do not have any visible symptoms associated, or those appear only when it is too late to act

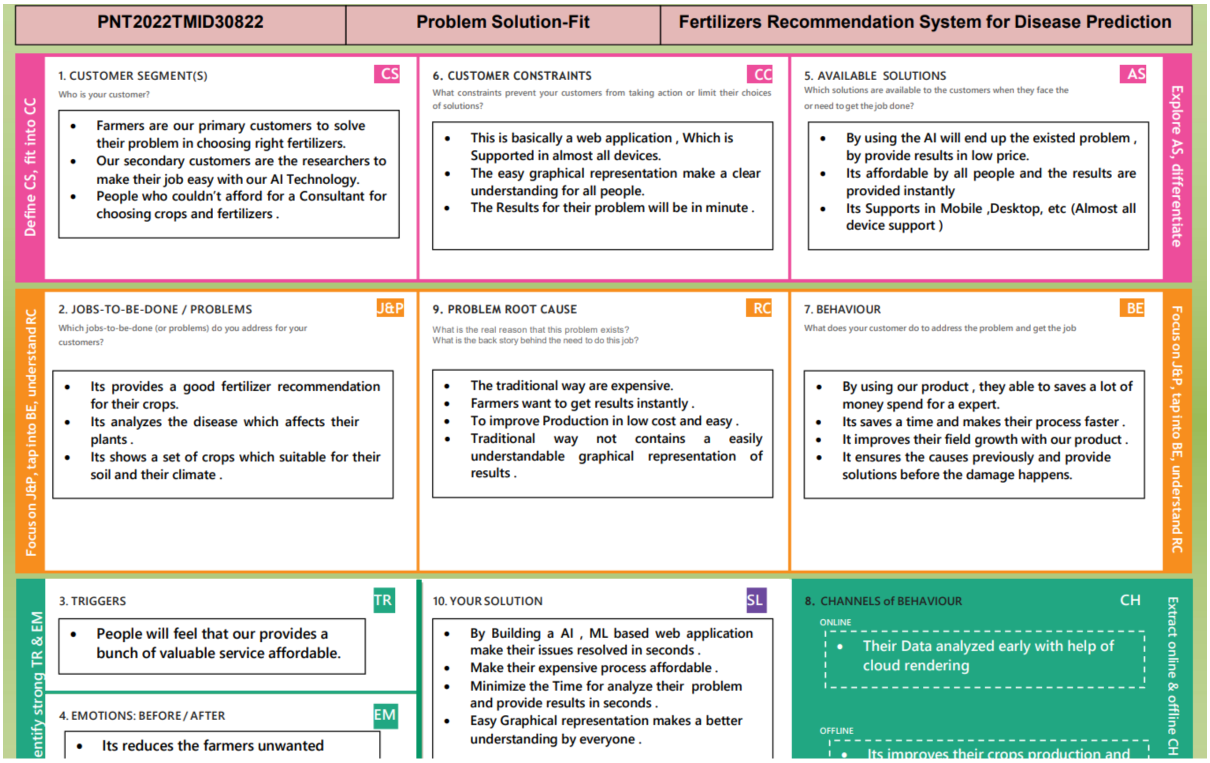
SNO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality.
2	Idea / Solution description	The solution to the problem is to provide a smart user friendly recommender system to the farmers. By providing their soil details like nitrogen , phosphorous, potassium, pH level , farmer get the idea of which fertilizer is best for the crop. By providing an image of a leaf farmer gets an idea of which disease caught their crop and they also suggest how you can prevent it.
3	Novelty / Uniqueness	Create a system for predicting crops according to soil details, predicting fertilizers according to soil and crop details, and detecting diseases in the plant.
4	Social Impact / Customer Satisfaction	Farmers are unaware of which crop to grow, and what is the right time and place to start due to uncertainty in climatic conditions. The usage of various fertilizers is also uncertain due to changes in seasonal climatic conditions and basic assets such as soil, water, and air. In this scenario, the crop yield rate is steadily declining. So this application can be more useful for smart farming.
5	Business Model (Revenue Model)	In addition to providing information about the use of fertilizers to the diseased crops , it also provides information about what crops can grown in the provided proportion of soil .
6	Scalability of the solution	Provide nutrients not available in the soil. Replace nutrients removed at harvest. Balance nutrients for better produce quality and higher yield using artificial intelligence . Scalability is quick and high and also very simple to do .

3.4. Problem Solution fit

Problem-solution fit is a term used to describe the **point validating that the base problem resulting in a business idea really exists and the proposed solution actually solves that problem**. Validate that the problem exists: When you validate your problem hypothesis using real-world data and feedback.

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.

1. A Minimum Viable Product (MVP).
2. Satisfied early adopters (earlyvangelists) who use your MVP.
3. A validated problem that you solve for the earlyvangelists.



4.REQUIREMENT ANALYSIS

4.1. Functional Requirement

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 Mobile number Registration through Gmail
FR-2	User Confirmation	Confirmation via OTP Confirmation via Email
FR-3	User Country	User should select the country to which they belong.
FR-4	Crop details	User can interact through providing details like image of the crop and soil or providing details of the soil like nitrogen, phosphorous, potassium, pH level.
FR-5	Prediction	The system will predict the issue from User details through train set and test data.
FR-6	Suggestion and Prevention	The system will suggest the solution to the issue through image or description.

4.2. Non-Functional Requirement

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system is highly user friendly as the User can provide details and get suggestions from wherever they are. User can easily provide the details of their crop issue and get prevention methods and detects if the crop is affected by diseases.
NFR-2	**Security**	The system provides good security. The system wants the User to provide only their Country and their crop details to detect the disease, suggest the solution and measures to prevent it.
NFR-3	**Reliability**	The system will operate in all kind of environments with proper User and crop details. Farmers are unaware of which crop to grow, and what is the right time and place to start due to uncertainty in climatic conditions. So this application can be more useful for smart farming.

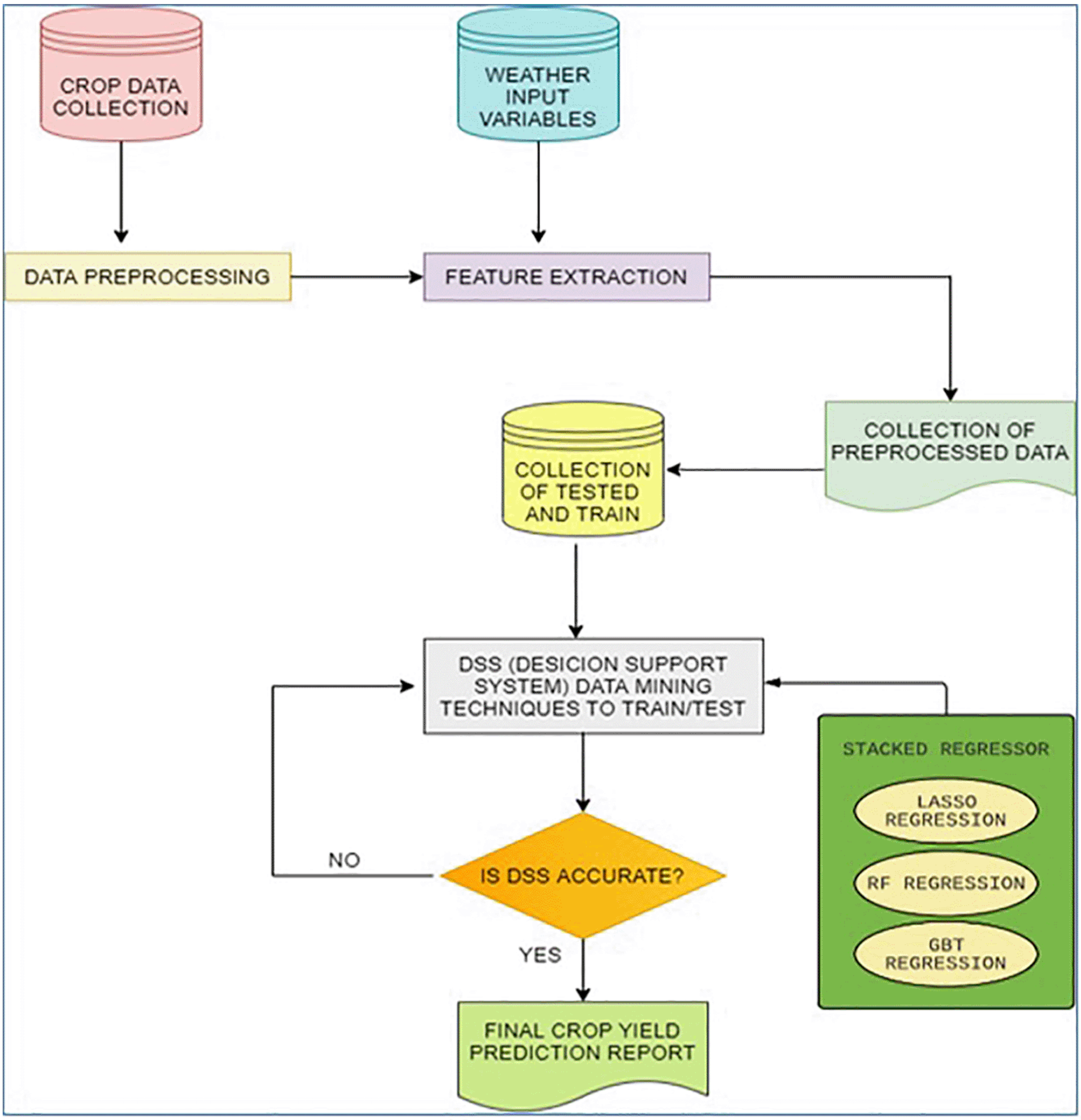
NFR-4	**Performance**	This application provide most accurate prediction of diseases in crop and suggest the required methods to cure it and provide information like what kind of fertilizers need to be used. This application will be a great support to farmers wherever they are.
NFR-5	**Availability**	The system can be used by any farmers across the list of Countries mentioned in the application. Wherever the farmer is, they can use this application and get benefit from accurate identification of plant diseases which is essential to ensure high quantity and best quality of crops.
NFR-6	**Scalability**	Provide nutrients not available in the soil. Replace nutrients removed at harvest. Balance nutrients for better produce quality and higher yield using artificial intelligence. Scalability is quick and high and also very simple to do.

5.PROJECT DESIGN

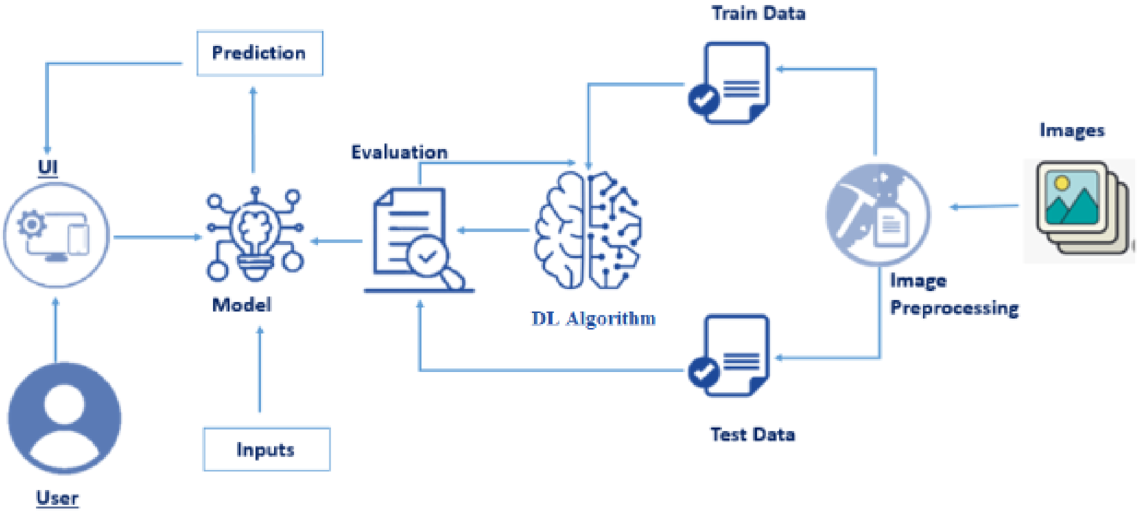
5.1.Data Flow Diagrams

A data flow diagram (DFD) is **a** graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM).

5.2. Solution and Technical Architecture



Solution Architecture

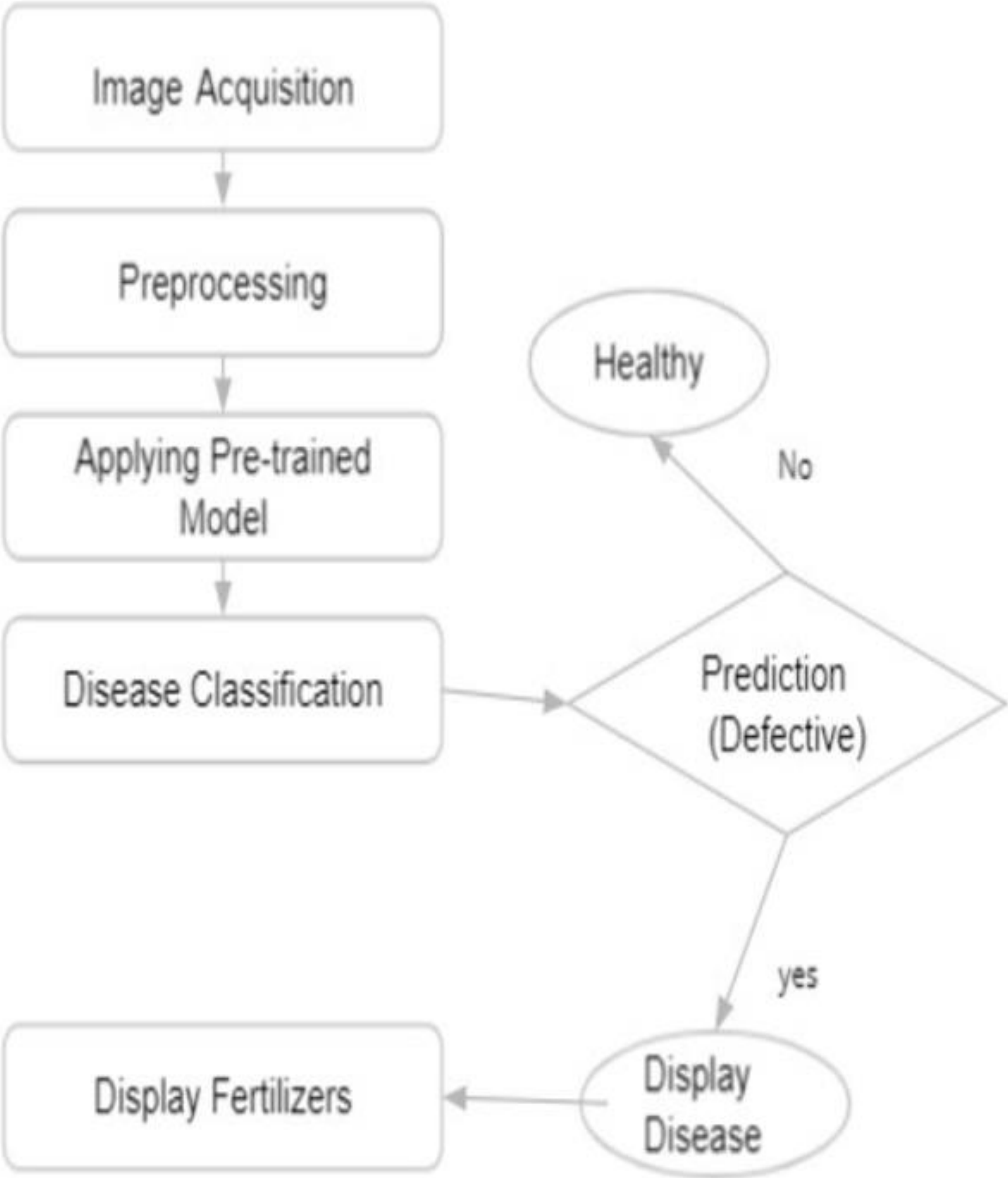


The GoogLeNet engineering comprises 22 layers (27 layers including pooling layers), and a piece of these layers are an aggregate of 9 inception modules. GoogLeNet is a sort of convolutional neural network dependent on Inception engineering. It uses Inception modules, which permit the network to pick between different convolutional channel sizes in each square. An Inception network stacks these modules on top of one another, with periodic max-pooling layers with stride 2 to split the resolution of the grid.

Inceptionv3 is a convolutional neural network for aiding image analysis and object recognition which has the accompanying feature - Using 5x5 convolution to two 3x3 convolution tasks to further develop computational speed. Inception V3 by Google is the third form in a progression of Deep Learning Convolutional Architectures. Inception V3 was prepared utilizing a dataset of 1,000 classes from the first ImageNet dataset which was prepared with more than 1 million preparing pictures, the Tensorflow version has 1,001 classes which is expected to be an extra background class not utilized in the first ImageNet. Inception V3 was prepared for the ImageNet Large Visual Recognition Challenge where it's anything but a first next in line.

MobileNetV2 is a convolutional neural network design that looks to perform well on mobile gadgets. It depends on a rearranged residual construction where the residual associations are between the bottleneck layers. The transitional extension layer utilizes lightweight depthwise convolutions to filter features as a source of non-linearity. All in all, the design of MobileNetV2 contains the underlying completely convolution layer with 32 filters, trailed by 19 residual bottleneck layers.

Flow chart diagram



5.3. User Stories

User stories are short statements about a feature, written from a user's perspective. A well-defined user story does not spell out the exact feature, but rather what the user aims to achieve, to give agile teams the freedom to identify the best possible way to implement the feature.

The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer. Note that "customers" don't have to be external end users in the traditional sense, they can also be internal customers or colleagues within your organization who depend on your team.

User stories are a few sentences in simple language that outline the desired outcome. They don't go into detail. Requirements are added later, once agreed upon by the team.

The user has to give an input image of the infected plant to the ML model and the model classifies the image explaining why the disease has occurred to the plant. It also suggests remedies to cure the plant cure disease.

6.PROJECT PLANNING & SCHEDULING

6.1. Sprint Planning & Estimation

A customer journey is the **end-to-end experience a customer has with your brand or business**. When you create a successful customer journey, you're able to reduce costs, increase revenue, and establish and nurture customer loyalty.

A customer journey outlines the different steps your customers take to become customers. Without the customer journey, your marketing funnel couldn't exist. A marketing funnel helps you market your products and services to customers based on where they are in their customer journey. For example, someone researching products is at the top of the marketing funnel or at the very beginning stages of their journey.

Ultimately, a customer journey map tells a story about how customers interact with your brand, including how they first discovered your business to whether or not they'll make a repeat purchase. The journey lays out different interactions someone could have with your brand, although not every customer needs to use all those touch points before converting Implements Guided active contour method. Unconstrained active contours applied to the difficult natural images. Dealing with unsatisfying contours, which would try and make their way through every possible grab cut in the border of the leaf. The proposed solution is used the polygonal model obtained after the first step not only as an initial leaf contour but also as a shape prior that will guide its evolution towards the real leaf boundary.

Vectors are constructed based on leaf features such as color, shape, textures. Then hyperplane constructed with conditions to categorize the preprocessed leaves and also implement multiclass classifier, to predict diseases in leaf image with improved accuracy.

Before you collaborate
Define your problem statement
Brainstorm
Group ideas
Prioritize
After you collaborate
Brainstorm
& idea prioritization
SUSMITHA
NARMADHA
SANKARI
SUBIKSHA
The soil fertilizer nutrient balance model can be used based on soil nutrient and yield goals.
'3414' fertilizer experiments can also be used.
An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.
The system consists : • The crop recommendation application • The fertilizer recommendation application • The plant disease prediction application
Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases
A wireless sensor network (WSN) system for smart estimation of soil conditions and the nutrient of soil
Analog pH sensor kit is used for measuring the pH of soil moisture content
Moisture sensor which is attached to Remote system which can be used to calculate the water content
To determine suitable crops and the fertilizers for the current state of soil
To provide farmers with actionable fertilizer advice based on the soil test results from the sensors.
Data mining techniques and algorithms can be used for recommending crops and also the fertilizers
Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases
Analog pH sensor kit is used for measuring the pH of soil moisture content
A wireless sensor network (WSN) system for smart estimation of soil conditions
An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.
Importance
Feasibility

6.2. Sprint Delivery and Schedule

SNO	PARAMETER	DESCRIPTION
1	Problem Statement (Problem to be solved)	Agriculture is the most important sector in today's life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality.
2	Idea / Solution description	The solution to the problem is to provide a smart user friendly recommender system to the farmers. By providing their soil details like nitrogen , phosphorous, potassium, pH level , farmer get the idea of which fertilizer is best for the crop. By providing an image of a leaf farmer gets an idea of which disease caught their crop and they also suggest how you can prevent it.
3	Novelty / Uniqueness	Create a system for predicting crops according to soil details, predicting fertilizers according to soil and crop details, and detecting diseases in the plant.
4	Social Impact / Customer Satisfaction	Farmers are unaware of which crop to grow, and what is the right time and place to start due to uncertainty in climatic conditions. The usage of various fertilizers is also uncertain due to changes in seasonal climatic conditions and basic assets such as soil, water, and air. In this scenario, the crop yield rate is steadily declining. So this application can be more useful for smart farming.
5	Business Model (Revenue Model)	In addition to providing information about the use of fertilizers to the diseased crops , it also provides information about what crops can grown in the provided proportion of soil .
6	Scalability of the solution	Provide nutrients not available in the soil. Replace nutrients removed at harvest. Balance nutrients for better produce quality and higher yield using artificial intelligence . Scalability is quick and high and also very simple to do .

Velocity is a measure of a team's rate of progress. It is calculated by summing the number of story points assigned to each user story that the team completed during the iteration.

ways of planning an iteration,

01. velocity-driven iteration planning
02. commitment-driven iteration planning

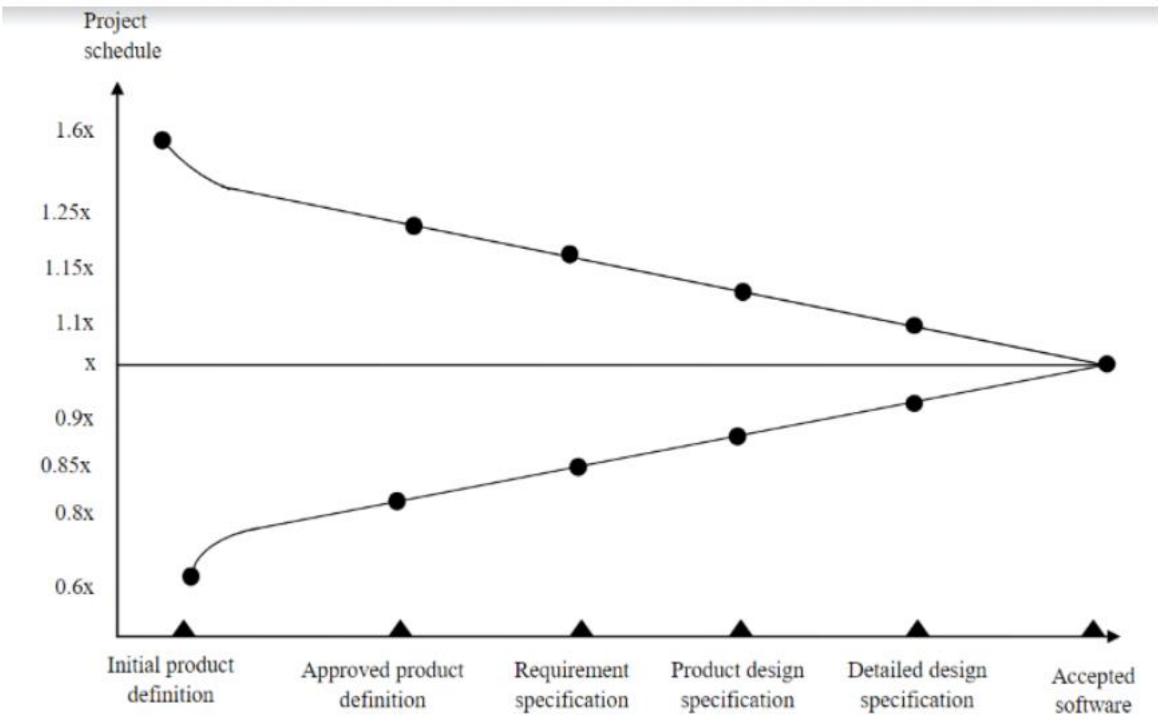
Commitment-Driven Iteration Planning

Here, rather than creating an iteration plan that according to the number of story points or ideal days already completed in the prior iteration, the team is asked to add stories to the iteration one-by-one until they can commit to completing no more.

Velocity Driven Iteration Planning

First, they identify the target velocity for the coming iteration. The team then selects an iteration goal, which is a general description of what they wish to accomplish during the coming iteration. After selecting an iteration goal, the team selects the top priority user stories that support that goal. Stories are selected as necessary for the sum of their ideal day or story point estimates to equal the target velocity. Finally, each selected story is split into tasks and each task is estimated.The commonly used assumption by most teams is that their velocity in thenext iteration will equal the velocity of the most recent iteration. By doing an industry survey we found that many teams select the velocity of the most recent iteration for as the basis of next iterations with some adjustments.

According to many resources including [1], the most preferred way to forecast velocity is to run at least 3 iterations and then estimate velocity from the observed velocity during the one to three iterations.



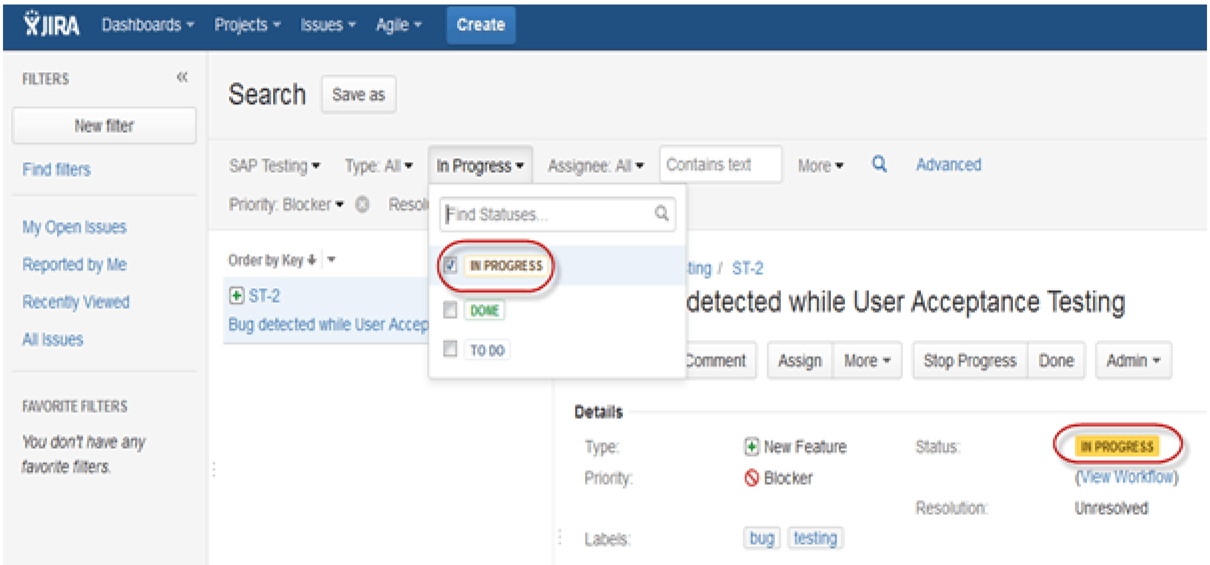
Source : Effort Estimation in Agile Development using Story Points by Evita Coelho and Anirban Basu

If the team has run a single Iteration, the “initial product definition” milestone is considered and the corresponding range is used. If two Iterations are completed, the range corresponding to the “approved product definition” milestone is used, and so on. For example, a team has run three iterations with an average velocity of 50 during that period. For three iterations the appropriate range is 85% to 115%. This means that if the team’s average velocity is 50 after three iterations, their actual true velocity by the end of the project will probably be in the range 42–58.

6.3. Reports from Jira

JIRA is a tool developed by Australian Company Atlassian. This software is used for **bug tracking, issue tracking,** and **project management**. The JIRA full form is actually inherited from the Japanese word “Gojira” which means “Godzilla”. The basic use of this tool is to track issue and bugs related to your software and Mobile apps.

It is also used for project management. The JIRA dashboard consists of many useful functions and features which make handling of issues easy. Some of the key features are listed below. Now in these Jira tutorials, let’s learn JIRA Defect and Project tracking software with this Training Course.



7. CODING AND SOLUTIONING

7.1. Feature1

Plant infection can easily stifle growth and have a negative influence on output. Every year, a financial hardship of up to $20 billion is assessed all across the world. The most difficult issue for analysts is dealing with a variety of situations. In addition, traditional tactics rely on pros, encounters, and guides, but the majority of them are pricey, time-consuming, and labor-intensive, with difficulty in accurately recognising them. Hence, a fast and exact approach to identify plant infections appears so critical for the good thing about trade and biology to agriculture. Disease control procedures can be a waste of money and time if the disease isn't properly identified, and it can lead to more plant loss. Our project proposes a deep learning-based model that will be trained with photos of healthy and diseased crop leaves from a dataset. The model will achieve its goal by categorising photos of leaves into unhealthy categories based on defect patterns.

7.1. Feature 2

Support vector machines (CNNs) are a set of related supervised learning methods used for classification and regression. Supervised learning involves analyzing a given set of labeled observations (the training set) so as to predict the labels of unlabelled future data (the test set). Specifically, the goal is to learn some function that describes the relationship betweenobservations and their labels. Multiclass CNN aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements. The dominant approach for doing so is to reduce the single multiclass problem into multiple binary classification problems. Common methods for such reduction include: building binary classifiers which distinguish between (i) one of the labels and the rest (one-versus-all) or (ii) between every pair of classes (one-versus-one). Classification of new instances for the one-versus-all case is

II. RELATED WORK

In 2018, Komal Bodake et al [2], developed a soil based fertilizer recommendation system that can be used for regional soil analysis. The advanced farming involves various techniques as IOT, Cloud computing and data mining. This helps the farmers to gather details regarding the fertilizers he can use from his soil sample. The tool was constructed in such a way involving regional languages. This makes it understandable to all the farmers and yield .

In 2019, R.Neela et al [4], proposed a new method for finding leaf diseases in plants. Plant disease, especially on leaves, is one of the major factors for reduction in both quality and quantity of the food crops. Finding the leaf disease is an important role of agriculture preservation. To identify the disease the image of the affected leaf is fed as input into the system. As a first step pre-processing of the image is carried out using median filter.

The filtered image then undergoes segmentation, which is carried out by Guided Active Contour method. Classification of the leaf disease was performed by using Support Vector Machine. They compared the performance of their proposed method with the existing CNN method. With the same set of given images for CNN was 0.7 and 0.8 for SVM. The accuracy in the identification of the disease was 0.6 for CNN and 0.8 for SVM.

In 2020, Shravani V et al [12], proposed a suggestion for crop and soil classification using machine learning approach. The results of such classification can be further combined with crop dataset to predict those crops that are suitable for the soil series of a particular region and its climatic conditions. Soil and crop dataset used by them comprises of chemical attributes and geographical attributes. In the proposed method algorithms like SVM and Ensembling technique were used to classify the soil series data and predict

In 2019, Santosh Mahagaonkar [9] proposed and implemented a system to predict crop yield from previous data. They achieved this by applying machine learning algorithms like Support Vector Machine and Random Forest on agricultural data and recommended fertilizer that is suitable for every particular crop. They focuses on the creation of a prediction model which will be used for future prediction of crop yield. It presented a brief analysis of the crop yield prediction using machine learning techniques. Random Forest was good with accuracy of 86.35% for soil classification when compared to Support Vector Machine. Predictive Support Vector Machine was good with an accuracy of 99.47% for crop yield when compared to Random Forest algorithm. Their work can be extended further to add mobile applications by helping farmers in uploading the image of farms. Also to implement Smart Irrigation System for farms to get higher yield.

In 2017, Rohit Kumar Rajak et al [8], proposed a new method for crop recommendation system to maximize the crop yield using machine learning technique. Their method was characterized by a soil database collected from the farm, crop proposed a new method for crop recommendation system to maximize the crop yield using machine learning technique. Their method was characterized by a soil database collected from the farm, crop. provided by agricultural experts and the achievement of parameters such as soil quality through soil testing lab dataset. The data from the soil testing lab was fed to the recommendation system that will use the collected data and do ensemble model with majority voting technique using support vector machine (SVM) and ANN as the learners to recommend a crop for site specific parameter with high accuracy and efficiency. Their future work was to improve their data set with large number of attributes.

In 2019, Viviliya B et al [15], proposed a hybrid model for recommending crops to south Indian states by considering various attributes. Their recommender model was built as a hybrid model using the classifier algorithm such as Naive Bayes, J48 and association rules. Based on the appropriate parameters, their system will recommend the crop. Their approach considered 15 attributes and 2 algorithms for the recommendation and their system was enhanced with more classifier algorithms. Technology- based crop recommendation system for agriculture helps the farmers to increase the crop yield by recommending a suitable crop for their land with the help of geographic and the climatic parameters. Their proposed hybrid recommender model was found to be effective in recommending a suitable crop.

In 2020, Yoganand S et al [13], addressed the problem of preventing crop disease using a naïve model which was monitored with the help of sensors. Internet of things (IoT) is a promising technology which provides efficient and relevant solutions towards the modernization of agricultural domains. Humidity and Temperature sensor was deployed to verify the humidity and the atmospheric temperature of the plant. Similarly soil moisture sensor was deployed to get status of the soil. Sensors, webcam, GSM and Controllers were used by them for receiving the data from the groundnut farm. The received data was analyzed using machine learning models (XG boost) and so the prediction of crop disease was done. Thus a novel approach for preventing the crop disease (Groundnut Crop) was proposed and their prediction was intimated to farmers through SMS/E-mail.

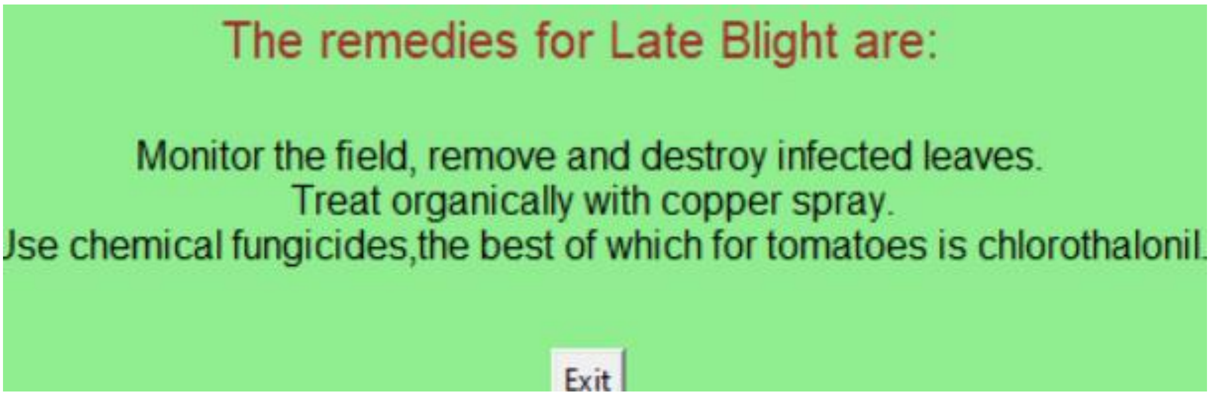
8. TESTING

8.1. Test cases

1.Input**:**

Website:https://github.com/IBM-EPBL/IBM-Project-30313-1660143894/blob/main/Project%20Design%20and%20Planning/Project%20Design%20Phase-II/Functional%20Requirements.pdf

Output:




Dr. Plant
Analyse Image
Status: UNHEALTHY
Disease Name: Late Blight
Click below for remedies...
Remedies

2.Input:

C:\Users\yeswanth\Documents\project plant ibm\test\test

Output:

8.2. User Acceptance Testing

For user acceptance testing**, you** can assign multiple testers to a set of tests. You can even assign the stakeholders who created the business requirements as testers. Select Test plans to see your test suites. You can select Mine to see your favorites or All to see all test plans.

2. Defect Analysis

This report shows 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	8	6	2	3	19
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	7	6	2	20	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	8	1	14
Totals	18	20	17	26	82

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3

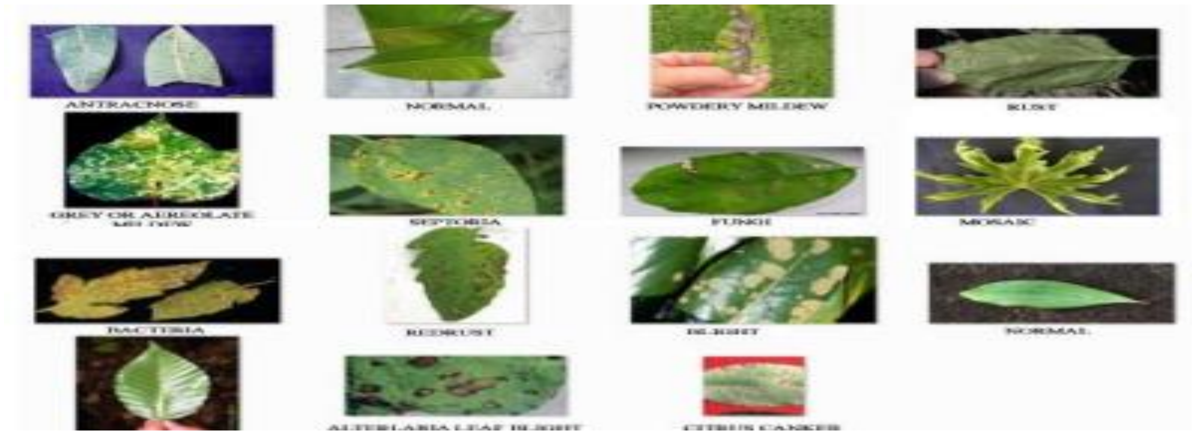
9.RESULTS

9.1. Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	To detect disease and prediction	Attached Below
2.	Accuracy	Training Accuracy Validation Accuracy	Attached Below
3.	Confidence Score (Only Yolo Projects)	Class Detected Confidence Score	Attached below

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].

Recommend the fertilizer for affected leaves based on severity level. Fertilizers may be organic or inorganic. Admin can store the fertilizers based on disease categorization with severity levels. The measurements of fertilizers suggested based on disease severity

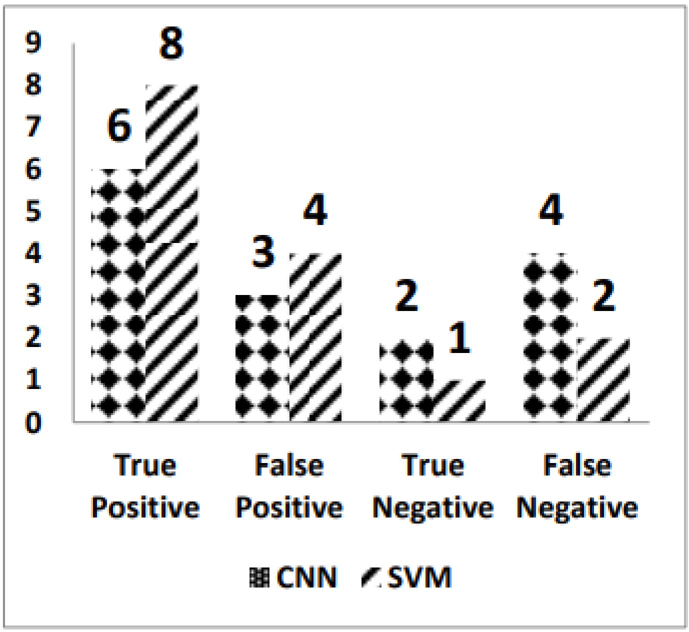


Firstly, some secondary metrics such as true positive (TP), true negative (TN), false positive (FP), and false-negative (FN) [18] are calculated as follows, True Positive: True Positive is an outcome where the model correctly predicts positive class.False Positive: False Positive is an outcome where the model incorrectly predicts positive class.True Negative: True Negative is an outcome where the model correctly predicts negative class.False Negative: False

Negative is an outcome where the model incorrectly predicts negative class.The True Positive, False Positive, True Negative, and False Negative value for captured 15 images are shown in table 1. The pictorial representation of this given below,

TABLE 1
COMPARISON OF CNN AND SVM IN TERMS OF TP, FP, TN, AND FN

Methods	TP	FP	TN	FN
Existing [CNN]	6	3	2	4
Proposed [SVM]	8	4	1	2



3 Performance comparison of CNN and SVM in terms of True Positive, False Positive, True Negative and False Negative.

Precision: The proportion of positive identification is actually correct.

Precision = TP / (TP+FP)

Recall: The proportion of actual positives is identified correctly.

Recall = TP / (TP+FN)

F-Measure: Defined as the weighted harmonic mean of precision

recall. F-Measure = 2TP / (2TP + FP + FN)

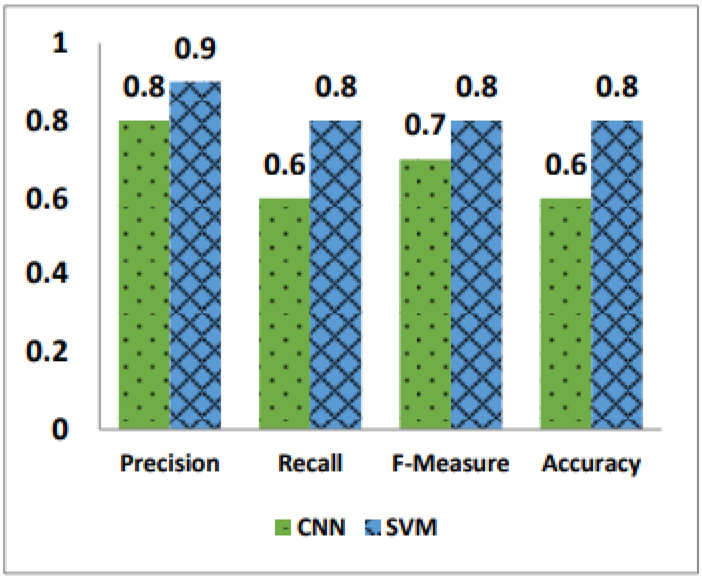
Accuracy: It refers to the closeness of a measured value to a standard or known value.

Accuracy = (TP + TN) / (FP + TP + FN + TN)

The Precision, Recall, F-Measure and Accuracy for the both CNN and SVM are calculated and given in table 2 the corresponding graph is given in Figure 4.

PRECISION, RECALL, F-MEASURE AND ACCURACY VALUES OF CNN AND SVM

Classifiers	Pre	Re	F-M	Acc
CNN	0.8	0.6	0.7	0.6
SVM	0.9	0.8	0.8	0.8



10.Advantages:

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 The result when compared **to existing CNN**.

Disadvantages:

The field of agriculture is in a great threat this includes the diseases that attack leaf. Our system finds the area of leaf that has been affected and also the disease that attacked the leaf. This is achieved by using Image Processing; there are systems that predict the diseases in the leaf. Our system uses K-Medoid clustering and Random Forest algorithm to produce more accuracy in the detection of disease in the leaf.

11.conclusion

The plant serves as the basic need for any living organisms. They are the most important and integral part of our surroundings. Just like a human or other living organism does plant do suffer from different kind of diseases. Such diseases are harmful to plant in a number of ways like can affect the growth of the plant, flowers, fruits, and leaves etc. due to which a plant may even die. So in this work, we have proposed a novel method named as Bacterial foraging optimization based Radial Basis Function Neural Network (BCNN) for identification and classification of plant leaf diseases. The results, when compared with other methods, show that the proposed method achieves higher performance both in terms of identification and classification of plant leaf diseases. Different approaches and models of Deep Learning methods were explored and used in this project so that it can detect and classify plant diseases correctly through image processing of leaves of the plants. The procedure starts from collecting the images used for training, testing and validation to image preprocessing and augmentation and finally comparison of different pretrained models over their accuracy. Finally, at the end , our model detects and distinguishes between a healthy plant and different diseases and provides suitable remedies so as to cure the disease.This paper proposed and developed a system which uses plant leaf images to detect different types of disease in tomato crops, and also provides appropriate fertilizer suggestions.

12. FUTURE SCOPE

In future work, we can extend our approach to improve the accuracy using neural network classification algorithms in order to increase the recognition rate and severity of the detected disease. 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 affected leaves and stems.

13.APPENDIX

13.1. Source Code

Python code

```
import cv2                    # working with, mainly resizing, images
import numpy as np            # dealing with arrays
import os                     # dealing with directories
from random import shuffle # mixing up or currently ordered data that might lead our network
astray in training.
from tqdm import tqdm      # a nice pretty percentage bar for tasks. Thanks to viewer Daniel
BA1/4hler for this suggestion
TRAIN_DIR = 'train/train'
TEST_DIR = 'test/test'
IMG_SIZE = 50
LR = 1e-3
MODEL_NAME = 'healthyvsunhealthy-{}-{}.model'.format(LR, '2conv-basic')

def label_img(img):
    word_label = img[0]

    if word_label == 'h': return [1,0,0,0]

    elif word_label == 'b': return [0,1,0,0]
    elif word_label == 'v': return [0,0,1,0]
    elif word_label == 'l': return [0,0,0,1]

def create_train_data():
    training_data = []
    for img in tqdm(os.listdir(TRAIN_DIR)):
        label = label_img(img)
        path = os.path.join(TRAIN_DIR,img)
        img = cv2.imread(path,cv2.IMREAD_COLOR)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
        training_data.append([np.array(img),np.array(label)])
    shuffle(training_data)
    np.save('train_data.npy', training_data)
    return training_data

def process_test_data():
    testing_data = []
    for img in tqdm(os.listdir(TEST_DIR)):
        path = os.path.join(TEST_DIR,img)
        img_num = img.split('.')[0]
        img = cv2.imread(path,cv2.IMREAD_COLOR)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
        testing_data.append([np.array(img), img_num])

    shuffle(testing_data)
    np.save('test_data.npy', testing_data)
    return testing_data

train_data = create_train_data()
# If you have already created the dataset:
#train_data = np.load('train_data.npy')


import tflearn
```

```python
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers.estimator import regression
import tensorflow as tf
tf.compat.v1.reset_default_graph()

convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 3], name='input')

convnet = conv_2d(convnet, 32, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = conv_2d(convnet, 64, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = conv_2d(convnet, 128, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = conv_2d(convnet, 32, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = conv_2d(convnet, 64, 3, activation='relu')
convnet = max_pool_2d(convnet, 3)

convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

convnet = fully_connected(convnet, 4, activation='softmax')
convnet = regression(convnet, optimizer='adam', learning_rate=LR,
loss='categorical_crossentropy', name='targets')

model = tflearn.DNN(convnet, tensorboard_dir='log')

if os.path.exists('{}.meta'.format(MODEL_NAME)):
    model.load(MODEL_NAME)
    print('model loaded!')

train = train_data[:-500]
test = train_data[-500:]

X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,3)
Y = [i[1] for i in train]

test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,3)
test_y = [i[1] for i in test]

model.fit({'input': X}, {'targets': Y}, n_epoch=8, validation_set=({'input': test_x},
{'targets': test_y}),
    snapshot_step=40, show_metric=True, run_id=MODEL_NAME)

model.save(MODEL_NAME)
```

FRUIT AND VEGETABLE DISEASE PREDICTION:

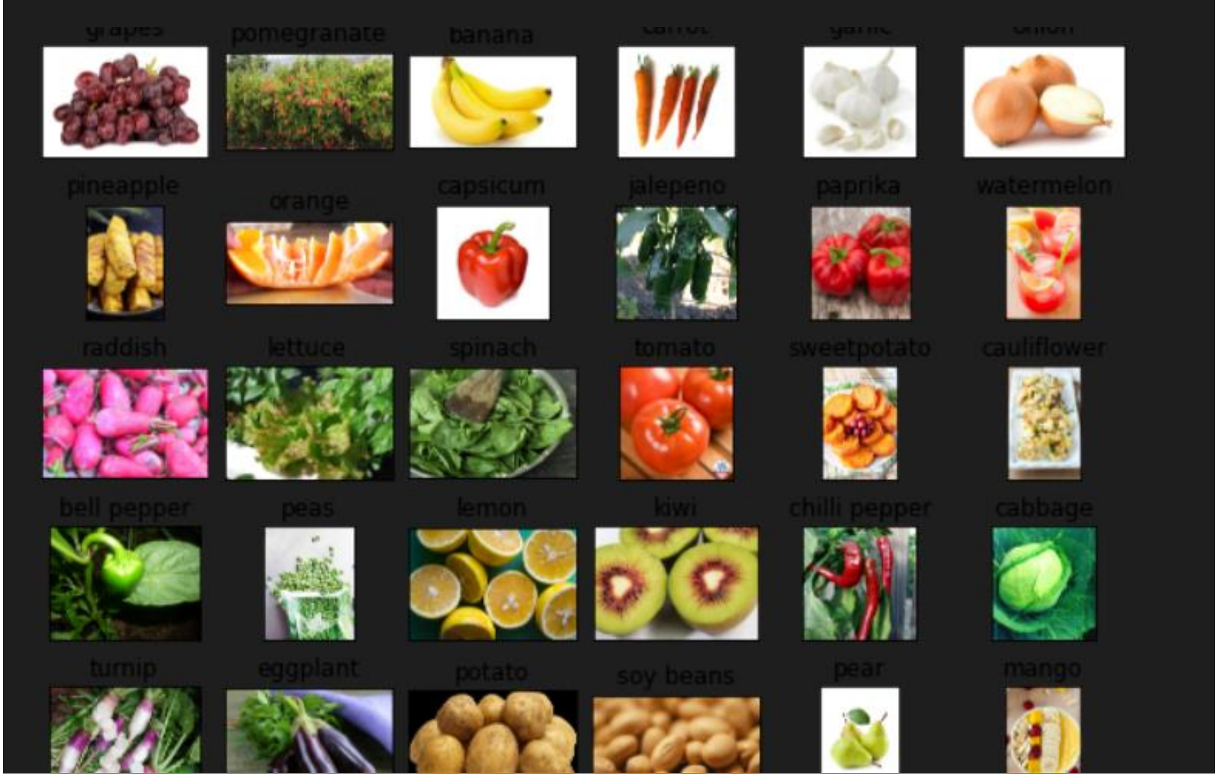
```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under
the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as
output when you create a version using "Save & Run All"
```

```
Output exceeds the size limit. Open the full output data in a text editor
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_4.jpg
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_2.jpg
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_7.jpg
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_5.jpg
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_10.jpg
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_1.jpg
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_8.jpg
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_3.JPG
/kaggle/input/fruit-and-vegetable-image-recognition/validation/capsicum/Image_6.jpg
```

```
import numpy as np
import pandas as pd
from pathlib import Path
import os.path
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img,img_to_array
print(tf.__version__)
# Create a list with the filepaths for training and testing
train_dir = Path('../input/fruit-and-vegetable-image-recognition/train')
train_filepaths = list(train_dir.glob(r'**/*.jpg'))
test_dir = Path('../input/fruit-and-vegetable-image-recognition/test')
test_filepaths = list(test_dir.glob(r'**/*.jpg'))
val_dir = Path('../input/fruit-and-vegetable-image-recognition/validation')
val_filepaths = list(test_dir.glob(r'**/*.jpg'))
def image_processing(filepath):
    """ Create a DataFrame with the filepath and the labels of the pictures
    """

    labels = [str(filepath[i]).split("/")[-2] \
              for i in range(len(filepath))]

    filepath = pd.Series(filepath, name='Filepath').astype(str)
    labels = pd.Series(labels, name='Label')
```

```
    # Concatenate filepaths and labels
    df = pd.concat([filepath, labels], axis=1)

    # Shuffle the DataFrame and reset index
    df = df.sample(frac=1).reset_index(drop = True)

    return df

train_df = image_processing(train_filepaths)
test_df = image_processing(test_filepaths)
val_df = image_processing(val_filepaths)
print('-- Training set --\n')
print(f'Number of pictures: {train_df.shape[0]}\n')
print(f'Number of different labels: {len(train_df.Label.unique())}\n')
print(f'Labels: {train_df.Label.unique()}')







train_generator = tf.keras.preprocessing.image.ImageDataGenerator(
    preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input
)

test_generator = tf.keras.preprocessing.image.ImageDataGenerator(
    preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input
)
train_images = train_generator.flow_from_dataframe(
    dataframe=train_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True,
    seed=0,
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest"
)
val_images = train_generator.flow_from_dataframe(
    dataframe=val_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True,
    seed=0,
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
```

```
    horizontal_flip=True,
    fill_mode="nearest"
)
test_images = test_generator.flow_from_dataframe(
    dataframe=test_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=False

)
```



```
Output exceeds the size limit. Open the full output data in a text editor

User settings:

   KMP_AFFINITY=granularity=fine,verbose,compact,1,0
   KMP_BLOCKTIME=0
   KMP_SETTINGS=1
   KMP_WARNINGS=0

Effective settings:

   KMP_ABORT_DELAY=0
   KMP_ADAPTIVE_LOCK_PROPS='1,1024'
   KMP_ALIGN_ALLOC=64
   KMP_ALL_THREADPRIVATE=128
   KMP_ATOMIC_MODE=2
   KMP_BLOCKTIME=0
   KMP_CPUINFO_FILE: value is not defined
   KMP_DETERMINISTIC_REDUCTION=false
   KMP_DEVICE_THREAD_LIMIT=2147483647
   KMP_DISP_NUM_BUFFERS=7
   KMP_DUPLICATE_LIB_OK=false
   KMP_ENABLE_TASK_THROTTLING=true
   KMP_FORCE_REDUCTION: value is not defined
   KMP_FOREIGN_THREADS_THREADPRIVATE=true
   KMP_FORKJOIN_BARRIER='2,2'
...
   OMP_WAIT_POLICY=PASSIVE
   KMP_AFFINITY='verbose,warnings,respect,granularity=fine,compact,1,0'
```

```python
# Predict the label of the test_images
pred = model.predict(test_images)
pred = np.argmax(pred,axis=1)
# Map the label
labels = (train_images.class_indices)
labels = dict((v,k) for k,v in labels.items())
pred1 = [labels[k] for k in pred]
pred1
def output(location):
    img=load_img(location,target_size=(224,224,3))
    img=img_to_array(img)
    img=img/255
    img=np.expand_dims(img,[0])
    answer=model.predict(img)
    y_class = answer.argmax(axis=-1)
    y = " ".join(str(x) for x in y_class)
    y = int(y)
    res = labels[y]
    return res
```

```
Output exceeds the size limit. Open the full output data in a text editor
['bell pepper',
 'carrot',
 'cauliflower',
 'paprika',
 'mango',
 'jalepeno',
 'pear',
 'beetroot',
 'raddish',
 'sweetpotato',
 'soy beans',
 'chilli pepper',
 'pear',
 'paprika',
 'turnip',
 'spinach',
 'raddish',
 'sweetcorn',
 ' . . . '

def output(location):
    img=load_img(location,target_size=(224,224,3))
    img=img_to_array(img)
    img=img/255
    img=np.expand_dims(img,[0])
    answer=model.predict(img)
    y_class = answer.argmax(axis=-1)
    y = " ".join(str(x) for x in y_class)
    y = int(y)
    res = labels[y]
    return res
img = output('../input/fruit-and-vegetable-image-recognition/test/cabbage/Image_1.jpg')
img
model.save('FV.h5')
```

FEATURE CODE:

```
import tkinter as tk
from tkinter.filedialog import askopenfilename
import shutil
import os
import sys
from PIL import Image, ImageTk

window = tk.Tk()

window.title("Dr. Plant")

window.geometry("500x510")
window.configure(background ="lightgreen")

title = tk.Label(text="Click below to choose picture for testing disease....", background =
"lightgreen", fg="Brown", font=("", 15))
```

```
title.grid()
def bact():
    window.destroy()
    window1 = tk.Tk()

    window1.title("Dr. Plant")

    window1.geometry("500x510")
    window1.configure(background="lightgreen")

    def exit():
        window1.destroy()
    rem = "The remedies for Bacterial Spot are:\n\n "
    remedies = tk.Label(text=rem, background="lightgreen",
                      fg="Brown", font=("", 15))
    remedies.grid(column=0, row=7, padx=10, pady=10)
    rem1 = " Discard or destroy any affected plants. \n  Do not compost them. \n  Rotate yoour tomato plants yearly to prevent re-infection next year. \n Use copper fungicites"
    remedies1 = tk.Label(text=rem1, background="lightgreen",
                        fg="Black", font=("", 12))
    remedies1.grid(column=0, row=8, padx=10, pady=10)

    button = tk.Button(text="Exit", command=exit)
    button.grid(column=0, row=9, padx=20, pady=20)

    window1.mainloop()


def vir():
    window.destroy()
    window1 = tk.Tk()

    window1.title("Dr. Plant")

    window1.geometry("650x510")
    window1.configure(background="lightgreen")

    def exit():
        window1.destroy()
    rem = "The remedies for Yellow leaf curl virus are: "
    remedies = tk.Label(text=rem, background="lightgreen",
                      fg="Brown", font=("", 15))
    remedies.grid(column=0, row=7, padx=10, pady=10)
    rem1 = " Monitor the field, handpick diseased plants and bury them. \n  Use sticky yellow plastic traps. \n  Spray insecticides such as organophosphates, carbametes during the seedliing stage. \n Use copper fungicites"
    remedies1 = tk.Label(text=rem1, background="lightgreen",
                         fg="Black", font=("", 12))
    remedies1.grid(column=0, row=8, padx=10, pady=10)

    button = tk.Button(text="Exit", command=exit)
    button.grid(column=0, row=9, padx=20, pady=20)

    window1.mainloop()

def latebl():
    window.destroy()
    window1 = tk.Tk()

    window1.title("Dr. Plant")
```

```
    window1.geometry("520x510")
    window1.configure(background="lightgreen")

    def exit():
        window1.destroy()
    rem = "The remedies for Late Blight are: "
    remedies = tk.Label(text=rem, background="lightgreen",
                      fg="Brown", font=("", 15))
    remedies.grid(column=0, row=7, padx=10, pady=10)

    rem1 = " Monitor the field, remove and destroy infected leaves. \n  Treat organically with
copper spray. \n  Use chemical fungicides,the best of which for tomatoes is chlorothalonil."
    remedies1 = tk.Label(text=rem1, background="lightgreen",
                         fg="Black", font=("", 12))
    remedies1.grid(column=0, row=8, padx=10, pady=10)

    button = tk.Button(text="Exit", command=exit)
    button.grid(column=0, row=9, padx=20, pady=20)

    window1.mainloop()


def analysis():
    import cv2  # working with, mainly resizing, images
    import numpy as np  # dealing with arrays
    import os  # dealing with directories
    from random import shuffle  # mixing up or currently ordered data that might lead our
network astray in training.
    from tqdm import \
        tqdm  # a nice pretty percentage bar for tasks. Thanks to viewer Daniel BA1/4hler for
this suggestion
    verify_dir = 'testpicture'
    IMG_SIZE = 50
    LR = 1e-3
    MODEL_NAME = 'healthyvsunhealthy-{}-{}.model'.format(LR, '2conv-basic')

    def process_verify_data():
        verifying_data = []
        for img in tqdm(os.listdir(verify_dir)):
            path = os.path.join(verify_dir, img)
            img_num = img.split('.')[0]
            img = cv2.imread(path, cv2.IMREAD_COLOR)
            img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
            verifying_data.append([np.array(img), img_num])
        np.save('verify_data.npy', verifying_data)
        return verifying_data

    verify_data = process_verify_data()
    #verify_data = np.load('verify_data.npy')

    import tflearn
    from tflearn.layers.conv import conv_2d, max_pool_2d
    from tflearn.layers.core import input_data, dropout, fully_connected
    from tflearn.layers.estimator import regression
    import tensorflow as tf
    tf.compat.v1.reset_default_graph()

    convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 3], name='input')

    convnet = conv_2d(convnet, 32, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)
```

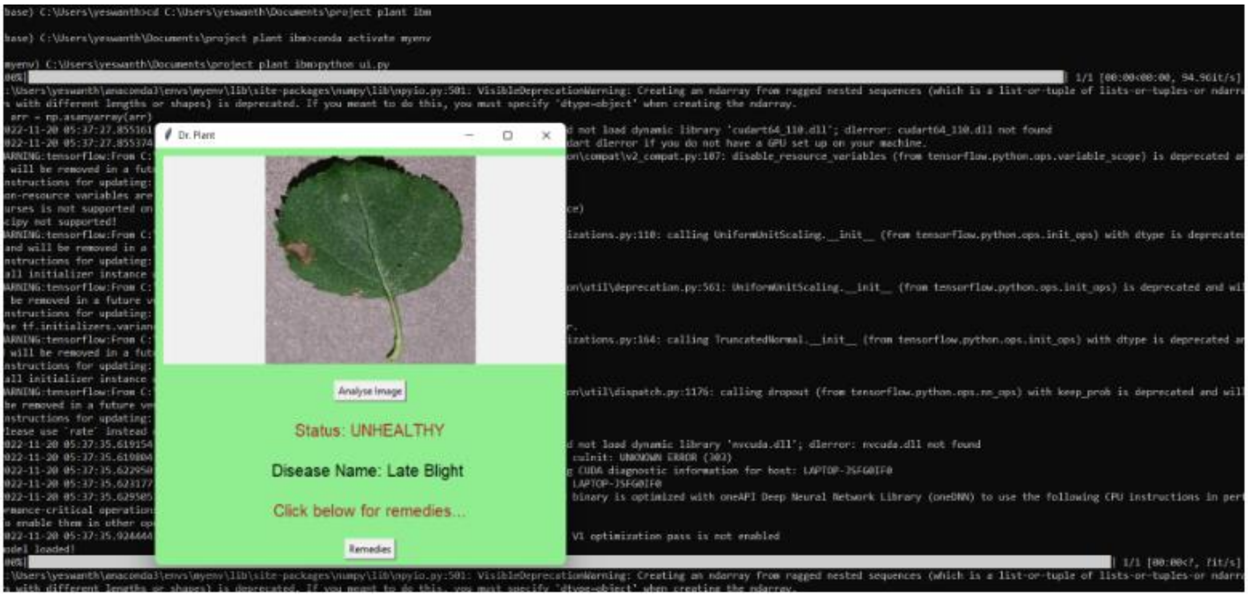
```
    convnet = conv_2d(convnet, 64, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = conv_2d(convnet, 128, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = conv_2d(convnet, 32, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = conv_2d(convnet, 64, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = fully_connected(convnet, 1024, activation='relu')
    convnet = dropout(convnet, 0.8)

    convnet = fully_connected(convnet, 4, activation='softmax')
    convnet = regression(convnet, optimizer='adam', learning_rate=LR,
loss='categorical_crossentropy', name='targets')

    model = tflearn.DNN(convnet, tensorboard_dir='log')

    if os.path.exists('{}.meta'.format(MODEL_NAME)):
        model.load(MODEL_NAME)
        print('model loaded!')

    import matplotlib.pyplot as plt

    fig = plt.figure()

    for num, data in enumerate(verify_data):

        img_num = data[1]
        img_data = data[0]

        y = fig.add_subplot(3, 4, num + 1)
        orig = img_data
        data = img_data.reshape(IMG_SIZE, IMG_SIZE, 3)
        # model_out = model.predict([data])[0]
        model_out = model.predict([data])[0]

        if np.argmax(model_out) == 0:
            str_label = 'healthy'
        elif np.argmax(model_out) == 1:
            str_label = 'bacterial'
        elif np.argmax(model_out) == 2:
            str_label = 'viral'
        elif np.argmax(model_out) == 3:
            str_label = 'lateblight'

        if str_label =='healthy':
            status ="HEALTHY"
        else:
            status = "UNHEALTHY"

        message = tk.Label(text='Status: '+status, background="lightgreen",
                           fg="Brown", font=("", 15))
        message.grid(column=0, row=3, padx=10, pady=10)
        if str_label == 'bacterial':
            diseasename = "Bacterial Spot "
            disease = tk.Label(text='Disease Name: ' + diseasename, background="lightgreen",
```

```
                               fg="Black", font=("", 15))
            disease.grid(column=0, row=4, padx=10, pady=10)
            r = tk.Label(text='Click below for remedies...', background="lightgreen",
fg="Brown", font=("", 15))
            r.grid(column=0, row=5, padx=10, pady=10)
            button3 = tk.Button(text="Remedies", command=bact)
            button3.grid(column=0, row=6, padx=10, pady=10)
        elif str_label == 'viral':
            diseasename = "Yellow leaf curl virus "
            disease = tk.Label(text='Disease Name: ' + diseasename, background="lightgreen",
                               fg="Black", font=("", 15))
            disease.grid(column=0, row=4, padx=10, pady=10)
            r = tk.Label(text='Click below for remedies...', background="lightgreen",
fg="Brown", font=("", 15))
            r.grid(column=0, row=5, padx=10, pady=10)
            button3 = tk.Button(text="Remedies", command=vir)
            button3.grid(column=0, row=6, padx=10, pady=10)
        elif str_label == 'lateblight':
            diseasename = "Late Blight "
            disease = tk.Label(text='Disease Name: ' + diseasename, background="lightgreen",
                               fg="Black", font=("", 15))
            disease.grid(column=0, row=4, padx=10, pady=10)
            r = tk.Label(text='Click below for remedies...', background="lightgreen",
fg="Brown", font=("", 15))
            r.grid(column=0, row=5, padx=10, pady=10)
            button3 = tk.Button(text="Remedies", command=latebl)
            button3.grid(column=0, row=6, padx=10, pady=10)
        else:
            r = tk.Label(text='Plant is healthy', background="lightgreen", fg="Black",
                         font=("", 15))
            r.grid(column=0, row=4, padx=10, pady=10)
            button = tk.Button(text="Exit", command=exit)
            button.grid(column=0, row=9, padx=20, pady=20)

def openphoto():
    dirPath = "testpicture"
    fileList = os.listdir(dirPath)
    for fileName in fileList:
        os.remove(dirPath + "/" + fileName)
    # C:/Users/sagpa/Downloads/images is the location of the image which you want to test.....
you can change it according to the image location you have
    fileName = askopenfilename(initialdir='C:/Users/yeswanth/Downloads/images', title='Select
image for analysis ',
                           filetypes=[('image files', '.jpg')])
    dst = "C:/Users/yeswanth/Documents/project plant ibm/testpicture"
    shutil.copy(fileName, dst)
    load = Image.open(fileName)
    render = ImageTk.PhotoImage(load)
    img = tk.Label(image=render, height="250", width="500")
    img.image = render
    img.place(x=0, y=0)
    img.grid(column=0, row=1, padx=10, pady = 10)
    title.destroy()
    button1.destroy()
    button2 = tk.Button(text="Analyse Image", command=analysis)
    button2.grid(column=0, row=2, padx=10, pady = 10)
button1 = tk.Button(text="Get Photo", command = openphoto)
button1.grid(column=0, row=1, padx=10, pady = 10)
```

```
window.main()
```

INPUT:

```
(base) C:\Users\yeswanth>cd C:\Users\yeswanth\Documents\project plant ibm

(base) C:\Users\yeswanth\Documents\project plant ibm>conda activate myenv

(myenv) C:\Users\yeswanth\Documents\project plant ibm>python ui.py
```



OUTPUT figure

VIDEO LINK:

https://www.kapwing.com/videos/6378f753a2c48e00ee48bfe3