A Minor Project Proposal Report on

# PlantCare: Disease Detection System

Submitted in Partial Fulfillment of the Requirements for the

Degree of **Bachelor of IT Engineering**

Under Pokhara University

Submitted by:

**Prakash Mahara, 211527**

**Prashanta Rokaya, 211528**

**Suman Paneru, 211536**

Date:

13 Mangsir 2081

Department of IT Engineering

**NEPAL COLLEGE OF INFORMATION TECHNOLOGY**

Balkumari, Lalitpur, Nepal

# ABSTRACT

PlantCare: Disease Detection System is a web-based application that helps users detect plant diseases by analyzing uploaded images of plant leaves. The system uses machine learning, specifically convolutional neural networks (CNNs), to identify diseases based on visual patterns and symptoms. Once a disease is detected, the system provides recommendations for treatment and prevention. The goal of this project is to simplify plant care by enabling users to easily diagnose diseases, which typically require expert knowledge. The system allows users to upload images of plant leaves and receive instant feedback about possible diseases, along with actionable recommendations for care. The system is built using React.js for the frontend and Flask for the backend, with a machine learning model powered by TensorFlow. The model is trained on datasets like PlantVillage, which includes images of common plant diseases. This makes PlantCare an accessible and effective tool for both hobbyists and professional gardeners, helping them manage plant health more efficiently.

***Keywords:*** *PlantCare, MachineLearning, CNNs, React.js, Flask, TensorFlow*

**Table of Contents**

## LIST OF FIGURES

# 1. INTRODUCTION

Plants are an integral part of ecosystems and agriculture, but their health is often compromised by diseases caused by bacteria, fungi, and viruses. Identifying these diseases accurately and promptly is crucial for effective treatment and prevention. However, traditional methods of disease detection rely on expert knowledge and manual observation, which can be time-consuming, error-prone, and inaccessible for many users.

PlantCare: Disease Detection System addresses this challenge by providing a web-based platform that uses machine learning to detect plant diseases through image analysis. By leveraging convolutional neural networks (CNNs), the system identifies visual patterns and symptoms on plant leaves to diagnose diseases with high accuracy. Users can upload images of diseased leaves, and the system provides instant feedback, including the disease name and actionable recommendations for treatment and prevention.

## 1.1 PROBLEM STATEMENT

- Identifying plant diseases is a critical challenge for farmers and gardeners. Many plants show visible symptoms like spots, discoloration, and fungal growth, but accurate identification requires expertise. Current methods are time-consuming, prone to human error, and inefficient at scaling for large-scale applications. This project addresses the need for an automated solution to detect diseases quickly and accurately, enabling users to take timely corrective actions.

## 1.2 PROJECT OBJECTIVES

The main objectives of this project are:

- Develop a system capable of detecting plant diseases using machine learning models.
- Enable users to upload plant images for real-time analysis.
- Provide disease-specific recommendations for treatment and prevention.
- Create a responsive and user-friendly web interface.

## 1.3 SCOPE AND LIMITATION

In the project plantcare disease detection system, system will be developed that detects the plant’s disease.

**SCOPE:**

- Detect diseases for plants included in the training dataset (e.g., PlantVillage).
- Provide treatment recommendations based on detected diseases.
- Accessible through any modern browser on both desktop and mobile devices.

**LIMITATIONS:**

- Disease detection is restricted to the trained dataset.
- Requires high-quality images for optimal results.
- Cannot detect multiple diseases in a single image.

## 1.4 SOFTWARE REQUIREMENTS

- Frontend Development: JavaScript, React.js
- Backend Development: Flask
- Database: MySQL
- IDE: VS-Code

Here's a brief introduction to JavaScript, React.js, MySQL, Flask and VS-Code:

**JavaScript:**

- JavaScript is a versatile scripting language that enables interactive and dynamic content on web pages. It is commonly used for client-side scripting, allowing developers to create responsive and engaging user interfaces.

**React.js:**

- React.js is a JavaScript library for building dynamic and interactive user interfaces. It simplifies development with its component-based structure and efficient virtual

DOM. React enables fast updates, making it ideal for scalable and engaging web applications.

**MySQL:**

- MySQL is a relational database management system (RDBMS) that stores and retrieves data. It is widely used for managing databases in web applications, providing a robust and scalable solution for storing, querying, and manipulating structured data.

**Flask:**

- Web server will be created using flask. Flask provides a way to send command from webpage to server over the internet. Flask is a micro framework for python which allows us to run the python script through the webpage.

**VS-Code:**

**-**VS-Code is a source-code editor developed by Microsoft for Windows, Linux, macOS and web browsers. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded version control with Git.

## 2. LITERATURE REVIEW

### Machine learning in detecting and classifying diseases of a plant leaf

Identification of diseases is one of major area in agriculture which needs to be taken care of, though many practices have been done and implemented to cope up with this issue, rapid and quick identification of the diseases still remains in state of inchoate. The use machine learning in facilitating the identification and detection helps to counter this problem to a much greater extent.

### Detection of abnormalities of the leaves of plants and training using papaya leaves.

This paper talks about the detection and recognition of abnormalities of plants for training and study papaya leaves were taken. Random forest classifier was used for classification and it got trained using images of leaves with an almost seventy percent accuracy [2].

### Apple leaf disease detection.

The common Apple leaf diseases like rust, grey spot, brown spot were discussed and found out with the help of deep learning algorithms and improved CNNs. The dataset for diseased leaves were generated, processed and collected. New deep CNN model designed to identify small diseased spots [3].

### Image processing techniques in identifying fungal crop diseases.

The most common bacterial, fungal and viral diseases are studied which affects the plant leaves and roots on wide scale and reduces the productivity of the plants can be easily studied and identified through RGB to grey scale conversions [4].

### Selection of algorithms

In machine learning various algorithms are available for feature extraction, clustering, segmentation. Selecting the most suitable as per needs of the task can be tough at times. To reduce the complexity and improve the time of response selection of the most suitable algorithm is required. We did a comparative study of the algorithms used in various previous projects and reasoned out the best one suitable for the project.

### Classification

The algorithms used for classification are commonly applied in similar projects, such as Support Vector Machine (SVM) and Convolutional Neural Networks (CNN). Classification is a necessary step as it compares the values received after the feature extraction step with a pre-calculated set of data. For this project, we have opted to use the CNN algorithm, which is a powerful deep learning method particularly suited for image classification tasks. CNNs are highly effective at recognizing patterns and features in images, making them ideal for plant disease detection.

# 3. METHODOLOGY

## 3.1 SOFTWARE DEVELOPMENT LIFECYCLE

We will use the iterative Model of software development, which involves incremental enhancements until the final system is achieved.

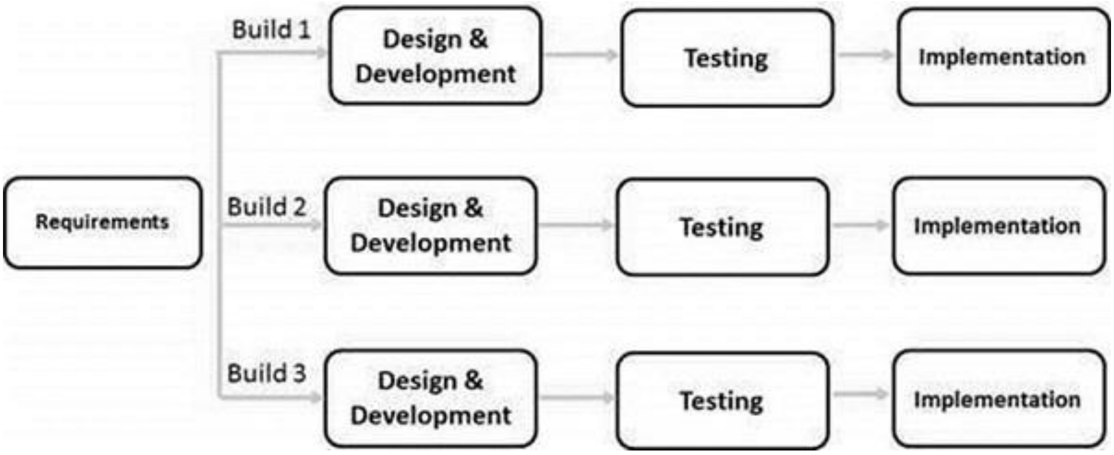

*Figure 1: Iterative Model*

Key phases include:

1. **REQUIREMENT ANALYSIS**:
    a. Identify plant diseases, datasets, and machine learning frameworks.
    b. Specify user workflows for image uploads and result visualization.

2. **DESIGN PHASE**:
    a. Create diagrams: context diagrams, data flow diagrams, and use case diagrams.
    b. Design a responsive user interface with React.js and Bootstrap.

3. **CODING PHASE**:
    a. Implement the frontend using React.js for an interactive experience.
    b. Use Flask to create backend APIs for communication with the machine learning model.

4. **TESTING PHASE**:
    a. Test disease detection accuracy using a test set from the PlantVillage dataset.
    b. Validate frontend responsiveness across devices and browsers.

## 3.2 PROPOSED SYSTEM

The proposed system for the PlantCare project focuses on the early detection and classification of plant leaf diseases using image processing and machine learning techniques. The system workflow is depicted in the provided diagram and involves the following steps:

- Input Leaf Image: Users upload images of plant leaves through the system interface. These images serve as inputs for disease detection.
- Pre-processing: The uploaded images undergo pre-processing steps, such as resizing and enhancement, to standardize the data and improve detection accuracy.
- Feature Extraction: The system extracts relevant features from the pre-processed images. These features are crucial for identifying patterns associated with different plant diseases.
- Database Integration: The system leverages a database containing pre-labeled training images of healthy and diseased leaves. This database is used to train the machine learning model.
- CNN Classification: A Convolutional Neural Network (CNN) model is employed to classify the leaf images. The CNN analyzes the extracted features to determine whether the leaf is "Normal" (healthy) or "Abnormal" (diseased).
- Defect Region Classification: If a leaf is classified as "Abnormal," the system further identifies and classifies the specific disease affecting the leaf, highlighting the defected regions.

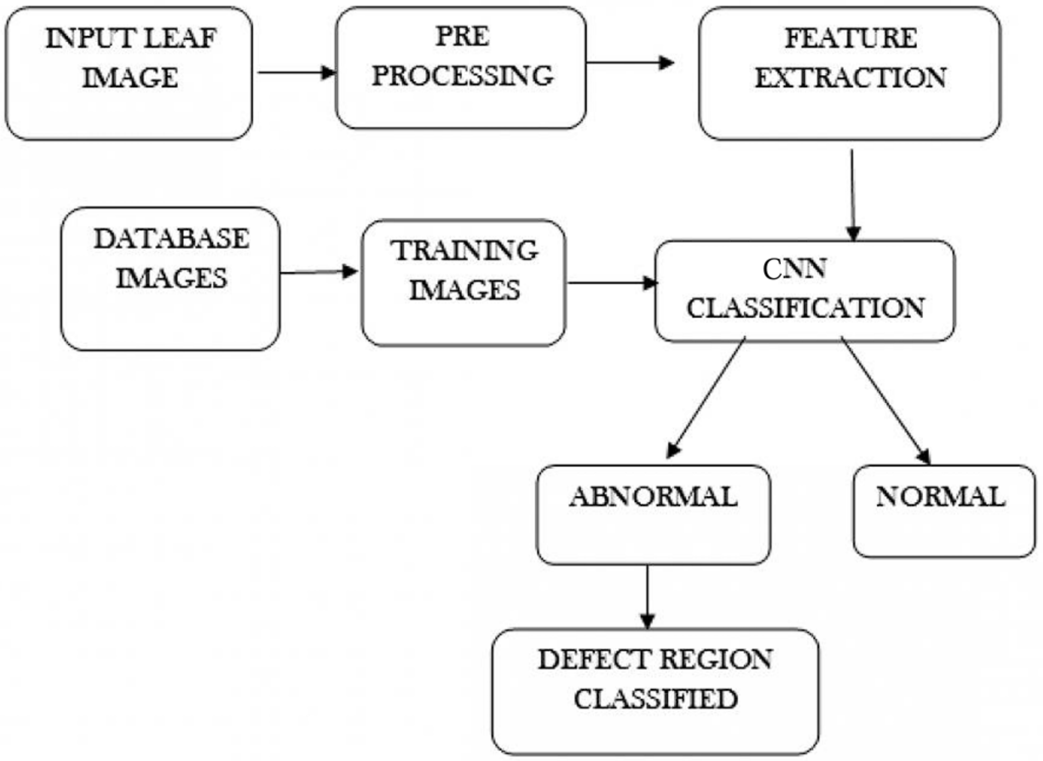

*Figure 2: System Architecture*

## 3.3 USE CASE DIAGRAM

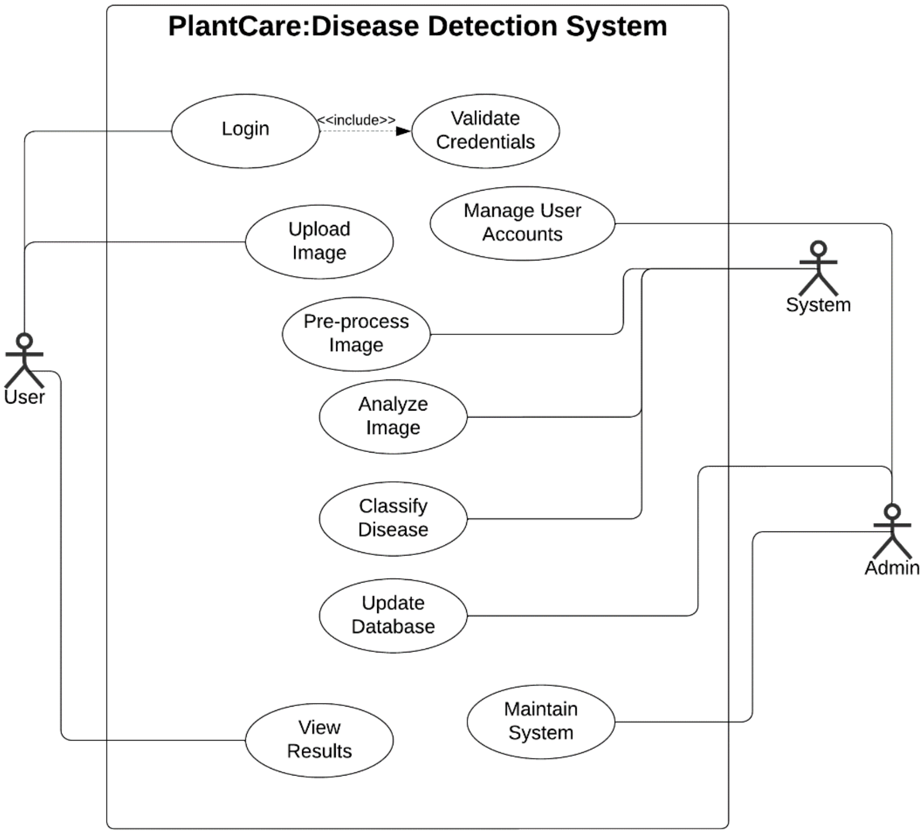

*Figure 3: Use Case Diagram*

# 4. PROJECT DELIVERABLE

The system proposed primarily aims to detect plant diseases using image processing and machine learning techniques. Disease detection will be performed by analyzing uploaded images of plant leaves, where the system identifies abnormalities and classifies the disease using a trained machine learning model. Users can upload the leaf images on the user site, receive disease detection results, and get recommendations for effectively managing the detected plant issues. The system will offer an intuitive interface designed to cater to users with varying levels of technical expertise, ensuring ease of use for both beginners and experienced gardeners.

## 5. PROJECT TIMELINE

The project is divided into two iterations: the first focuses on initial development, including requirements analysis, dataset preparation, model training and testing. The second iteration refines these tasks, enhancing features and fine-tuning the system. Documentation runs parallel throughout, with the project concluding in deployment and finalization within the two-month timeline.

| Task | Duration (Weeks) | Start Date | End Date | Description |
|---|---|---|---|---|
| Requirements Analysis & Design | 1 | Week 1 | Week 1 | Define project requirements, scope, and design architecture. |
| Dataset Collection & Preprocessing | 1 | Week 2 | Week 2 | Gather plant disease images, clean, preprocess, and prepare the dataset. |
| Model Training and Evaluation | 2 | Week 3 | Week 4 | Train a CNN model, evaluate its accuracy, and finalize it for integration. |
| Frontend Development (React.js) | 2 | Week 4 | Week 5 | Develop the React.js frontend for uploading images and displaying results. |
| Backend Development (Flask) | 2 | Week 4 | Week 5 | Build Flask APIs for processing uploaded images and connecting the model. |
| Frontend and Backend Integration | 1 | Week 6 | Week 6 | Integrate the React.js frontend with the Flask backend for seamless operation. |
| System Testing & Debugging | 1 | Week 7 | Week 7 | Perform end-to-end testing, debug issues, and optimize performance. |
| Deployment & Final Documentation | 1 | Week 8 | Week 8 | Deploy the system to the cloud and finalize project documentation. |

# 5.1 GANTT CHART

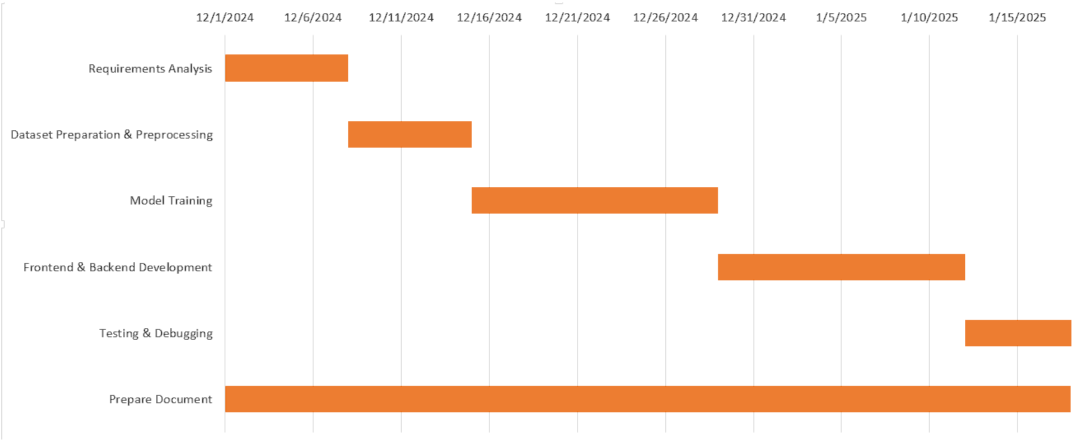

*Figure 4: Iteration 1*

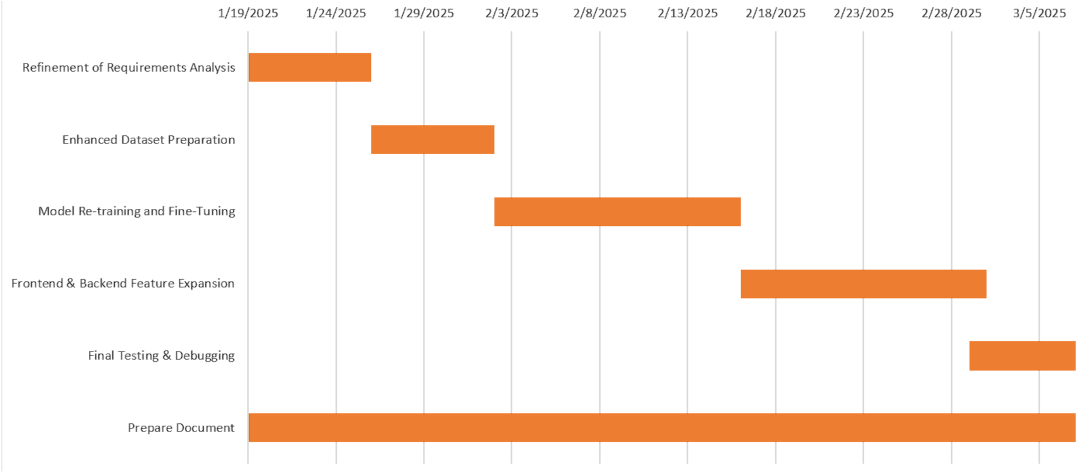

*Figure 5: Iteration 2*

# 6. REFERENCES:

[1] PlantVillage Dataset, https://www.kaggle.com/datasets/emmarex/plantdisease. [Accessed: Nov. 28, 2024].

[2] Z. N. Reza, F. Nuzhat, N. A. Mahsa, and M. H. Ali, "Detecting jute plant disease using image processing and machine learning," in *Proc. 3rd Int. Conf. on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, 2016, pp. 1-6.

[3] S. Ramesh *et al.*, "Plant Disease Detection Using Machine Learning," in *Proc. 2018 Int. Conf. on Design Innovations for 3Cs Compute Communicate Control (ICDI3C)*, Bangalore, 2018, pp. 41-45.

[4] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, “Identification and classification of fungal disease affected on agriculture/horticulture crops using image processing techniques,” in *Proc. IEEE Int. Conf. on Computational Intelligence and Computing Research*, Coimbatore, 2014, pp. 1-4.

[5] NCIT, "NCIT LMS," [Online]. Available: info@ncit.edu.np