PROJECT BASED LEARNING - II REPORT

ON

Machine Learning algorithm for Plant Disease

REPORT SUBMITTED TOWARDS PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
(IT)

SUBMITTED BY-

Ankita Udgata (19070124007)

Alnoor Lalani (19070124036)

Mrunmayee Patil (20070124501)

Poonam Solanki (20070124502)

Under the Guidance of

Vijayshri Khedkar

SYMBIOSIS INSTITUTE OF TECHNOLOGY
(A CONSTITUTENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)
Pune – 412115

CERTIFICATE

The project titled “**Machine Learning algorithm for Plant Disease”** submitted to the Symbiosis Institute of Technology, Pune for the third-year project in **IT** is based on our original work carried out under the guidance of **Vijayshri Khedkar.**. The report has not been submitted elsewhere for the award of any degree or for any other research related activity.

The material borrowed from other source and incorporated in the report has been duly acknowledged and/or referenced.

We understand that we could be held responsible and accountable for plagiarism, if any, detected later.

Date: 27th April, 2022 **Signature of the candidate**

Research Guide **Head of the Department**

(Vijayshri Khedkar) (Dr. Deepali Vora)

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary project report on “**Machine Learning Algorithm for Plant Disease**”.

I would like to take this opportunity to thank my internal guide, **Vijayshri Khedkar**, for giving us all the help and guidance we needed. I am grateful to them for their kind support. Their valuable suggestions were very helpful.

I am also grateful to Dr. Deepali Vora, Head of Department, SIT, Pune for her indispensable support, suggestions.

In the end our special thanks to Symbiosis Institute of Technology for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for Our Project.

Ankita Udgata (19070124007)
Alnoor Lalani (19070124036)
Mrunmayee Patil(20070124501)
Poonam Solanki(20070124502)

ABSTRACT

Plants play an important part in climate change, agriculture, and the economics of a country. Thereby Plant care is quite important. Plants, like people, have emotions affected by bacteria, fungus, and viruses that cause a variety of diseases It is critical to identify and treat these diseases as soon as possible to keep the entire plant from being destroyed [Objective] This is a paper that proposes the plant disease model, which is based on deep learning detector. The model can detect a variety of plant illnesses.Using images of their leaves as a starting point [Methodology] Plant blight. A neural network is used to create a detection model. First and foremost,To enhance the sample size, augmentation is applied to the dataset. Later, with several convolution and pooling layers, a Convolution Neural Network (CNN) is utilised. The model is trained using the PlantVillage dataset. After the model has been trained, it is thoroughly tested to ensure that the results are accurate. This model has been used in a variety of experiments. PlantVillage data, which includes photographs of healthy and damaged plants, is used for testing purposes 15% of the time. The proposed model has a testing accuracy of 98.3 percent. [Conclusion] The goal of this research is to develop a deep learning model for detecting disease in plant leaves. However, in the future, the model might be combined with a drone or another technology to detect diseases in plants in real time and transmit the location of infected plants to people so that they can be treated appropriately.

Detecting fire does not only means detection of region for fire but it considers several set of factors like increasing nature of fire, which are the components for fire causing, getting minute regions of fire, detecting amount of fire etc. With the use of multi-expert system the detection of fire is done in fast as well as in an accurate manner. So the proposed system provides the fire detection using combination of three fire parameters: color, motion and shape in conjunction with the use of fuzzy logic. Thus the system tries to improve the efficiency of fire detection by using the available resources.

Contribution of this project includes:

- Recognition of fire pixel using Binary Threshold.
- Use of temporal difference technique for identifying motion of fire.
- Detecting shape of fire using morphology of fire to identify the location.
- Implementing multi expert system with the integration of fuzzy logic

Detecting motion of fire plays the very important role in identifying the harmfulness of fire. The increasing nature of fire which has more probability to cause damage can be easily identified by using the temporal difference technique. In addition to this the shape is detected using morphology of fire which gives the exact location of fire so that quick measures are taken to control the fire.

Furthermore with the use of fuzzy logic after extraction of all major parameters makes sure that the false alarm rate has been reduced to the great extent. Thus the proposed system provides the best possible method for detecting the fire with the use of existing sources in a cost effective manner that provides the higher accuracy rate.

.

Table of Contents

List of Figures

CHAPTER 1
INTRODUCTION

1.1 Overview

Modern technology have enabled human society to generate enough food to feed more than 7 billion people. However, food security is still threatened by a variety of factors such as climate change (Tai et al., 2014), pollinator decline (Report of the Plenary of the Intergovernmental Science-Policy Platform on Biodiversity Ecosystem and Services on the work of its fourth session, 2016), plant diseases (Strange and Scott, 2005), and others. Plant diseases not only pose a worldwide danger to food security, but they can also have disastrous effects for smallholder farmers whose livelihoods rely on healthy crops. Smallholder farmers generate more than 80% of agricultural production in the developing countries (UNEP, 2013), and reports of yield losses of more than 50% due to pests and illnesses are typical (Harvey et al., 2014). Furthermore, smallholder farming households account for the majority of hungry people (50 percent) (Sanchez and Swaminathan, 2005), making smallholder farmers particularly sensitive to pathogen-related interruptions in food supply.

Several initiatives have been created to prevent crop loss due to disease. In the last decade, integrated pest management (IPM) approaches have increasingly supplemented historical approaches of extensive pesticide use (Ehler, 2006). Regardless of approach, appropriately diagnosing an illness when it first arises is a critical step in disease management. Historically, agricultural extension agencies or other institutions, such as local plant clinics, have assisted in disease detection. In recent years, such efforts have been bolstered by giving disease diagnosis information online, utilising the world's expanding Internet use. Even more recently, mobile phone-based tools have proliferated, capitalising on the historically unprecedented rapid adoption of mobile phone technology in all areas of the world (ITU, 2015).

Because of their computer power, high-resolution displays, and broad built-in sets of accessories, such as powerful HD cameras, smartphones in particular offer quite unique techniques to assisting in disease identification. It is widely predicted that by 2020, there will be between 5 and 6 billion cellphones on the planet. By the end of 2015, 69 percent of the global population had access to mobile broadband coverage, with mobile broadband penetration reaching 47 percent in 2015, a 12-fold increase since 2007. (ITU, 2015). The combination of broad smartphone use, HD cameras, and high performance processors in mobile devices creates a situation in which disease diagnosis based on automatic picture identification, if theoretically feasible, can be made available on a previously unheard-of scale. We show the technical viability of a deep learning strategy using 54,306 photos of 14 crop species with 26 illnesses (or healthy) made publicly available through the PlantVillage project (Hughes and Salathé, 2015).

1.2 Project Idea

To get to know how plants can have the disease and now with the help of the machine learning algorithm we would identify the disease and try to get to know how the disease came to the plant and try to find the cure by training the data set.

1.3 Motivation

Plants are of various types. The main types of the plants which provide us fruit and vegetables. And there are some natural or man-made conditions where plants are affected by the disease and the fruit and vegetables which are produced by that plant are not of that quality to eat. So, to avoid all circumstances our group has come up with this idea that first we would identify the source of the disease and then we would find that disease cure. Which has motivated us to do this project.

CHAPTER 2
LITERATURE SURVEY

S. Khirade et al. [1] used digital image processing techniques and a back propagation neural network (BPNN) to solve the challenge of plant disease detection in 2015.The authors developed various strategies for detecting plant illness utilising the pictures of leaves They used Otsu's thresholding.

which was followed to segment the contaminated region of the leaf, a border detection and spot detection technique is used. They then retrieved characteristics such as colour, texture, morphology, and edges etc. for plant disease categorization BPNN is used for categorization, that is, to identify the Plant blight.

In their study, Shiroop Madiwalar and Medha Wyawahare investigated several image processing algorithms for plant disease identification. The authors looked at colour and textural traits to detect plant illness. They have tried several things.

Algorithms were applied to a collection of 110 RGB photos. The retrieved characteristics for categorization were the mean and standard deviation of the RGB and YCbCr channels, the grey level cooccurrence matrix (GLCM) characteristics, and the image's mean and standard deviation.

convolved using the Gabor filter For this, a support vector machine classifier was deployed.

The authors came to the conclusion that GCLM characteristics are beneficial in detecting normal leaves. Color characteristics and Gabor filter features, on the other hand, are thought to be the finest for identifying anthracnose-affected leaves and leaf spot They have accomplished greatest accuracy of 83.34 percent when all variables are used.

Peyman Moghadam et al. demonstrated the use of hyperspectral imaging in the identification of plant diseases. This study made use of the visible and near-infrared (VNIR) and short-wave infrared (SWIR) spectrums. The authors employed k-means For leaf segmentation, a spectral clustering technique is used. They've got To remove the grid from hyperspectral pictures, a unique grid removal technique was presented.

The authors attained an accuracy of 83 percent using vegetation indicators in the VNIR spectral range With full spectrum, the range is 93 percent and the accuracy is 93 percent. Despite the fact that the offered strategy succeeded greater precision, a hyperspectral camera having 324 spectral bands is required, thus the The solution becomes too expensive.

Sharath D. M. et al. created a Bacterial Blight detection method for the Pomegranate plant by using characteristics such as colour, mean, homogeneity, SD, variance, correlation, entropy, edges, and so on. For segmenting, the authors used grab cut segmentation. the image's region of interest [4]. To extract the, a Canny edge detector was employed. Borders removed from the images The authors have successfully created a system that can anticipate the degree of infection in the fruit.

To identify plant illness, Garima Shrestha et al. used a convolutional neural network [5]. The authors categorised 12 plant diseases with 88.80 percent accuracy. For this purpose, a collection of 3000 high quality RGB photographs was employed. The network is made up of three blocks of convolution and pooling layers. This increases the network's computing cost. The model's F1 score is also 0.12. which is extremely low due to a larger frequency of erroneous negative predictions.

CHAPTER 3
PROBLEM DEFINITION AND OBJECTIVES

3.1 Problem Definition

In a developing country like India agriculture plays a significant role. Agricultural interventions in India's rural livelihoods account for about 58%. Therefore, the prevention of significant losses and yields of crops is highly dependent on the identification and classification of diseases that the plant may have. Recent and inspiring technologies such as image processing are used to solve such problems using a variety of techniques and algorithms. Initially, the leaves of the plant are affected, when the plant develops a certain type of disease.

3.2 Scope and Objectives

3.2.1 Scope

- Diagnosis of plant diseases can avoid the misconceptions caused by the selection of genetic factors, make the plant genetic factor more ambitious, and improve the efficiency of research and the speed of technological change.
- introduces current trends and challenges of plant leaf disease using in-depth study and advanced imaging techniques

3.2.2 Objectives

In this project, four consecutive stages are used to diagnose the disease. The four categories include pre-processing, leaf separation, feature extraction and classification. To remove the noise we do pre-processing and isolate the affected or damaged area of the leaf, image separation is used. During the fatal phase, the user is recommended treatment. Many living plants are adversely affected by disease.

- Detect the diseases Using camera and IoT without adding extra cost to the existing system
- System can be implement in all the scenarios This review provides the development of in-depth research technology in the field of leaf spot identification in recent years.
- It has become a research hotspot in the field of agricultural plant

protection, such as plant disease recognition and pest range assessment, etc.

3.3 Hardware and Software Requirements

3.3.1 Software Requirement

1) Platform: Python
2) Technology : Latest version of Visual Studio Code and the Python
3) IDE: Python
4) Database: MYSQL 5.5
5) Comm API and JMF (Java Media File) Library

3.3.2 Hardware Requirement

Hardware must be-

- **1 System of following configuration**

1) Processor: Dual Core of 2.2 GHZ
2) Hard Disc: 201 GB
3) RAM : 8GB
4) Monitor, Mouse, Keyboard and Operating Software.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM DESIGN

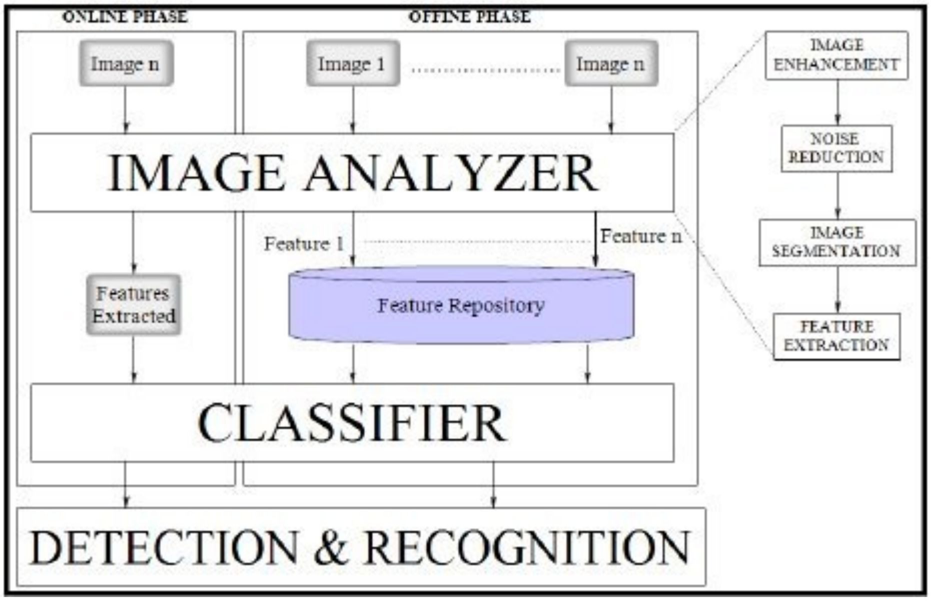

Figure 1 System Design

4.1.1 USE CASE DIAGRAM

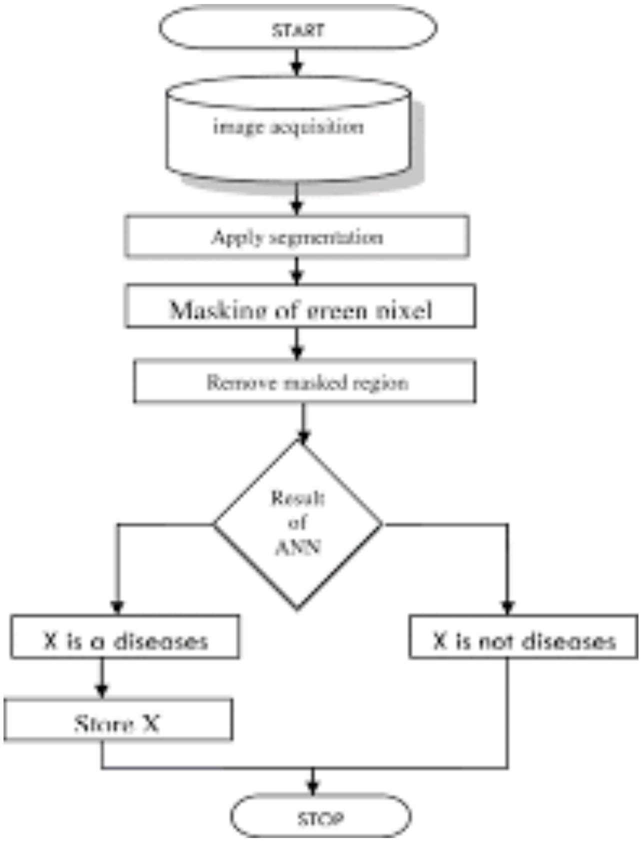

Figure 2 Use Case Diagram

4.1.2 ACTIVITY DIAGRAM

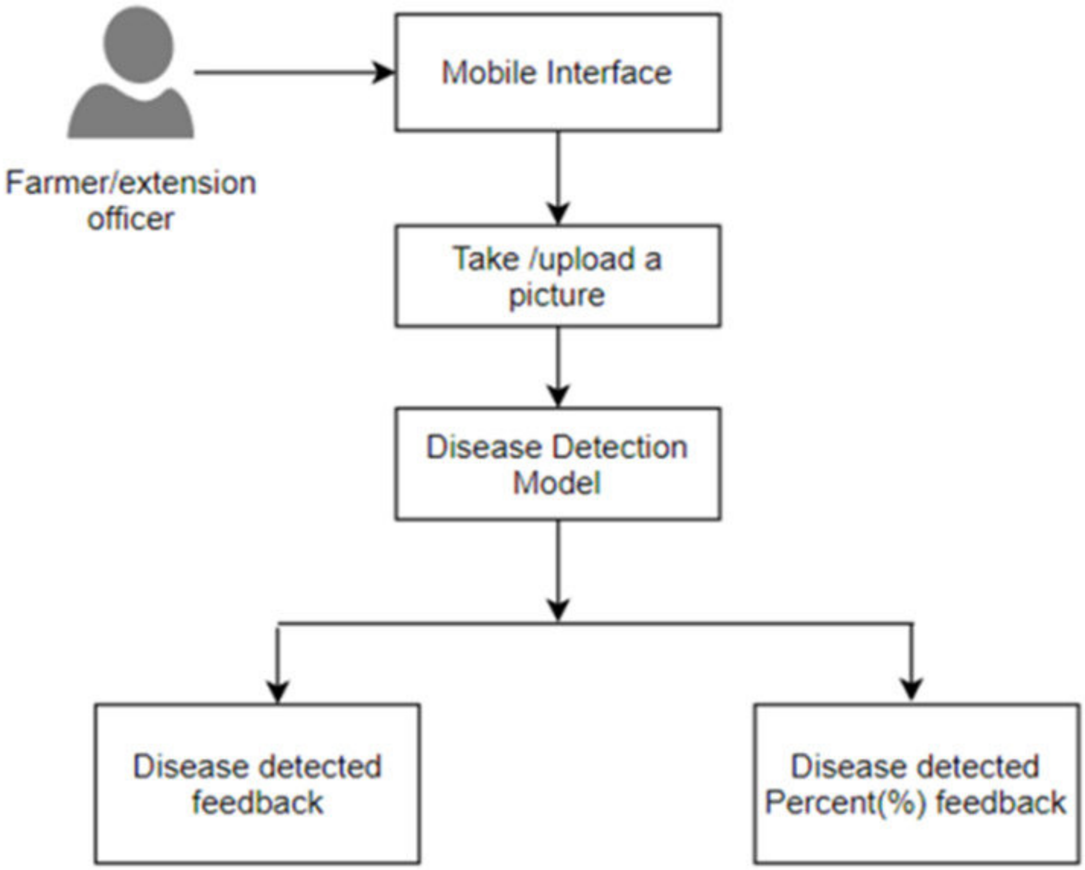

Figure 3 Activity Diagram

4.1.3 Block Diagram

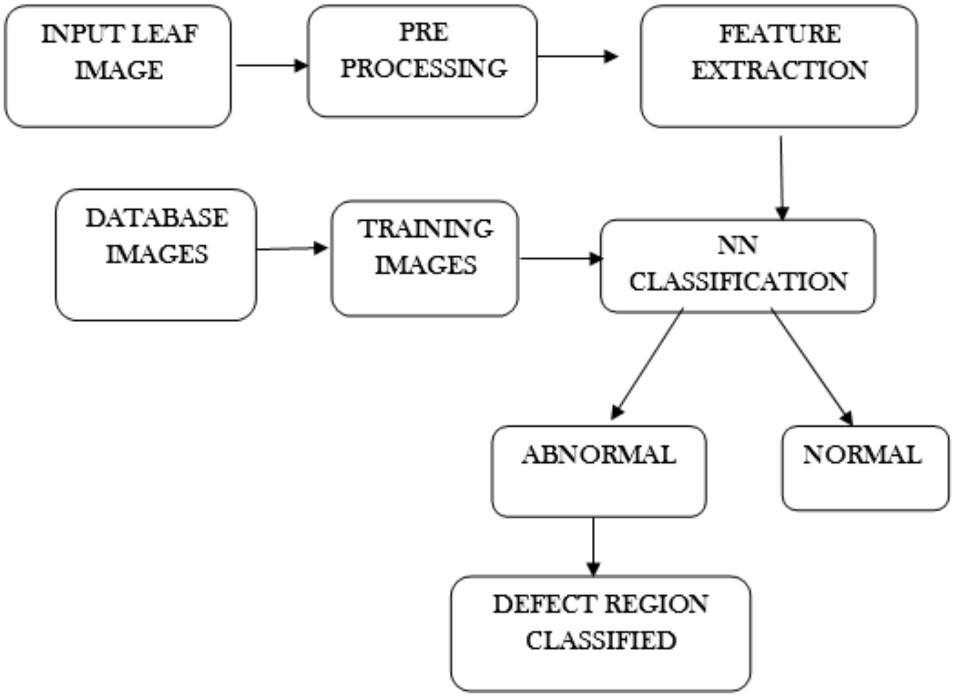

Figure 4 Block Diagram

4.1.4 Deployment Diagram

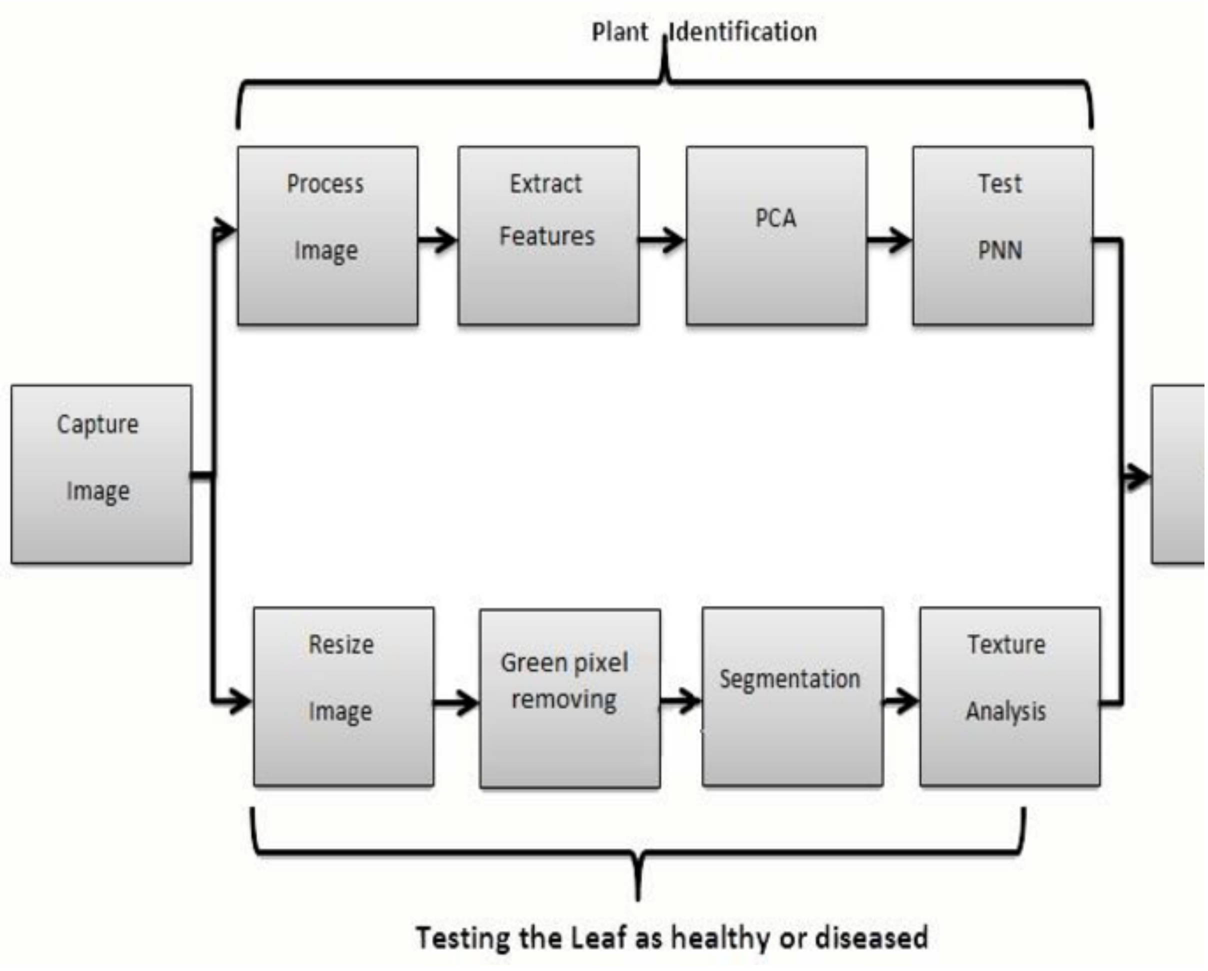

Figure 5 Deployment Diagram

4.1.5 State Transition Diagram

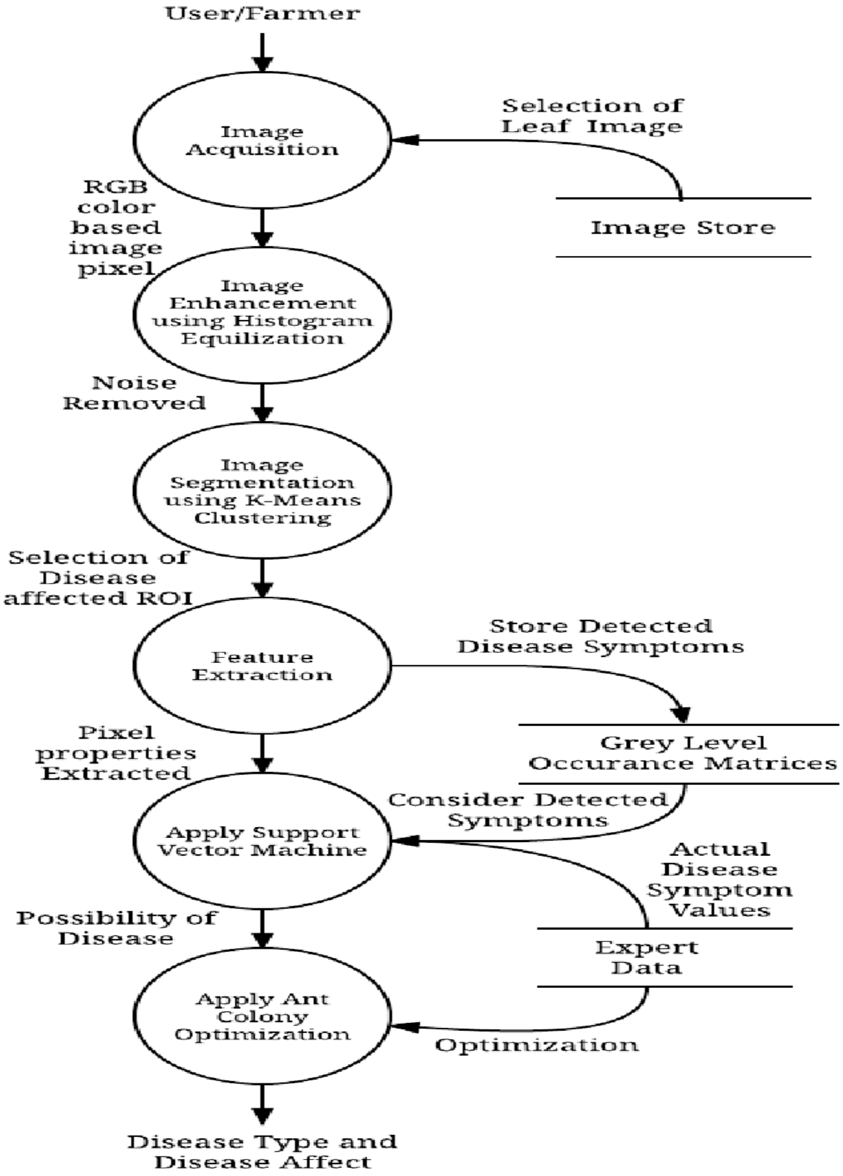
User/Farmer
Image Acquisition
Selection of Leaf Image
Image Store
RGB color based image pixel
Image Enhancement using Histogram Equilization
Noise Removed
Image Segmentation using K-Means Clustering
Selection of Disease affected ROI
Feature Extraction
Store Detected Disease Symptoms
Grey Level Occurance Matrices
Pixel properties Extracted
Consider Detected Symptoms
Apply Support Vector Machine
Actual Disease Symptom Values
Expert Data
Possibility of Disease
Apply Ant Colony Optimization
Optimization
Disease Type and Disease Affect

Figure 6 State Transition

4.1.6 Package Diagram

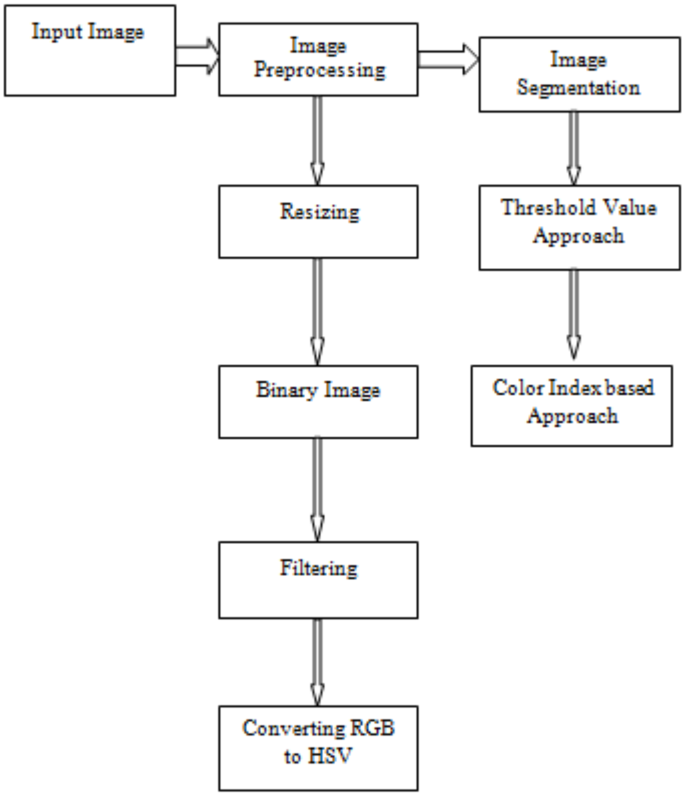

Figure 7 Package Diagram

CHAPTER 5

DATA SET DESCRIPTION

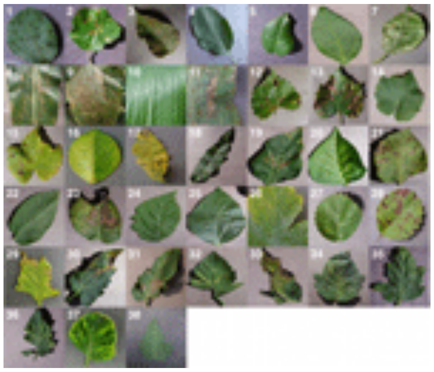

Figure 8

Figure 1 shows an example of leaf photos from the PlantVillage collection, with each crop-disease pair represented. (1)Apple Scab, *Venturia inaequalis* (2) Apple Black Rot, *Botryosphaeria obtusa* (3) Apple Cedar Rust, *Gymnosporangium juniperi-virginianae* (4) Apple healthy (5) Blueberry healthy (6) Cherry healthy (7) Cherry Powdery Mildew, *Podoshaera clandestine* (8) Corn Gray Leaf Spot, *Cercospora zeae-maydis* (9) Corn Common Rust, *Puccinia sorghi* (10) Corn healthy (11)Corn Northern Leaf Blight, *Exserohilum turcicum* (12) Grape Black Rot, *Guignardia bidwellii*, (13) Grape Black Measles (Esca), *Phaeomoniella aleophilum, Phaeomoniella chlamydospora* (14) Grape Healthy (15) Grape Leaf Blight, *Pseudocercospora vitis*(16) Orange Huanglongbing (Citrus Greening), *Candidatus Liberibacter spp*. (17) Peach Bacterial Spot, *Xanthomonas campestris* (18) Peach healthy (19) Bell Pepper Bacterial Spot, *Xanthomonas campestris* (20) Bell Pepper healthy (21) Potato Early Blight, *Alternaria solani* (22) Potato healthy (23) Potato Late Blight, *Phytophthora infestans* (24) Raspberry healthy (25) Soybean healthy (26)Squash Powdery Mildew, *Erysiphe cichoracearum* (27) Strawberry Healthy (28) Strawberry Leaf Scorch, *Diplocarpon earlianum* (29) Tomato Bacterial Spot, *Xanthomonas campestris pv. vesicatoria* (30) Tomato Early Blight, *Alternaria solani* (31) Tomato Late Blight, *Phytophthora infestans* (32) Tomato Leaf Mold, *Passalora fulva* (33) Tomato Septoria Leaf Spot, *Septoria lycopersici* (34) Tomato Two Spotted Spider Mite, *Tetranychus urticae* (35) Tomato Target Spot, *Corynespora cassiicola* (36) Tomato Mosaic Virus (37) Tomato Yellow Leaf Curl Virus (38) Tomato healthy.

We look at 54,306 photos of plant leaves that have been assigned 38 different class designations. Each class label represents a crop-disease pair, and we attempt to predict the crop-disease pair using only the plant leaf image. Each crop-disease pair from the PlantVillage dataset is represented in Figure 1. We resize the photos to 256 256 pixels in all of the methodologies outlined in this research, and we perform model optimization and predictions on these downscaled images.

We use three different versions of the PlantVillage dataset in all of our tests. We begin with the PlantVillage dataset in its original color; then we experiment with a gray-scaled version of the PlantVillage dataset; and finally, we run all of the experiments on a version of the PlantVillage dataset in which the leaves have been segmented, thus removing all of the extra background information that could introduce some inherent bias in the dataset due to the regularized data collection process in the case of the PlantVillage dataset. Segmentation was automated using a script that was adjusted to work well with our dataset.

We settled on a method based on a collection of masks developed by analyzing the color, brightness, and saturation components of various regions of the photos in many color spaces (Lab and HSB). One of the phases

in that processing also enabled us to readily repair color casts, which were particularly strong in certain of the dataset's subgroups, removing still another potential bias.
This collection of tests was created to see if the neural network genuinely learns the "notion" of plant illnesses, or if it is simply learning the dataset's inherent biases. Figure 2 depicts the various versions of the same leaf for a set of leaves chosen at random.

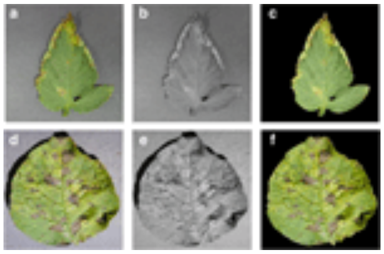

Figure 9

Figure 2 shows photos from three distinct versions of the PlantVillage dataset that were used in different experimental setups. (A) Leaf 1 color, (B) Leaf 1 grayscale, (C) Leaf 1 segmented, (D) Leaf 2 color, (E) Leaf 2 gray-scale, (F) Leaf 2 segmented.

CHAPTER 6
IMPLEMENTATION

5.1 IMPLEMENTATION METHODS

The user takes pictures of the parts of the plant with the lesions such as leaves and fruits and carry plant disease application for recognition. It asks the user about the type of plant / component shown in the selected image in to check the relevant rules for diagnosing the disease.

Additional information provided by the user can help to move forward high precision recognition process. The image processing method produces the following characteristics of the wound: the number of spots, their degree of gray matter and location.

Limitations of regions in this histogram with high pixel concentration as and their tops are used to determine the disease corresponds to the leaf of the image used. Separation means the separation of an image into various parts of the same or similar features. separation can be done using various methods like otsu 'way, k-means to merge, to convert an RGB image into HIS model etc. Based on the shape of the leaf again, we can find disease, We can also note which part is most affected.

6.1 Algorithms

Step 1: Start

Step 2: Get path of an folder of the dataset

Step 3: Print the number of the classes been used

Step 4: Load the image

Step 5: Give path to the classes

Step 6: Resize the image

Step 7: Show the resized image

Step 8: Display the dimensions of the Dataset

Step 9: Creating the training data

Step 10: Give path to the classes

Step 11: Shuffle all the image to avoid the overfit

Step 12: Separate the classes and features

Step 13: Declare the target variables

Step 14: Convert the image into the numpy array to reshape

Step 15: Image Augmentation

Step 16: Normalizing the image

Step 17: Compiling the model

Step 18: Fit the model and save it in the history

Step 19: Summarize the history for the accuracy

Step 20: Summarize the history for the loss

Step 21: Stop

CHAPTER 7
RESULTS AND DISCUSSION

Genetic algorithm prepares both variables correctly, continuously or separately.

•Search in a large sample of cost area.

•A large number of variables can be processed at the same time.

•It can develop flexibility with very complex cost areas.

•It offers many good solutions, not just one solution. So different effects of image classification can be achieved at the same time

In the project Image separation is the process of separating or merging an image into separate parts. There are currently many different ways to create image classification, from the simple cutting method to the advanced color separation methods. These parts are often associated with something that people can easily distinguish and view as individual.

Computers do not have the ability to recognize objects intelligently, and many different ways to distinguish images have been developed. The process of separation is based on the various elements found in the image. This could be color information, borders or part of an image.

The only way to diagnose plant diseases is simply to consult a specialist when diagnosing and diagnosing plant diseases. In doing so, a large team of experts and continuous crop monitoring is required, which is very costly when we do large farms. At the same time, in some countries, farmers do not have the right facilities or even the idea of contacting an expert.

As a result, consultants are very expensive and time consuming. In such cases, the proposed procedure appears to be beneficial in monitoring large plant fields. Automatic detection of symptoms on plant leaves makes it easier and cheaper. This also supports machine vision to provide automatic process control based on image, scan, and direction of the robot.

Therefore, in the agricultural sector, the discovery of plant diseases plays an important role. Early detection of a plant disease, using a spontaneous diagnostic method is beneficial.

CHAPTER 8
CONCLUSION AND FUTURE SCOPE

The successful diagnosis of plant diseases is great is important for crop growth and this can be done by using image processing. The image processing method analyzes the color features of the spots inside parts of plants. The first rating results in recognition of the number of spots and their location in the plant leaves show more than 90% accuracy.

Plant diseases can increase agricultural costs production can also cause economic disaster for the produce if not curable at first. Manufacturers monitor their plants and get the first signs to prevent the spread of plant disease at low cost and should save a large portion of production. Hiring an expert farmers may not be able to afford it especially in remote areas isolated areas. Machine vision provides A different solution to crop monitoring and such a method it may be controlled by an expert to donate his or her own low cost services. Although, there are several

strategies and tests to be done in order confirm a specific disease, but the image processing method can tell us what really happened in the field.

REFERENCES

[1] https://arxiv.org/pdf/2106.10698.pdf

[2] https://www.researchgate.net/publication/347972088_Plant_Leaf_Disease_Classification_and_Detection_System_Using_Machine_Learning

[3] https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full

[4] https://iopscience.iop.org/article/10.1088/1742-6596/1962/1/012024/pdf

[5] https://www.degruyter.com/document/doi/10.1515/comp-2020-0122/html?lang=en

[6] https://towardsdatascience.com/plant-ai-plant-disease-detection-using-convolutional-neural-network-9b58a96f2289

[7] https://www.youtube.com/watch?v=amt9ZmGofJk

[8] https://www.hindawi.com/journals/ddns/2020/2479172/

[9] https://drive.google.com/drive/folders/1o6-HLMqcARb1q5wN28YFVTF8iiFxUqVM?usp=sharing

[10] G.C.A. Barbedo, "Digital image processing techniques for detecting quantifying and classifying plant diseases," Springer Plus, 2:660, Dec. 2013.

[11] J.C. Lai, B. Ming, S.K. Li, K.R. Wang, R.Z. Xie and S.J. Gao, "An Image-Based Diagnostic Expert System for Corn Diseases," Elsevier Agricultural Sciences in China, vol. 9, no. 8, pp. 1221- 1229, Aug. 2010

[12] D.E. Purcell, M.G. O' Shea, R.A. Johnson and S. Kokot, "Near infrared spectroscopy for the prediction of disease rating for Fiji leaf gall in sugarcane clones", Applied Spectroscopy, vol. 63, no. 4, pp. 450-457, Apr 2009

[13] https://www.sciencedirect.com/topics/medicine-and-dentistry/plant-disease

[14] https://www.planetnatural.com/pest-problem-solver/plant-disease

[15] https://ejmcm.com/article_4832_f0fa5d4060331521d4926bde1fae8f50.pdf

[16] https://www.academia.edu/38240782/IRJET-A_Review_of_Different_Classification_Techniques_in_Machine_Learning_using_Weka_for_Plant_Disease_Detection

[17] https://www.researchgate.net/publication/327065422_Plant_Disease_Detection_Using_Machine_Learning

[18] http://ijiet.com/wp-content/uploads/2016/09/80.pdf

[19] https://link.springer.com/article/10.1007/s40747-021-00536-1

[20] https://www.degruyter.com/document/doi/10.1515/comp-2020-0122/html?lang=en

[21] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7903739/

[22] https://ieeexplore.ieee.org/document/9532909

[23] https://www.sciencedirect.com/science/article/pii/S2214785321051403

[24] https://towardsdatascience.com/plant-disease-detection-using-advanced-deep-learning-and-reactjs-6bb913d9e8a8

APPENDIX

TEST CASES

1. INTRODUCTION

The construction of a software system entails a series of production activities in which there are several opportunities to introduce human fallibilities. Error can arise early in the process, when the objectives are incorrectly or incompletely articulated, as well as later in the design and development stages. Software testing is an important part of software quality assurance since it is the final examination of the specification, design, and code generation.

Software must be tested once source code has been generated to find as many problems as possible. Our goal is to create a set of test scenarios with a high probability of detecting errors. This is where software-testing approaches come into play. These methods provide step-by-step instructions for creating tests that:

1. Experiment with software components' inherent logic.
2. Exercise the program's input and output domains to find faults in its function, behaviour, and performance.

1.1 Test Process

The following is the testing procedure to be followed:
The testing procedure outlines the high-level steps that must be completed. Each of these phases has a set of steps that must be completed. The following are the several broad-level phases:

1. Determine the requirements that will be tested. The current Design Specification is used to generate all test cases.
2. Determine the expected outcomes for each test.

3. Determine whatever testing-related equipment and reference documents are needed to complete the testing process. Create a testing environment.
4. Test Design
5. Test Execution.

1.2 Types of Testing

Along with the type of testing, specify the approach to be used for the testing, such as Manual or Automated Testing. For detailed planning of automation tasks, use the Automated Testing Plan.

The different types of testing that may be carried out in the project are as follows:

- Unit testing
- Integration Testing
- System Testing
- Validation Testing
- White Box Testing
- Black Box Testing
- GUI Testing

1.2.1 Unit Testing

To assure their quality, each component is examined separately. The goal is to find design and implementation flaws, such as

- Data structure in component
- Program logic and program structure in a component
- Component interface
- Functions and operations of a component

1.2.2 Integration Testing

To ensure the quality of their integration unit, a set of dependent components is evaluated together. To circumvent the "big-bang" problem, this solution uses incremental integration. This is when the full software is assembled from all units and thoroughly tested. The big-bang method frequently leads to turmoil, which is avoided through progressive integration. Top down and bottom up testing are two approaches to incremental integration testing. The option of regression integration is also available.

When modules are integrated from the top down, they are connected by working their way down the control hierarchy, starting with the main control module. Modules that report to the primary control module are incorporated into the main structure in a depth-first or breadth-first fashion. Early in the test phase, top down integration verifies significant controls or decision points. If there are major control issues, early detection is critical.

Bottom-up integration testing starts with the simplest layers of the programme structure and works its way up. Because modules are connected from the bottom up, processing for modules that are subordinate to a certain level is always available, and test stubs are no longer required.

The focus is to uncover errors in:

- Design and construction of software architecture
- Integrated functions or operations at sub-system level
- Interfaces and interaction and/or environment integration

1.2.3 System Testing

The entire system software is tested. It ensures that all system functions and performance are met in the target environment by ensuring that all elements mesh appropriately. The following are the main areas of focus:

- System functions and performance
- System reliability and recoverability (recovery test)
- System behavior in the special conditions (stress and load test)
- System User operations (acceptance test/alpha test)

- Hardware and software integration collaboration
- Integration of external software and the system

1.2.4 Validation Testing

Validation can be characterized in a variety of ways, but a simple definition is that it succeeds when software performs in a way that the client can reasonably expect.

Validation of software is accomplished by a set of black-box tests that demonstrate compliance with requirements. A test plan specifies the types of tests to be performed, whereas a test process specifies the test cases that will be used to demonstrate compliance with requirements. All functional requirements are met, all behavioral characteristics are met, all performance criteria are met, documentation is right, and human engineered and other requirements are met, according to the plan and method.

1.2.5 White Box Testing

White-box testing allows you to look inside the "box," and it focuses on using the software's own knowledge to influence the selection of test data. Structural, glass-box, and clear-box are all synonyms for white-box.

The cost of white box testing is significantly higher than that of black box testing. It necessitates the production of source code before tests can be scheduled, and it is significantly more time consuming to choose appropriate input data and determine whether or not the software is valid. This testing is solely for the purpose of evaluating the software product; it cannot guarantee that the entire specification has been met.

1.2.6 Black Box Testing

The system is treated as a "black-box" in black-box test design, therefore knowledge of the internal structure is not explicitly used. The term "black-box" test design refers to a method of testing that focuses on functional requirements. Behavioral, functional, opaque-box, and closed-box are all synonyms for black box. Black box testing focuses solely on the specification; it cannot ensure that all aspects of the implementation have been thoroughly evaluated. As a result, black box testing is used.Testing against the standard will reveal omission errors, indicating that a section of the specification has not been met.

1.2.7 GUI Testing

For software engineers, graphical user interfaces (GUIs) bring interesting difficulties. The building of the user interface has become less time consuming and more precise thanks to reusable components given as part of GUI development environments. At the same time, the complexity of GUIs has increased, making test case design and execution more difficult. A set of test cases can be derived because many modern GUIs have the same appearance and feel.

1.3 Feasibility Study

A feasibility study evaluates a system plan for its workability, organizational impact, ability to meet User needs, and efficient use of resources. It focuses on evaluating existing systems and procedures, as well as cost estimates for alternative candidate systems. A feasibility analysis was carried out to see if the system was practical.

Resources and delivery dates are more likely to stymie the development of a computer-based system or a product. Feasibility studies assist analysts in determining whether or not to proceed with a project, alter it, postpone it, or cancel it, which is especially crucial when the project is vast, complex, and costly.

Following the completion of the user demand analysis, the system must verify the software's compatibility and practicality.

1.3.1 Technical Feasibility

The technology employed can be constructed using current equipment and has the technological capacity to keep the data that the new system requires.This technique is in line with current technological trends.Technologies that are more easily accessible and secure.The existing system's technical viability and its ability to accommodate the proposed addition.

1.3.2 Operational Feasibility

Because it is built on Swings coding, the proposed system is simple to build. The MySQL server was used to generate the database, which is more secure and easier to manage. The resources needed to implement/install them are already in place. The organization's personnel have already had enough computer exposure. As a result, the project is possible on a practical level.

1.3.3 Economical Feasibility

The most common method for evaluating the effectiveness of a new system is economic analysis. The method, also known as cost/benefit analysis, involves determining the projected advantages and savings from a proposed system and comparing them to the expenses.

If the advantages outweigh the costs, the decision to develop and deploy the system is made. Before taking action, an entrepreneur must carefully balance the costs and advantages. This technique, which assesses brain capacity with a quick and online exam, is more cost-effective. As a result, it is a financially sound project.

1.4 TEST CASES:

Sr. No.	Test Cases	Actual Accuracy	Expected Accuracy	Result
1	Apple Scab	0.9600	greater than 0.75 and less than 1	Pass
2	Apple Black rot	0.9954	greater than 0.75 and less than 1	Pass
3	Apple Cedar apple rust	0.9783	greater than 0.75 and less than 1	Pass
4	Apple healthy	0.9937	greater than 0.75 and less than 1	Pass
5	Blueberry healthy	0.9998	greater than 0.75 and less than 1	Pass
6	Cherry healthy	0.8408	greater than 0.75 and less than 1	Pass
7	Cherry Powdery mildew	0.9481	greater than 0.75 and less than 1	Pass
8	Corn Cercospora leaf spot Gray leaf spot	0.8780	greater than 0.75 and less than 1	Pass
9	Corn Common rust	0.8990	greater than 0.75 and less than 1	Pass
10	Corn healthy	0.8987	greater than 0.75 and less than 1	Pass
11	Corn Northern Leaf Blight	0.8912	greater than 0.75 and less than 1	Pass
12	Raspberry healthy	0.9998	greater than 0.75 and less than 1	Pass
13	Squash Powdery Mildew	0.9995	greater than 0.75 and less than 1	Pass

14	Strawberry Leaf Scorch	0.9972	greater than 0.75 and less than 1	Pass
15	Strawberry Healthy	0.9993	greater than 0.75 and less than 1	Pass

Table: Accuracy