

CHAPTER – 1

INTRODUCTION

As we know Nepal is an Agricultural country. In countries like Nepal, it is of utmost importance to bring technological advancement in the fields related to crop productivity. Modern technologies have given human society the ability to produce enough food to meet the demand of more than 3 crore people. However, food security remains threatened by several factors including climate change, the decline in pollinators, plant diseases, and others. Identifying diseases from images of plant leaves is one of the most important research areas in precision agriculture. Advances in artificial intelligence, image processing, and graphical processing units (GPUs) can expand and improve the practice of precise plant protection and growth.

1.2 Problem Statement

Many people and technological groups are involved in the field of agriculture to increase the yield and throughput. There have been various techniques used in the past to solve problems related to disease spread in a variety of plants. With the advancement in technology, plant disease detection has become easier and more precise. In our system, a different approach, i.e., CNN algorithm is used for the same. Various kinds of methods have been used recently to determine the type of plant disease. Some of these involve analysis and study of the chemical analysis method to determine plant diseases, and ways which are indirect by implementing physical techniques, like spectroscopy of the leaf and imaging, to get information related to properties of a plant. Following this, the merits of the project contrasted with the existing technologies are related to the underlying points: The system avoids the process involved in gathering inputs for studying them in the laboratory, because of pre-existing images taken in place of the plant diseases. It examines the chances where a particular plant is concurrently simulated with higher than one pest or disease in the unchanged recorded input. The outlook deploys inputting of various images apprehended by various cameras with diverse resolutions, like mobile phones and the other available cameras devices. The project is systematically packed with different conditions related to illuminations, the size of actors in an image, surrounding distinction, etc., holding across the neighboring part of that particular plant. It imparts a

feasible functioning approach that can maneuver in the domain by not using costly and complex and compound technologies.

1.3 Objectives

- To enhance the given input image by Image acquisition and Image preprocessing.
- To identify the affected part through texture analysis and Segmentation.
- To classify the healthy and affected leaf part by feature extraction and classification.
- To train the model by using testing data for accurate results.

1.4 Scope and Limitation

1.4.1 Scope

The image processing could be used in the field of agriculture for several applications. It includes detection of diseased leaf, stem or fruit, to measure the affected area by disease, to determine the color of the affected area. Plant cultivating is one of the most remunerative farming enterprises in Nepal. The naked eye observation by the experts is an approach usually taken in identification and detection of plants. This approach is time consuming in huge farms or land areas. The use of image processing techniques in detection and identification of various plant diseases in the earlier stages and thereby the quality of the product could be increased. These systems monitor the plant such as leaves and stem and any variation observed from its characteristic features, variation will be automatically identified and also will be informed to the user.

1.4.2 Limitations

- The user has to provide the dataset by taking a large number of images of leaves.
- It takes much time to train the data for a better result.

1.5 Development Methodology

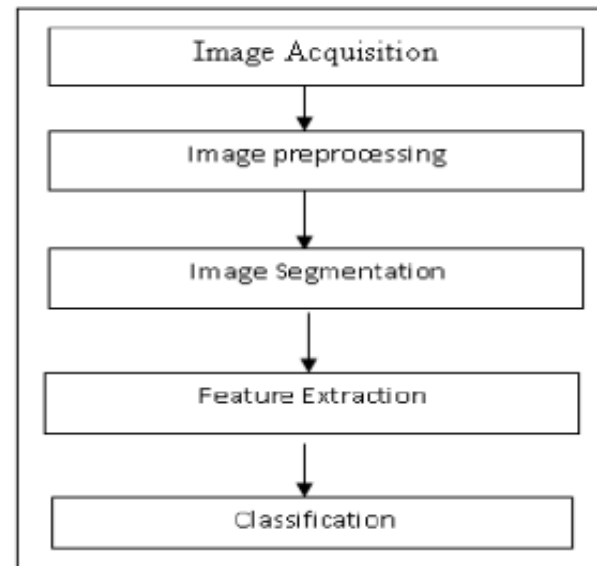


Figure 1: Development Methodology

1.5.1 Image Acquisition

Initially, the digital images are acquired from the circumstances using a digital mobile camera or digital camera and given as input to the identification system. This is the image in which the leaf disease has to be identified by the system.

1.5.2 Image Pre-processing:

Image pre-processing tasks are the initial stage before feature extraction. There are three steps of image preprocessing, i.e., image cropping, image converting and image enhancement. The image is cropped on leaf disease areas, and then converted to gray levels. In this process, the noise is eliminated from the captured image to enhance the image quality.

1.5.3 Image Segmentation

Image segmentation is the methodology of apportioning a digital image into different fragments (sets of pixels, otherwise called super pixels). The objective of segmentation is to improve and/or change the representation of an image into something that is more significant and less demanding to examine. The consequence of image segmentation is a

set of sections that aggregate cover the whole image or a set of forms removed from the image. Each of the pixels in a locale is comparative as for some trademark or registered property, for example, shading, force, or surface. Nearby districts are essentially diverse as for the same characteristic(s). At the point when connected to a pile of images, normal in restorative imaging, the subsequent forms after image segmentation can be utilized to make 3D recreations with the assistance of interpolation calculations like marching cubes.

1.5.4 Feature Extraction

Feature extraction is a special form of dimensionality reduction. At the point when the information to a calculation is complex, it would be impossible to be transformed and it is suspected to be exceptionally repetitive, then the data information will be changed into a decreased representation set of features (likewise named features vector). Changing the information into the set of features is called features extraction.

1.5.5 Classification

Image classification, a theme of example distinguishable in PC vision, is a methodology of arrangement in view of relevant data in images. "Logical" implies this methodology is concentrating on the relationship of the adjacent pixels, which is additionally called a neighborhood. The objective of this methodology is to arrange the images by utilizing the logical data. According to the classification result the disease of the leaf is identified and the appropriate action can be taken by the farmers in the initial stage of disease for its control.

CHAPTER – 2

BACKGROUND STUDY AND LITERATURE VIEW

2.1 Background Study

Plant diseases and pests' detection is a very important research content in the field of machine vision. It is a technology that uses machine vision equipment to acquire images to judge whether there are diseases and pests in the collected plant images. At present, machine vision-based plant diseases and pests' detection equipment has been initially applied in agriculture and has replaced the traditional naked eye identification to some extent.

For traditional machine vision-based plant diseases and pests' detection methods, conventional image processing algorithms or manual design of features plus classifiers are often used. This kind of method usually makes use of the different properties of plant diseases and pests to design the imaging scheme and chooses appropriate light sources and shooting angles, which is helpful to obtain images with uniform illumination. Although carefully constructed imaging schemes can greatly reduce the difficulty of classical algorithm design, but also increase the application cost. At the same time, under the natural environment, it is often unrealistic to expect the classical algorithms designed to eliminate the impact of scene changes on the recognition results. In a real complex natural environment, plant diseases and pests' detection is faced with many challenges, such as the small difference between the lesion area and the background, low contrast, large variations in the scale of the lesion area and various types, and a lot of noise in the lesion image. Also, there are a lot of disturbances when collecting plant diseases and pests' images under natural light conditions. At this time, the traditional classical methods often appear helpless, and it is difficult to achieve better detection results (Zelner & Fergus, 2014).

In recent years, with the successful application of deep learning models represented by a convolutional neural network (CNN) in many fields of computer vision (CV, computer-vision), for example, traffic detection, medical Image Recognition, Scenario text detection, expression recognition, face recognition, etc. Several plant diseases and pests'

detection methods based on deep learning are applied in real agricultural practice, and some domestic and foreign companies have developed a variety of deep learning-based plant diseases and pests' detection. WeChat applet and photo recognition APP software. Therefore, plant diseases and pests' detection methods based on deep learning not only have important academic research value but also have a very broad market application prospect (G., Y., & J., 2017).

Because of the lack of comprehensive and detailed discussion on plant diseases and pests' detection methods based on deep learning, this study summarizes and combs the relevant literature from 2014 to 2022, aiming to help researchers quickly and systematically understand the relevant methods and technologies in this field. The content of this study is arranged as follows: “Definition of plant diseases and pests detection problem” section defines plant diseases and pests detection problem; “Image recognition technology based on deep learning” section focuses on the detailed introduction of image recognition technology based on deep learning; “Plant diseases and pests detection methods based on deep learning” section analyses the three kinds of plant diseases and pests detection methods based on deep learning according to network structure, including classification, detection and segmentation network; “Dataset and performance comparison” section introduces some datasets of plant diseases and pests detection and compares the performance of the existing studies; “Challenges” section puts forward the challenges of plant diseases and pests detection based on deep learning; “Conclusions and future directions” section prospects the possible research focus and development direction in the future (A. & F., 2018).

2.2 Literature Review

Fungi usually cause diseases that affect the plants, and they typically attack the leaves. Viral and bacterial pathogens cause many others. Precision in agriculture has improved with the increased use of ML and its related features. The reduced production quantity in agriculture hurts many people and animals, which requires modern technology to solve. The extraction and detection of diseases are easier when the image-based detection system is used because of its high accuracy and reduced complications and duplication of data. In some plants like tomatoes, the use of the images to determine the diseases that affect them and the extent of the damages cannot be achieved unless there is a high accuracy rate. The survey on plant diseases shows that many diverse factors determine

how technology-based image detection is applied. In other words, the diseases that cause visible dents and changes on the plants are the ones that can be detected using this technology as opposed to the ones that cause damages that cannot be detected from the plants' images . The analysis in this research shows that plant diseases are usually detected when they start showing an impact on the physical appearance of the plants.

The main challenge affecting the field of agriculture is the reduction in production and poor-quality production in plants. The challenge is a result of the poor detection and management of the diseases that affect the plants. The challenge is also extended to affect human beings in several ways. The reduced plant cover due to plant diseases means that global warming, famine, and reduced air purification ensue. Hyperspectral imaging has become a reliable way of detecting crop diseases on time. It is hard to determine the factors that lead to the diseases unless they are detected on time. In other words, if a disease is detected on time, it is easy to relate it to the possible factors that lead to its occurrence. For example, scientists could determine if there was a change in weather or climate that could have led to the occurrence of the disease.

Further research by shows an inadequate database that could be used to provide background knowledge for comparing the images taken. The other challenge is that the symptoms and characteristics of the diseases are diverse and could be similar to a certain degree. For example, many diseases could lead to the wilting of leaves. The challenge is yet to be resolved because more and new images are uploaded progressively by experts.

The other challenge is the lack of suitable instruments for use in the work of image detection. Most of the experts in the field do not have the equipment they require to analyze the images they get from the field, and this makes it hard for them to acquire accurate data and identify the diseases. The other one is that there is a low rate of implementation in some areas due to the regulations put in place to ensure the credibility and reliability of the data from these analyses. For example, after the 4th and 6th International Conference on Machine Learning and Soft Computing, there have been many regulations that may derail the use of ML in some parts. The rules discourage some of the results from the ML functions from being applied in practice because they do not meet the required parameters.

CHAPTER – 3

SYSTEM ANALYSIS

The direct result of requirements analysis is requirements specification. Hardware requirements specifications list the necessary hardware for the proper functioning of the project. Software requirements specifications is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of use cases that describe interactions the users will have the software. In software engineering, a functional requirement defines the function of a system and its components. A function is described as a set of inputs, the behavior, and outputs. A non-functional requirement that specifies the criteria that can be used to judge the operation of a system, rather than specific behavior.

3.1. Requirement Analysis

A Software Requirements means a requirement specification for a software system is a complete description of a behavior of the system to be developed. In addition to a description of a software functions, the SRS also contains non-functional requirements. Software requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification and validation of requirements for software.

3.1.1 Software Requirements

- Develop Platform : Microsoft Windows 10,
- Development tools and environment: Python, OpenCV, TensorFlow Lite.
- Software Application on Windows System : Windows Google Chrome , Jupyter Notebook, Visual Studio Code
- Hardware Support: Camera Phone, PC,
- Application Development : Python

3.1.2 Hardware Requirements

- **RAM** : 2 GB or above
- **Processor** : Intel dual core or above
- **Processor Speed** : 1.0 GHZ or above
- **Hard disk** : 20 GB hard disk or above

3.1.3 Functional Requirements

Functional requirements describe the system functionality, while the non-functional requirements describe system properties and constraints. Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks, or the functions the system is required to perform. This lays out important concepts and discusses capturing functional requirements in such a way they can drive architectural decisions and be used to validate the architecture.

3.1.4 Non-Functional Requirement

Non-functional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's quality attributes. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute the project.

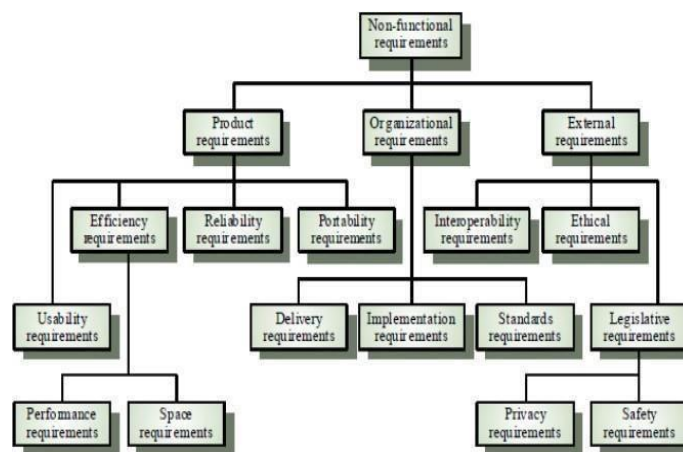


Figure 2: Non-Functional Requirements

- **Reliability:**

The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a customer has revealed a couple of enhancements. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.

- **Maintainability**

There should not be an excess of occupations running on diverse machines such that it gets hard to screen whether the The system watching and upkeep should be fundamental and focus in its approach. employments are running without lapses.

- **Performance**

The framework will be utilized by numerous representatives all the while. Since the system will be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every piece of its customers.

- **Portability**

The framework should to be effectively versatile to another framework. This is obliged when the web server, which s facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another framework.

3.2 Feasibility Analysis

As Nepal is an agricultural country, it is necessary to study feasibility analysis whether it is feasible in the farmer world or not . So it is very essential to analysis it on the real world.

3.2.1 Technical Feasibility

The system is technically feasible to implement. The technology used guarantees the reliability, accuracy, and security. The technical aspects of the system can easily cover cross-platforms like mobile, desktop, web etc. and is easily scalable. The technical manpower is also easily available.

3.2.2 Operational Feasibility

From the user's perspective the system lays out the most basic features needed by the user that is choosing if they require the output in audio or text which makes the system operationally feasible. The proposed system is operationally feasible since the system is used efficiently when it is developed. There will be very low resistance from the users since the application is user friendly that affects the possible application benefits. The project can be supported by management. This is the new system based on machine learning that fascinates the users who are overwhelmed by the old and tedious system.

3.2.3 Economical Feasibility

All the tools required for the development are freeware tools. The data needed for the application is free of cost. This project provides the user with the benefit of acknowledging the public opinion and insight before launching any business or project. The development costs, annual operating costs, annual benefits and intangible costs and benefits make the system affordable and economically feasible. On behalf of cost- benefit analysis, the proposed system is feasible and is economical regarding its pre-assumed cost for making a system.

3.3.4 Schedule Feasibility

Activities	Week 01	Week 02	Week 03	Week 04	Week 05	Week 06	Week 07	Week 08	Week 09	Week 10	Week 11	Week 12
Planning												
Analysis												
Design												
Implementation												
Testing												
Documentation												
Review												
Presentation												*

Figure 3.2: Gantt chart showing the project timeline. This project is design with its structure in planning, analysis, designing, implementation, testing, documentation ,and review all counted to complete within a time span of 12 weeks (D.B.victory, 2020)

The Figure above shows the schedule of the project with the major phases. Initially, the requirements for the systems were gathered and analyzed properly. Later the system

designing, user interfaces designing, and database designing was done. The coding phase was the longest phase of the project life cycle.

3.3 Analysis

3.3.1 Activity Diagram

The above flow represents the flow from one activity to another activity, the activity starts from input leaf image through digital camera, and then input leaf is preprocessed and extracted from the features like color, shape, texture and so on. Now, the processed image is classified as Normal or Abnormal, if Abnormal is found in the leaf, then remedies will be suggested.

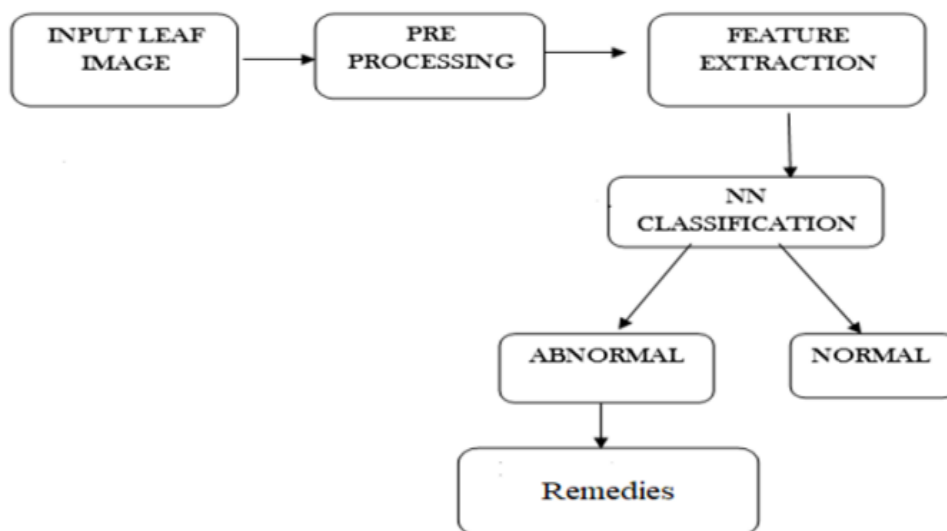


Figure 3: Activity Diagram

The above flow represents the flow from one activity to another activity, the activity starts from input leaf image through digital camera, and then input leaf is preprocessed and extracted from the features like color, shape, texture and so on. Now, the processed image is classified as Normal or Abnormal, if Abnormal is found in the leaf, then remedies will be suggested.

3.3.2 DataFlow diagram

A data flow Diagram is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary

step to create an overview of the system without going into great detail, which can later be elaborated.

i. Context Level DFD/ Level 0 DFD

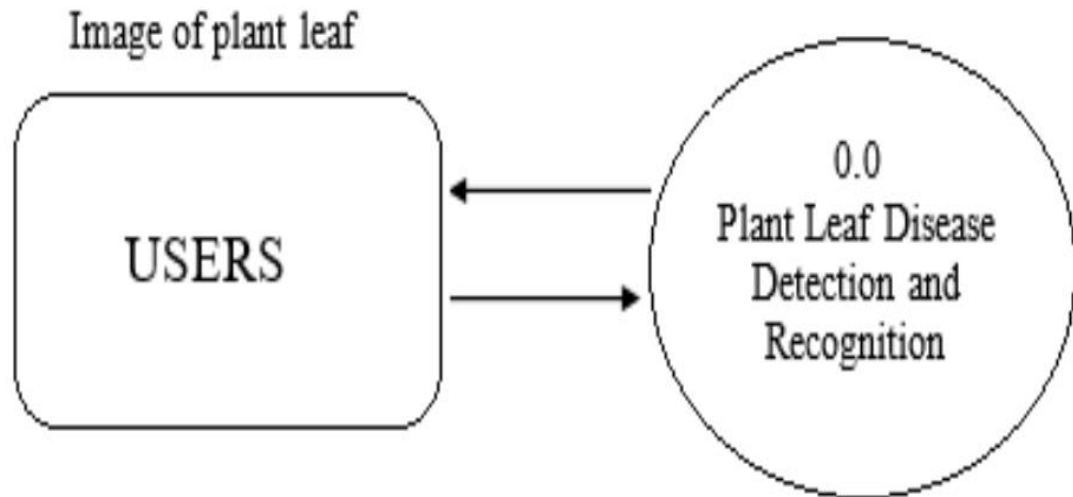


Figure 4: Level 0 DFD

ii. Level 1 DFD

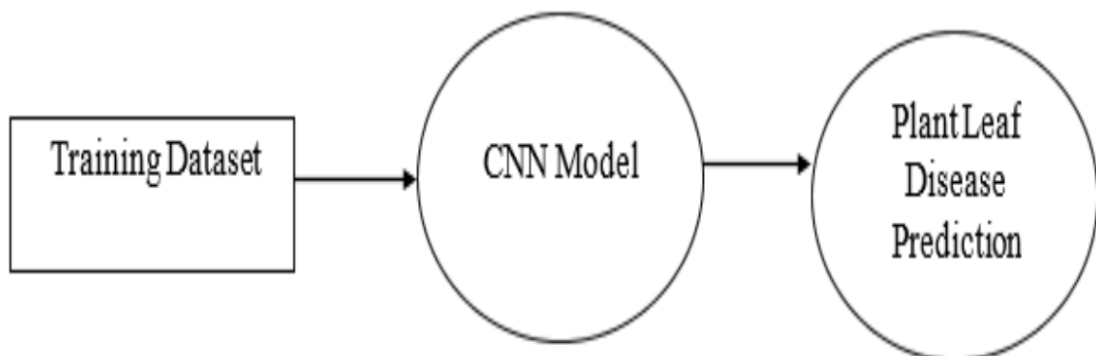


Figure 5: Level 1 DFD

3.3.3 Sequence Diagram

Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows as parallel vertical lines (lifelines), different processes or objects that live simultaneously and as horizontal arrows, the messages exchanged between them in the order in which they occur.

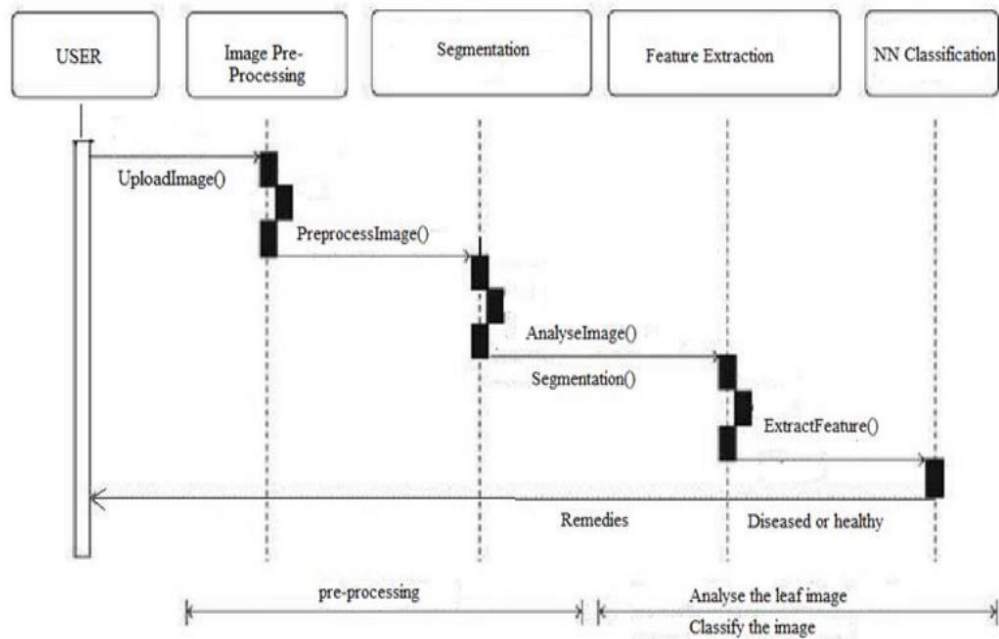


Figure 6: Sequence Diagram

CHAPTER – 4

SYSTEM DESIGN

4.1 Use Case Diagram

Use case diagram is a graphic depiction of the interactions among the elements of a system. Use cases will specify the expected behavior, and the exact method of making it happen. Use cases once specified can be denoted both textual and visual representation.

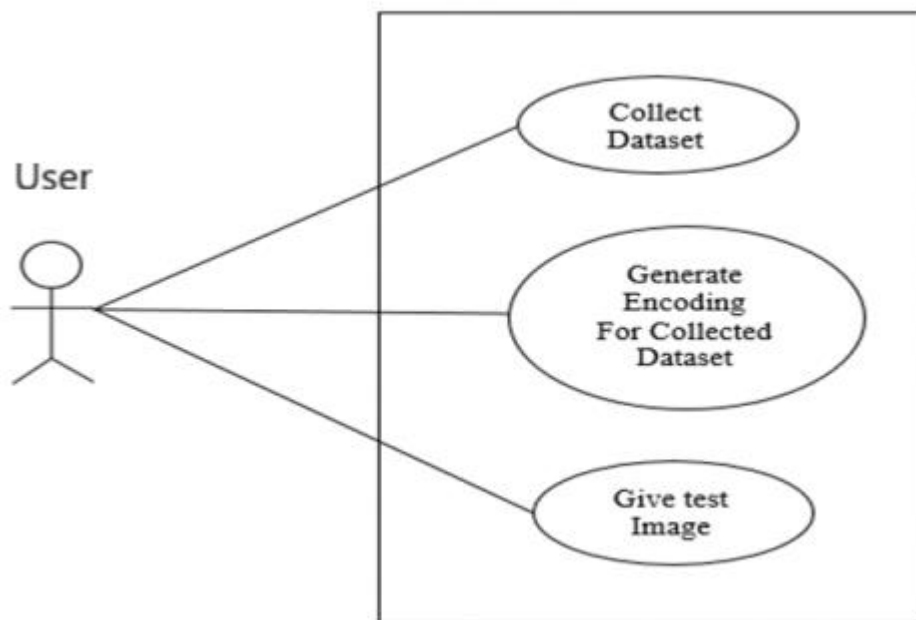


Figure 7: Use Case Diagram

Use case diagrams are used to specify:

- Requirements (external), required usages of a system under design or analysis - to capture what the system is supposed to do.
- The functionality offered by a subject – what the system can do.
- Requirements the specified subject poses on its environment - by defining how the environment should interact with the subject so that it will be able to perform its services.

4.2 System Architecture

The proposed System architecture comprises data acquisition from a huge dataset, processing at different convolutional layers and then the classification of plant diseases which declares if the plant image is of a healthy class or diseased class.

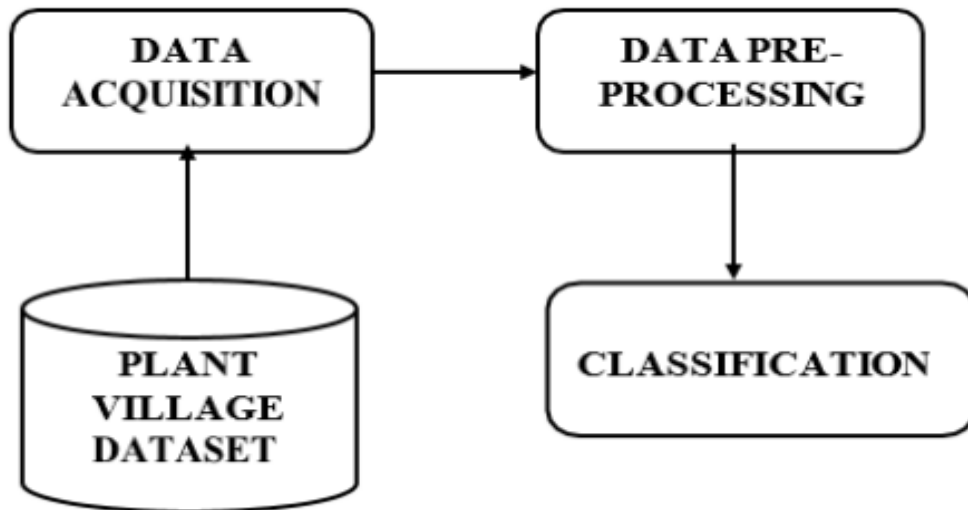


Figure 8: System Architecture

4.3 Algorithm

CNN deals with the basic four steps:

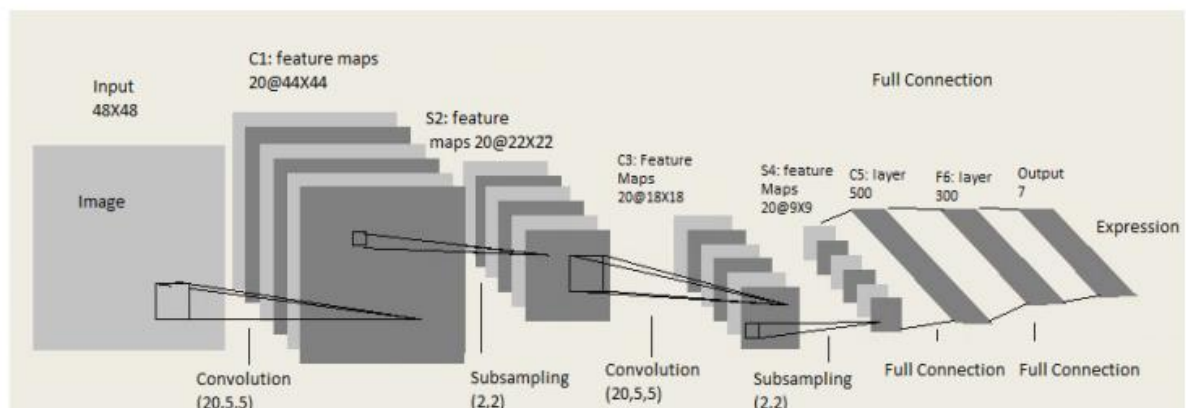


Figure 9: CNN Architecture

- **Input Layer:** It contains data in the form of an image. The parameters include height, width, depth and color information of the image (RGB). Input size is fixed to 224 X 224 RGB image.

- **Convo Layer:** Convolutional layer is also called feature extraction layer. This layer extracts the prominent features from the given collection of images using dot products of the image dimensions.
- **Max Pooling Layer:** The pooling layer helps to reduce the computational power in order to process the data by decreasing (or) reducing the dimensions of the featured matrix obtained by using the dot products.
- **Fully Connected Layer:** It comprises loads, neurons and biases. It connects neurons from one convolutional layer to another.

CHAPTER – 5

IMPLEMENTATION AND TESTING

5.1 Implementation

This part of the report illustrates the approach employed to classify the leaves into diseased or healthy and if the leaf is diseased, name of the disease is mentioned along with the remedies. Our methodology primarily revolves around the following five steps:

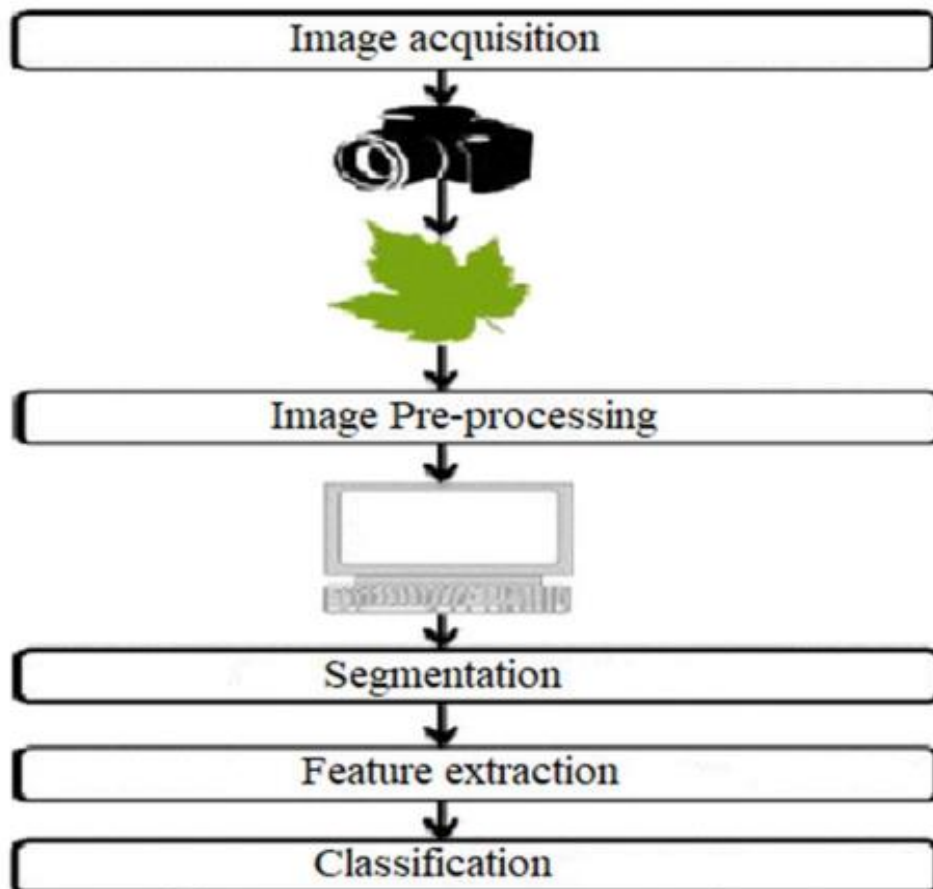


Figure 10: Implementation

- Algorithm written below illustrated the step-by-step approach for the proposed image recognition and segmentation processes:
- Image acquisition is the very first step that requires capturing an image with the help of a digital camera.
- Pre-processing of input image to improve the quality of image and to remove the undesired distortion from the image. Clipping of the leaf image is performed to get

the interested image region and then image smoothing is done using the smoothing filter. To increase the contrast, Image enhancement is also done.

- Mostly green coloured pixels, in this step, are masked. In this, we computed a threshold value that is used for these pixels. Then in the following way mostly green pixels are masked: if pixel intensity of the green component is less than the precomputed threshold value, then zero value is assigned to the red, green and blue components of this pixel.
- In the infected clusters, inside the boundaries, remove the masked cells.
- Obtain the useful segments to classify the leaf diseases.

5.1.1 Tools Used

Different tools were used during the development of this project. For example, HTML, CSS is used for UI designing, sklearn is used for data preprocessing and so on.

Table 5.1 Tools used for Plant Disease Detection

Tools name	Purpose
Visual Studio Code	IDE for developing software
Jupyter Notebook	Server-Side Programming
Flask	Framework
MS-Word	Documentation
Git, GitHub	Version Controlling
Lucid Chart	For Activity, Class, Sequence diagrams

5.1.1.1 Languages Used

- Programming Languages: Python
- Markup Languages: HTML
- Style-Sheet Languages: CSS

5.1.2 Implementation Details of Modules

- NumPy: For working with arrays
- Pandas: For data analysis and manipulation
- Matplotlib: For Visualization
- Sklearn : For Preprocessing of data
- Seaborn: For interacting and Visualization
- Flask: Web microframework for building web applications with Python.
- Keras : API designed for human beings, not machines.
- TensorFlow: End-to-end open source platform for ML

5.2 Testing

Testing is the process of testing the functionality and correctness of software. Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors.

5.2.1 Unit testing

Unit testing was performed to test correctness of different modules.

Objective

The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews.

Testing Process

- Checking for availability of Code Walk-thru reports which have documented the existence of and conformance to coding standards.
- Review of Unit Test Specifications
 - Verify the Unit Test Specifications conform to the program specifications.
 - Verify that all boundary and null data conditions are included

5.2.2 System testing

System testing tests a completely integrated system and outputs generated by it to verify that the system meets its requirements. It is also used to check logic changes made in it with the intention of finding errors. This process helps in validating the system by testing

the system as a whole that covers each module of the application, database specifications and underlying configurations.

5.3 Result Analysis

Prediction is far better if there is white background with no other object in the frame of the camera apart from the hand. Accuracy increases with the condition applied. When we try to load the healthy and infected leaves of the plants, we are successful to detect the healthy and infected leaves with the name of the diseases and the various ways to eliminate the infections with the necessary supplements provided and also the types of diseases are classified. We gain accuracy of more than 98 Percent.



Figure 11: Training and Validation Loss

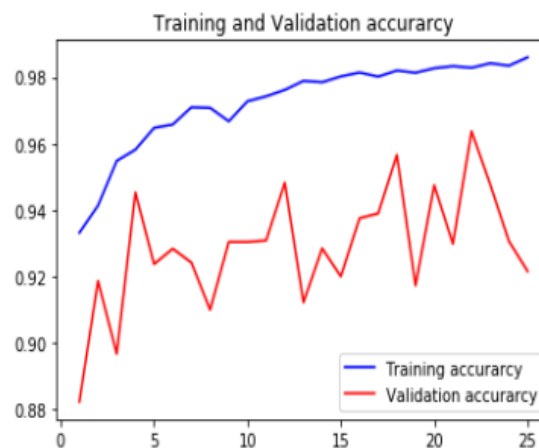


Figure 12: Training and Validation accuracy

CHAPTER – 6

CONCLUSION AND FUTURE RECOMMENDATIONS

6.1 Conclusion

This project was undertaken to solve the underlying issue faced by farmers who can now identify the types of diseases and the curing method. They don't have to worry about which types of supplements they have to use, now they get the information on the webapp and can look up the supplements.

This project helps in eradicating the problems of crop dead due to which they were unknown about the types of diseases but now they are well aware about the diseases and the curing methods.

The application provides the necessary platform to solve with much ease and gives them the ability to interpret without any external help.

6.2 Future Recommendations

- To improve recognition rate of final classification process hybrid algorithms like Artificial Neural Network, Bayes classifier, Fuzzy Logic can also be used.
- Mobile application can be developed which is handy and easy to use.
- An extension of this work will focus on automatically estimating the severity of the detected disease.
- As future enhancement of the project is to develop the open multimedia (Audio/Video) about the diseases and their solution automatically once the disease is detected.

REFERENCES

- A., K., & F., X. P.-B. (2018). "Deep learning in agriculture: a survey," Computers and Electronics in Agriculture". 70-90.
- D.B.victory. (2020). <https://www.gantt.com>. From planning-and-scheduling-with-gantt-chart: <https://www.gantt.com/blog/planning-and-scheduling-with-gantt-chart>
- G., W., Y., S., & J., W. (2017). Automatic image-based plant disease severity estimation using deep learning. 32-40.
- Zelner, S., & Fergus, W. (2014). Deep convolutional neural network. 43-48.
- K. Elangoran, S. Nalini, (2011) "Detection and classification of leaf diseases using K- means-based segmentation and neural-networks-based classification." Inform. Technol. J., 10: 267-275. DOI: 10.3923/itj.2011.267.275.
- S Arivazhagan, R Newlin shebiah, S Ananthi, S Vishnu varthini (2013) "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features."

Appendix

```
nn.Conv2d(in_channels=3, out_channels=32, kernel_size=3, padding=1),
nn.ReLU(),
nn.BatchNorm2d(32),
nn.Conv2d(in_channels=32, out_channels=32, kernel_size=3, padding=1),
nn.ReLU(),
nn.BatchNorm2d(32),
nn.MaxPool2d(2),
# conv2
nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, padding=1),
nn.ReLU(),
nn.BatchNorm2d(64),
nn.Conv2d(in_channels=64, out_channels=64, kernel_size=3, padding=1),
nn.ReLU(),
nn.BatchNorm2d(64),
nn.MaxPool2d(2),
# conv3
nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, padding=1),
nn.ReLU(),
nn.BatchNorm2d(128),
nn.Conv2d(in_channels=128, out_channels=128, kernel_size=3, padding=1),
nn.ReLU(),
nn.BatchNorm2d(128),
nn.MaxPool2d(2),
# conv4
nn.Conv2d(in_channels=128, out_channels=256, kernel_size=3, padding=1),
nn.ReLU(),
nn.BatchNorm2d(256),
nn.Conv2d(in_channels=256, out_channels=256, kernel_size=3, padding=1),
nn.ReLU(),
nn.BatchNorm2d(256),
nn.MaxPool2d(2),
```


Layer (type)	Output Shape	Param #
=====		
Conv2d-1	[-1, 32, 224, 224]	896
ReLU-2	[-1, 32, 224, 224]	0
BatchNorm2d-3	[-1, 32, 224, 224]	64
Conv2d-4	[-1, 32, 224, 224]	9,248
ReLU-5	[-1, 32, 224, 224]	0
BatchNorm2d-6	[-1, 32, 224, 224]	64
MaxPool2d-7	[-1, 32, 112, 112]	0
Conv2d-8	[-1, 64, 112, 112]	18,496
ReLU-9	[-1, 64, 112, 112]	0
BatchNorm2d-10	[-1, 64, 112, 112]	128
Conv2d-11	[-1, 64, 112, 112]	36,928
ReLU-12	[-1, 64, 112, 112]	0
BatchNorm2d-13	[-1, 64, 112, 112]	128
MaxPool2d-14	[-1, 64, 56, 56]	0
Conv2d-15	[-1, 128, 56, 56]	73,856
ReLU-16	[-1, 128, 56, 56]	0
BatchNorm2d-17	[-1, 128, 56, 56]	256
Conv2d-18	[-1, 128, 56, 56]	147,584
ReLU-19	[-1, 128, 56, 56]	0
BatchNorm2d-20	[-1, 128, 56, 56]	256
MaxPool2d-21	[-1, 128, 28, 28]	0
Conv2d-22	[-1, 256, 28, 28]	295,168
ReLU-23	[-1, 256, 28, 28]	0
BatchNorm2d-24	[-1, 256, 28, 28]	512
Conv2d-25	[-1, 256, 28, 28]	590,080
ReLU-26	[-1, 256, 28, 28]	0
BatchNorm2d-27	[-1, 256, 28, 28]	512
MaxPool2d-28	[-1, 256, 14, 14]	0
Dropout-29	[-1, 50176]	0
Linear-30	[-1, 1024]	51,381,248
ReLU-31	[-1, 1024]	0
Dropout-32	[-1, 1024]	0
Linear-33	[-1, 39]	39,975
=====		
Total params: 52,595,399		
Trainable params: 52,595,399		
Non-trainable params: 0		

Input size (MB): 0.57		
Forward/backward pass size (MB): 143.96		
Params size (MB): 200.64		
Estimated Total Size (MB): 345.17		



```
train_losses, validation_losses = batch_gd(  
    model, criterion, train_loader, validation_loader, 5  
)
```



```
train_acc = accuracy(train_loader)  
test_acc = accuracy(test_loader)  
validation_acc = accuracy(validation_loader)
```

Plant Disease Detection

This AI Engine Will Help To Detect Disease From Following Fruits And Veggies

AI Engine



Apple



Blueberry



Cherry



Corn



Grape



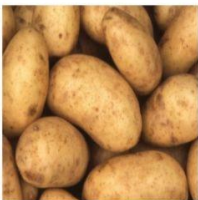
Orange



Peach



Pepper Bell



Potato



Raspberry



Soybean



Squash




Strawberry



Tomato

[Home](#) [AI Engine](#) [Supplements](#) [Contact-Us](#)




AI Engine

Let AI Engine Will Help You To Detect Disease

Why is it necessary to detect disease in plant ?

Plant diseases affect the growth of their respective species. In addition, some research gaps are identified from which to obtain greater transparency for detecting diseases in plants, even before their symptoms appear clearly, diagnosis is one of the most important aspects of a plant pathologist's training. Without proper identification of the disease and the disease-causing agent, disease control measures can be a waste of time and money and can lead to further plant losses. Proper disease diagnosis is necessary.



Choose File

No file chosen

Simply upload your plant's leaf image and then see the magic of AI.

Submit

Prevent Plant Disease follow below steps:

1. Follow Good Sanitation Practices.
2. Fertilize to Keep Your Plants Healthy.
3. Inspect Plants for Diseases Before You Bring Them Home.
4. Allow the Soil to Warm Before Planting.
5. Ensure a Healthy Vegetable Garden By Rotating Crops.
6. Provide Good Air Circulation
7. Remove Diseased Stems and Foliage

More info

[Home](#) [Services](#) [About](#) [AI Engine](#)

Created by Kushal and Ashish.

28

Apple : Black Rot



Brief Description :

Leaf symptoms first occur early in the spring when the leaves are unfolding. They appear as small, purple specks on the upper surface of the leaves that enlarge into circular lesions 1/8 to 1/4 inch (3-6 mm) in diameter. The margin of the lesions remains purple, while the center turns tan to brown. In a few weeks, secondary enlargement of these leaf spots occurs. Heavily infected leaves become chlorotic and defoliation occurs. As the rotted area enlarges, a series of concentric bands of uniform width form which alternate in color from black to brown. The flesh of the rotted area remains firm and leathery. Black pycnidia are often seen on the surface of the infected fruit.

To prevent plant from diseases, follow these steps:

Remove the cankers by pruning at least 15 inches below the end and burn or bury them. Also take preventative care with new season prunings and burn them, too. You are better off pruning during the dormant season. This will minimize the odds that fire blight will infect your tree and produce dead tissue that can easily be infected by Botryosphaeria. You should also take precautions if the bark is damaged by hail, or branches break during a windstorm. Using a copper-based fungicide will protect against both black rot and fire blight.

Supplements :



Magic FungiX For Fungal disease

[Buy Product](#)

Supplements

Buy Supplements & Fertilizer at one place

Supplements (Diseased):



For Apple : Scab
Katayani Prozol Propiconazole 25%
EC Systematic Fungicide

[Buy Product](#)

Supplements (Diseased):



For Apple : Black Rot
Magic FungiX For Fungal disease

[Buy Product](#)

Supplements (Diseased):



For Apple : Cedar rust
Katayani All in 1 Organic Fungicide

[Buy Product](#)

Fertilizer (Healthy):



For Apple : Healthy
BIO COMPOST Plant Fertilizer

[Buy Product](#)

Fertilizer (Healthy):



For Blueberry : Healthy
GreenStix Fertilizer

[Buy Product](#)

Supplements (Diseased):



For Cherry : Powdery Mildew
ROM Mildew Clean

[Buy Product](#)

Fertilizer (Healthy):



For Cherry : Healthy
Plantic Organic BloomDrop Liquid
Plant Food

[Buy Product](#)

Supplements (Diseased):



For Corn : Cercospora Leaf Spot |
Gray Leaf Spot
ANTRACOL FUNGICIDE

[Buy Product](#)