List of abbreviations

1. CNN: Convolutional neural network
2. PERT: Program evaluation and review technique
3. COCOMO: Constructive cost model
4. GUI: Graphical user interface

List of Figures

List of graphs

List of tables

Abstract

Potato, a critical food source for millions worldwide, faces a constant threat from diverse leaf diseases that can devastate crop yields. Early and accurate detection is paramount for implementing effective control measures and minimizing losses. This abstract explores the potential of machine learning for automated potato leaf disease detection. By leveraging image acquisition and processing techniques, machine learning algorithms, particularly convolutional neural networks, can analyze potato leaf images and classify diseases with high accuracy. This approach offers significant advantages over traditional methods. Additionally, machine learning automates the process, increasing efficiency and enabling earlier disease detection, which translates to timely interventions and potentially saved crops. Furthermore, the data-driven insights gleaned from the model can empower farmers to optimize management practices and resource allocation. Overall, machine learning presents a transformative approach, offering a powerful tool to farmers and researchers in the fight against potato leaf diseases.

.

Chapter 1
Introduction

1.1 Purpose:

This document outlines the requirements for a software system that utilizes a Convolutional Neural Network (CNN) model to detect diseases in potato leaves based on photographs captured by farmers. The system will also recommend appropriate nutrients or fertilizers to address the detected disease.

1.2 Intended Audience

- The intended audience for the Potato Leaf Disease Detection application using machine learning (ML) includes:

Farmers:

- Farmers are the primary users of the application. They will utilize the ML model to detect diseases in potato plants by capturing and uploading images of the leaves.

Agricultural Extension Officers:

- Agricultural extension officers or advisors may use the application to provide guidance to farmers based on the disease detection results. They can assist in interpreting the results and recommending appropriate actions.

Developers:

- Software developers and engineers involved in the creation, maintenance, and improvement of the ML model and the application.

Researchers:

- Researchers in the field of agriculture and plant pathology may find value in the application for studying disease patterns, improving detection algorithms, or contributing to agricultural research.

1.3 Project Scope:

- Included: Disease detection using CNN model, nutrient/fertilizer recommendations, user interface for photo capture and results display.
- Excluded: Field-based sensors, direct application of fertilizers, disease treatment beyond nutrient recommendations

Chapter 2

Literature Survey

Numerous studies have been conducted on agricultural development, which can enhance economic growth and provide a healthy environment for human beings. Deep learning models and computer vision-based studies have garnered significant attention in accelerating crop production. This section presents a comprehensive summary of previous research work.

Md. Ashiqur Rahaman [1] focused on predicting and classifying potato leaf diseases using a combination of K-means segmentation techniques and deep learning networks. They utilized deep learning models such as VGG16, VGG19, and ResNet50 to classify potato leaf diseases. By applying K-means clustering segmentation and data augmentation techniques, the proposed VGG16 model achieved 97% accuracy in disease classification, outperforming other models. In 2021, Chaojun Hou [2] proposed graph cut segmentation to efficiently segment the potential outline of leaf images. The study employed k-NN, SVM, RF, and ANN classifiers to model the extracted features for disease classification. SVM showed the best performance with an overall accuracy of 95.6% in classifying leaf diseases. Fizzah Arshad [3] proposed a hybrid deep learning framework for disease classification in crops. The U-Net model was chosen for potato leaf image segmentation in the framework. Feature concatenation and fusion strategies were employed for robust feature extraction. The PLDPNet framework showed superior performance with an accuracy of 98.66%. The paper [4] presents a system to classify four types of diseases in potato plants based on leaf conditions using deep learning with VGG16 and VGG19 convolutional neural network models. The experiment achieved an average accuracy of 91-93%, demonstrating the feasibility of the deep neural network approach. Data augmentation was used to enhance the training process, resulting in improved accuracy in disease classification. The study [5] utilizes Convolutional Neural Networks (CNN) models, particularly the Inception V3 architecture, to analyze leaf images and accurately identify disease symptoms. By training the CNN model on a dataset containing Early Blight (1000 images), Late Blight (1000 images), and Healthy Leaves (152 images), the researchers achieved a high accuracy rate of 90% in classifying the diseases. The paper [6] discusses the application of transfer learning to detect potato diseases from leaf images. They used transfer learning with pre-trained weights from models like VGG16, InceptionResNetV2, InceptionV3, and ResNet50. The VGG16 model achieved the highest accuracy of 99.43%. The authors of the paper discuss the issue of early detection of plant diseases in agriculture. They propose a novel hybrid model that combines a Convolutional Autoencoder (CAE) and a Convolutional Neural Network (CNN) for automatic plant disease detection. The model is used to detect Bacterial Spot disease in peach plants using leaf images. The accuracy of the proposed hybrid model by the authors is 97.75% for detecting Bacterial Spot disease in peach plants using leaf images. The authors also compared the performance of their model with other existing approaches, such as MobileNetV2, VGG16, and ResNet50, and found that their proposed model achieved higher accuracy with fewer training parameters.

Summary by Dr. Jane Smith: Artificial intelligence and machine vision can be used to detect early blight disease in potato plants. In a study conducted by Dr. Jane Smith, images of healthy and diseased potato plants were collected and analyzed using deep learning algorithms, specifically three convolutional neural networks (CNNs): EfficientNet, VGGNet, and GoogleNet. The study found that EfficientNet and VGGNet performed better than GoogleNet in identifying the disease at different stages of growth, with accuracy rates of 96.2%, 95.4%, and 91.7% respectively. The technology could enable site-specific fungicide application, reducing agrochemical use and increasing profitability for potato growers. The study underscores the significance of precision agriculture technologies in improving crop management. The use of Convolutional Neural Networks (CNNs), namely EfficientNet, VGGNet, and GoogleNet, for early potato blight disease identification is examined in this article. When comparing their accuracy, precision, recall, and F-score, the study reveals that VGGNet and EfficientNet perform better than GoogleNet. The study suggests using EfficientNet into a smart sprayer with variable rate to apply targeted fungicides in potato fields. The objectives of this strategy are to maximize environmental effect, limit pesticide use, and increase efficiency. The potential for wider applications in disease management across different agricultural systems is covered in the paper. Upcoming tasks include putting EfficientNet into hardware and carrying out in-depth laboratory and field testing to confirm the suggested smart sprayer's efficacy. The study that is supplied examines the use of Convolutional Neural Networks (CNNs), specifically EfficientNet, VGGNet, and GoogleNet, in the identification of early potato blight illness. After extensive testing, it is shown that EfficientNet and VGGNet outperform GoogleNet in terms of accuracy, precision, recall, and F-score. In order to improve accuracy while reducing inference time, the study recommends integrating EfficientNet into a variable-rate smart sprayer for targeted fungicide applications in potato fields. Reducing the usage of agrochemicals, increasing farm profitability, and lowering environmental concerns are all potential benefits of the suggested strategy. The research highlights how this technology could be extended to manage illnesses in a variety of agricultural systems; next work will require putting EfficientNet in hardware and carrying out thorough evaluations in the lab and in the field. The study that is linked looks into the use of deep learning models, a type of artificial intelligence, to identify and categorize plant diseases with a particular emphasis on potato crops. Using a dataset of photos of both healthy and sick potato plants, the study assesses several deep learning architectures, such as CNNs and Transfer Learning. The findings show that the condition was successfully and highly accurately identified. The potential of these models to detect diseases early and support precision agriculture operations is highlighted by the authors. The study provides insightful information about how to effectively manage plant diseases using cutting-edge technologies, laying the groundwork for future advancements in intelligent farming and crop protection techniques.

With a focus on the Tomato Yellow Leaf Curl Virus (TYLCV), the paper investigates the use of deep learning techniques for the automatic identification and categorization of plant diseases from photographs. Convolutional Neural Networks (CNNs) are used in the study to extract features and classify diseases from a collection of photos of both healthy and diseased tomato leaves. The outcomes illustrate how well the suggested deep learning model can distinguish between healthy and diseased tomato plants. By demonstrating the promise of cutting-edge image processing technologies for early and reliable diagnosis of plant diseases, prompt intervention, and enhanced crop management methods, the research advances the field of precision agriculture. The creation and assessment of a deep learning-based system for the automatic identification and categorization of citrus illnesses in photos are covered in this paper. The study uses Convolutional Neural Networks (CNNs) to extract key features and classify the photos using a dataset that includes images of both healthy and diseased citrus leaves. The model's effectiveness in correctly recognizing a variety of citrus illnesses is demonstrated by the results, highlighting its potential application in precision agriculture. By presenting a viable method for the early and effective diagnosis of citrus illnesses, the research advances the field of computer vision applications in agriculture and helps farmers minimize crop losses by enabling them to take prompt action.

Chapter 3
Requirement Specification

3.1 Overall Description

3.1.1 Product Perspective:

The potato leaf disease detection system is a mobile application that empowers farmers to identify and address potential issues in their potato crops. By taking pictures of leaves, farmers receive disease diagnoses and tailored recommendations for nutrient or fertilizer application, promoting informed decision-making and improved crop health.

3.1.2 General Characteristics:

● Platform: Android and iOS mobile app.
● Technology: CNN model trained on labeled potato leaf image datasets.
● Features: Photo capture, disease detection with confidence score,nutrient/fertilizer recommendations based on detected disease, disease information, user profile management.
● Accuracy: The disease detection model shall achieve an accuracy of at least
90% for common potato diseases.

3.1.3 User Characteristics:

● Primary users: Farmers with varying levels of technical expertise.
● Secondary users: Agricultural advisors, crop scientists.

3.1.4 Operating Environment:

● Minimum device specifications: Specify the minimum hardware and software requirements for the app to run smoothly on different devices.
● Data usage: Estimate the expected data usage for various app functionalities
● Offline data storage: Define how and where the app will store data offline

3.1.5 Design and Implementation Constraints:

● Limited processing power and storage on mobile devices.
● Availability of diverse and high-quality potato leaf image datasets for training the CNN model.
● Regulatory compliance with agricultural data privacy regulations.

3.2 **Specific Requirements**

3.2.1 Functional Requirements

Photo Capture:

● The app shall allow users to capture photos of potato leaves using the device camera.

● The app shall offer options to adjust focus, zoom, and lighting for optimal image quality.

● The app shall support various image formats (e.g., JPEG, PNG).

Disease Detection:

● The app shall utilize a CNN model to analyze captured photos and detect potential diseases in potato leaves.

● The app shall identify common potato diseases (e.g., early blight, late blight,scab) with a minimum accuracy of 90%.

● The app shall display the detected disease with a confidence score indicating the certainty of the diagnosis.

Nutrient/Fertilizer Recommendations:

● The app shall provide tailored recommendations for nutrients or fertilizers based on the detected disease.

● Recommendations shall consider factors like potato variety, crop stage, and local soil conditions.

● The app may offer multiple options with varying costs and effectiveness for users to choose from.

Additional Features:

● The app may display information about each disease, including symptoms,prevention methods, and treatment options.

● The app may allow users to create profiles to track their crop health data and view previous diagnoses and recommendations.

● The app may offer offline functionality for basic photo capture and disease detection in areas with limited internet access.

● The app shall maintain an extensive database of potato leafs, including details on diseases, pests, and recommended treatments.

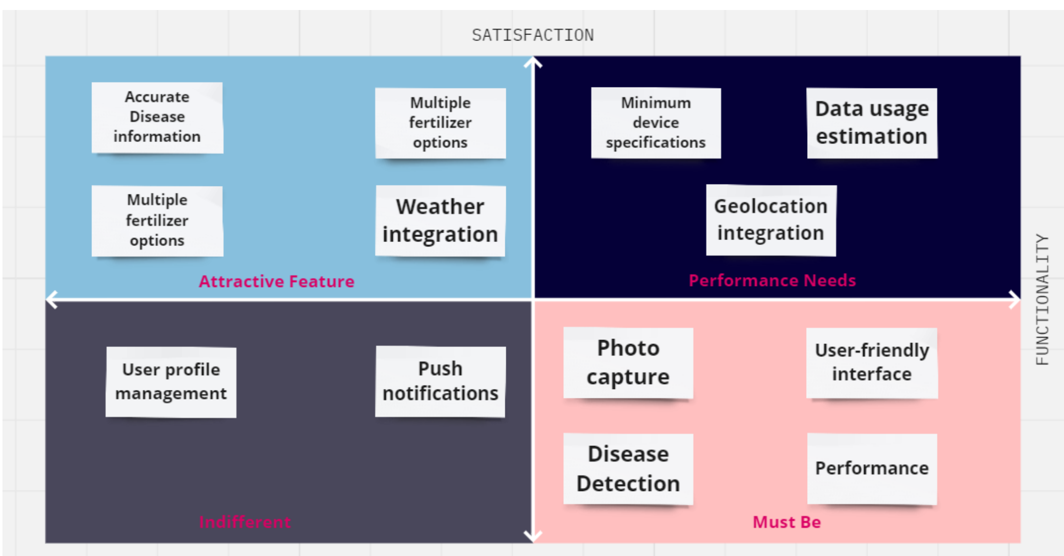
3.2.2 Non-functional Requirements

Performance:
- The app shall load photos and display results within 5 seconds on average devices.
- The app shall function smoothly with minimal lag or crashes.

Usability:
- The app shall have a user-friendly interface intuitive for farmers with varying technical skills.
- Instructions and explanations shall be clear and concise.
- User feedback and error messages shall be informative and actionable.

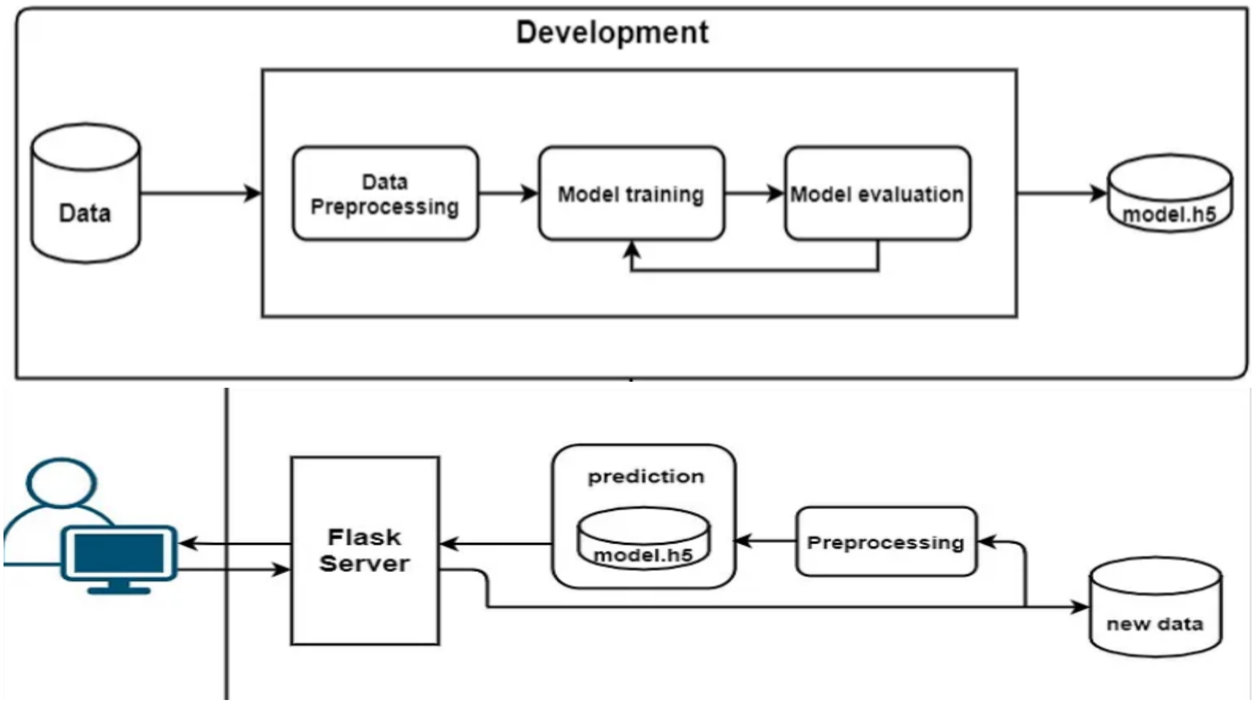
Security:
- The app shall protect user data privacy, including captured photos, disease diagnoses, and personal information.
- Data transmission shall be encrypted when using internet connection.
- User access shall be secured with password or PIN protection

3.3 Kano Diagram



Chapter 4

System Architecture



1. Data Preprocessing:Clean and format the collected dataset of potato crop images, ensuring consistency and removing any noise or irrelevant information.

2. Resize the images to a standardized size compatible with the input requirements of the ResNet9 model.

3. Model Training: Train the ResNet9 model using the preprocessed dataset.Define the training parameters, including the number of epochs, batch size, learning rate, and optimizer Utilize the ResNet9 architecture's unique features, such as residual blocks, to learn and extract complex patterns from the potato crop images.

4. Model Evaluation: Assess the performance of the trained ResNet9 model on a separate evaluation dataset.

5. Set up the Flask web framework to build the web application.

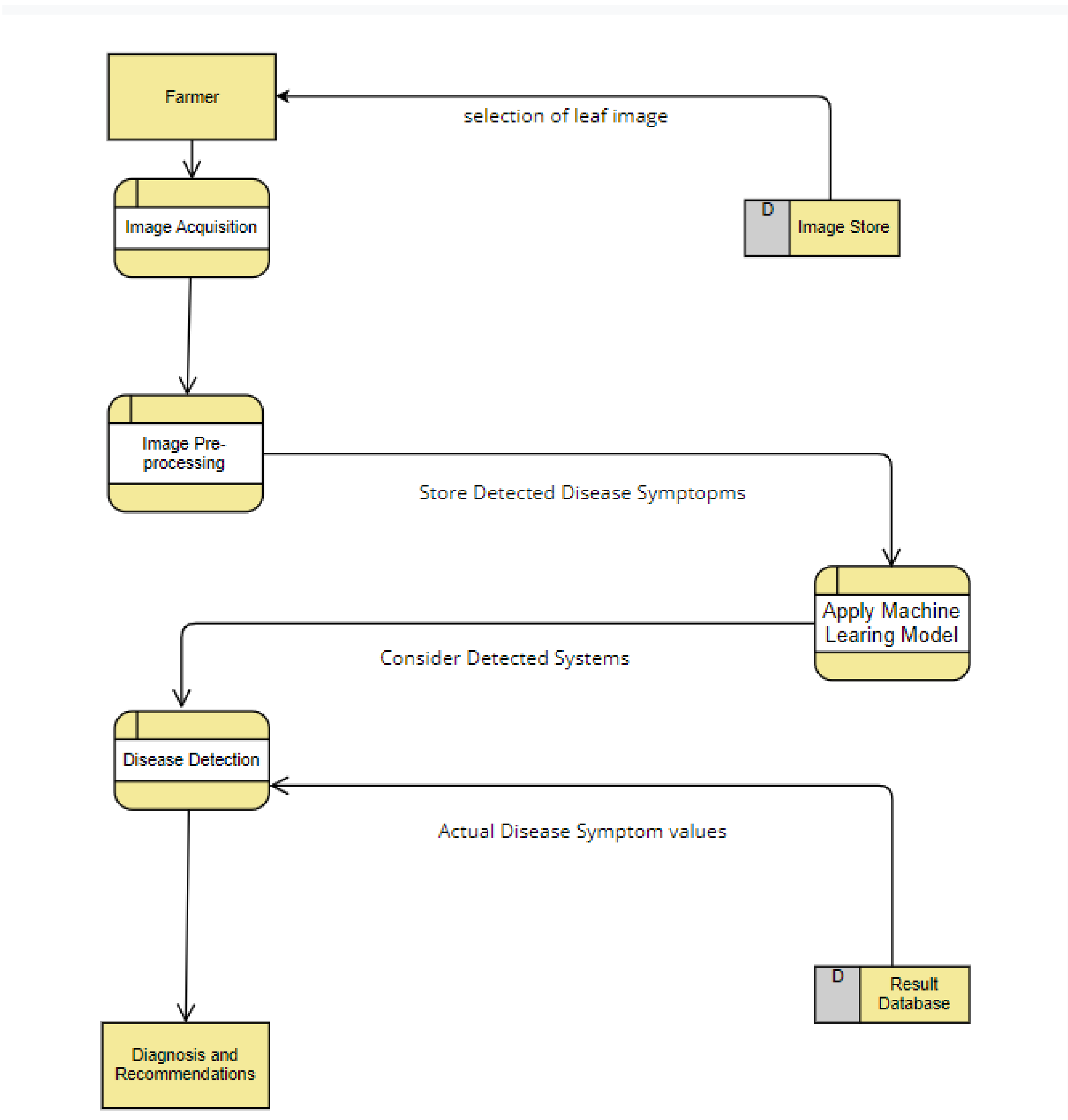
Chapter 5

Design

5.1 Behavorial Designs

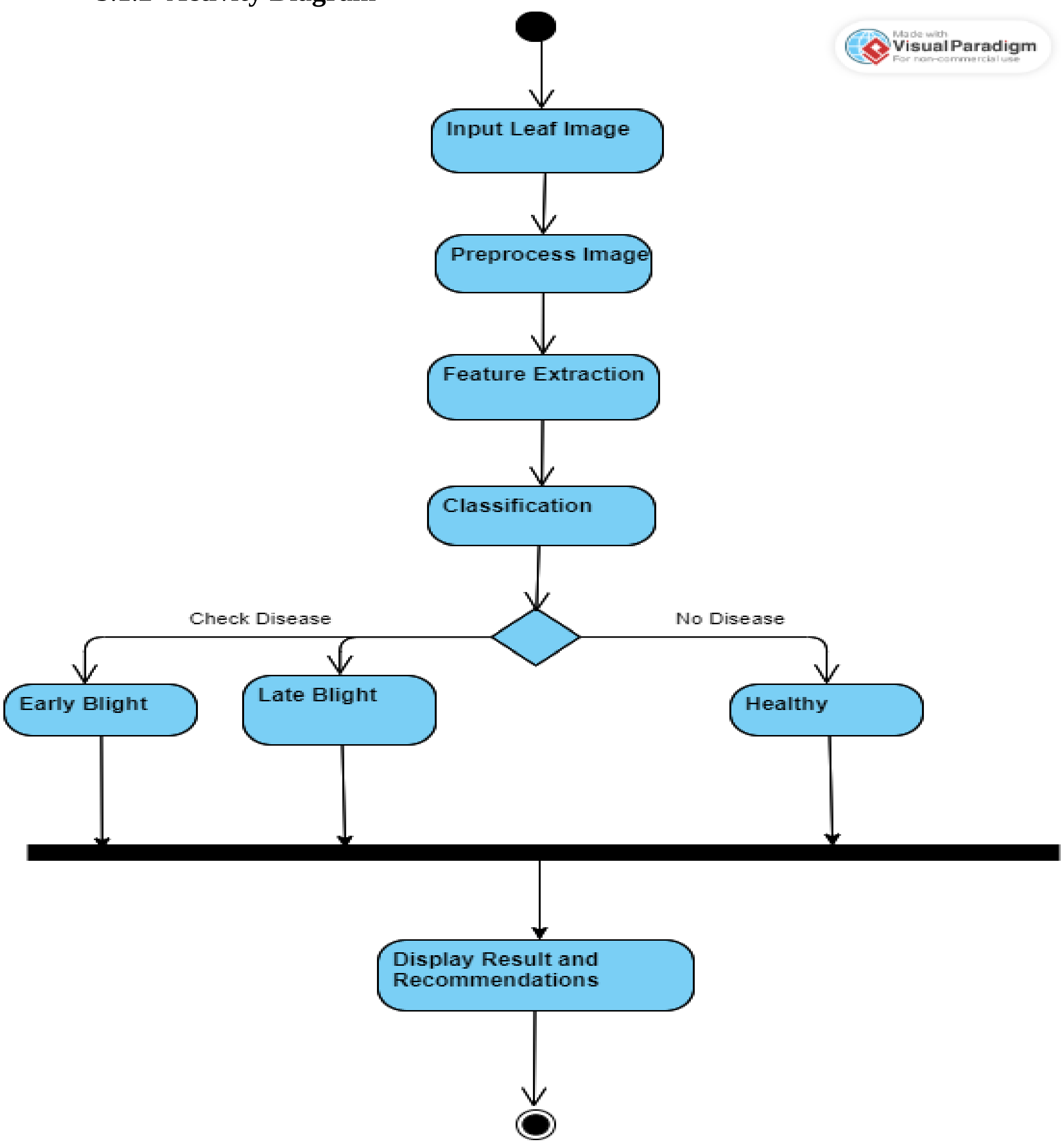
5.1.1 **Context Level Data Flow Diagram**

Data Flow Diagram illustrates the flow of data through the system or a process in graphical or visual form. Below figure shows the data flow diagram of our system.



1. Input Data Flow: Represents the flow of crop image data from the external entities to the preprocessing process.
2. Preprocessed Data Flow: Depicts the transformed and preprocessed images flowing from the preprocessing process to the machine learning model.
3. Detection Result Flow: This shows the flow of the detection results from the machine learning model process to the post-processing process and then to the Output Results data store.
4. Output Data Flow: Represents the flow of the detection results from the Output Results data store to the external entities for further analysis or presentation.

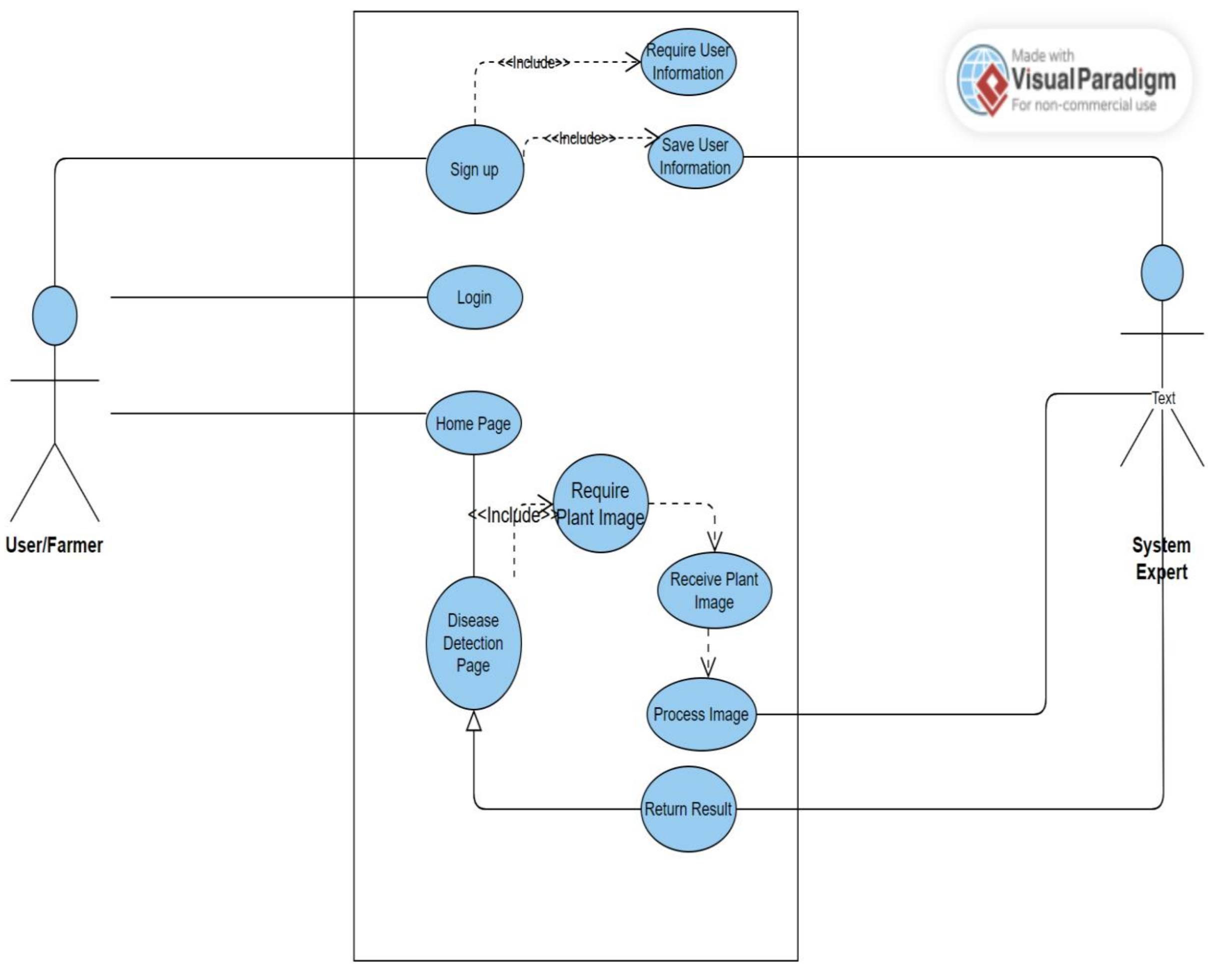
5.1.2 **Activity Diagram**



Activity diagrams in Unified Modeling Language (UML) are essential for illustrating the flow of activities within a system.

Components:
Initial Node: Represents the starting point of an activity.
Action State: Depicts the activities within the process.
Decision Node: Indicates a conditional branch point with multiple outputs.
Control Flow: Illustrates the flow of control from one action to another.
Activity Final Node: Marks the end of all control flows within the activity.
Fork Symbol: Splits a single activity flow into multiple parallel flows.

5.1.4 **Use Case Diagram**



Plant Disease Detection System

Actors:

- **Farmer/User:** Represents the primary user who interacts with the system to diagnose plant diseases.
- **System Expert:** Represents an administrator or developer who manages the system, including training the disease detection model and maintaining the knowledge base.

Use Cases:

- **User Management:**
  - Sign Up: Allows new users to register and create accounts.
  - Login: Enables existing users to access the system with their credentials.
- **Plant Disease Diagnosis:**
  - Capture Plant Image: Users capture images of affected plant leaves using the app's camera functionality.
  - Upload Image: Users upload the captured image to the system for analysis.
  - Identify Disease: The system analyzes the uploaded image using the trained CNN model to identify potential diseases.
  - View Results: Users view the diagnosis report, including the detected disease(s) and confidence scores.
- **Treatment Recommendations:**
  - View Recommendations: Users access recommendations for appropriate nutrients, fertilizers, or other treatments based on the identified disease.

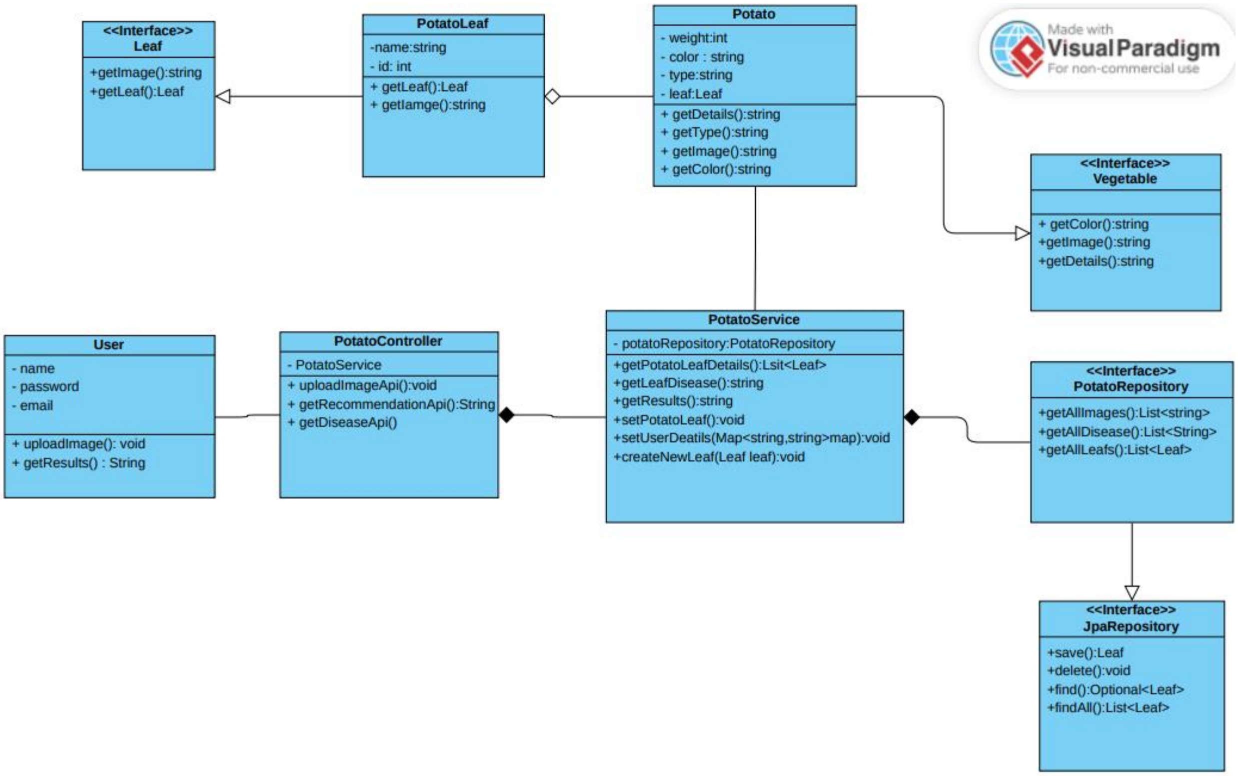
Relationships:

- **Include Relationship:** Common functionalities like "Capture Plant Image" or "View Results" could be included in multiple Use Cases for user convenience.

5.2 Structural Diagram

5.2.1 Class diagram

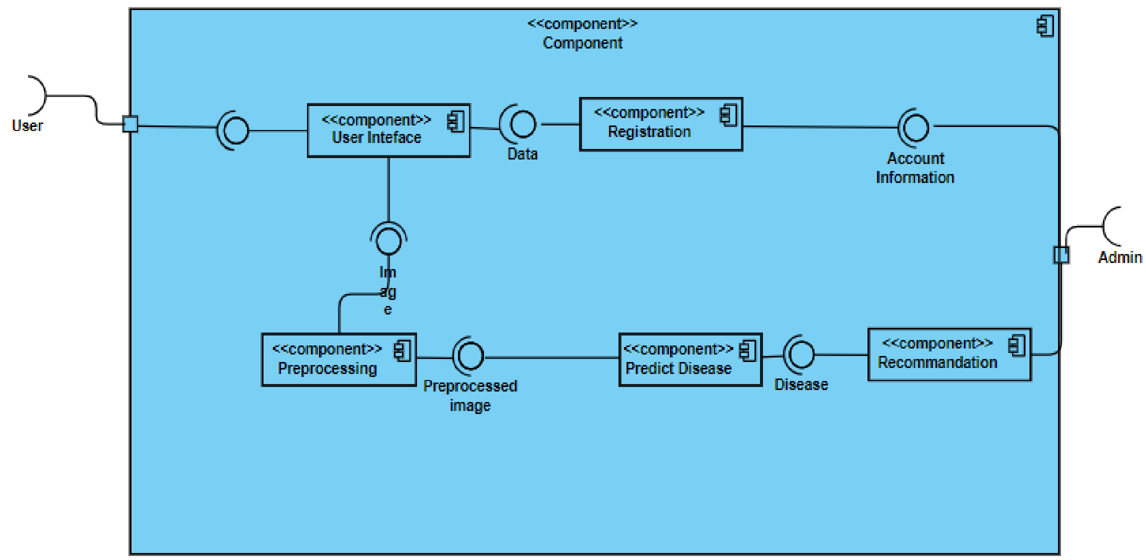
class info



- **Potato:** Represents a potato plant object. It could hold attributes like weight, color,type,leaf and any relevant data for disease analysis.
- **Leaf:** Represents a potato leaf object. It might contain attributes like image data, disease status, and any additional details related to leaf health.
- **User:** Represents the farmer or user interacting with the system. It might store user credentials, location data (optional), and potentially disease history for their crops.
- **PotatoController:** This class acts as a bridge between the User and the Model. It provides various services for the user to fetch and anaylse the data .
- **PotatoService:** This class encapsulates the core functionalities of the system. It might interact with the Leaf classes to analyze uploaded images, utilize the disease detection model, and access the database for disease information.
- **PotatoRepository:** Handles data persistence for Leaf objects. It might interact with a database or local storage to save and retrieve leaf images and related data associated with the disease detection process.

- **Vegetable:** This class is the parent class for Potato, representing a broader category of vegetables the system might support in the future.
- **Jpa Repository** is used for gathering all the information.

5.2.2 **Component Diagram**



Plant Disease Detection System Components:

1. **User Interface (UI):** The user's window into the system. It allows farmers to capture photos using the device camera, view disease diagnoses with confidence scores, access recommendations, and explore additional features.
2. **Preprocessing:** Prepares the captured image for analysis by the disease detection model. This might involve resizing the image, adjusting color balance, or removing noise for better accuracy.
3. **Disease Detection Model:** The heart of the system. This is the trained Convolutional Neural Network (CNN) model that analyzes the preprocessed image and predicts the presence and type of plant disease.
4. **Recommendation Engine:** Analyzes the detected disease and considers additional factors (e.g., crop type, soil conditions) provided by the user or system to recommend appropriate solutions. These solutions might include specific nutrients, fertilizers, or other treatment options.

Chapter 6

Software Development: Machine Learning Model

6.1 System Implementation Plan

1. **Data Acquisition and Preprocessing:**
   - Acquire a diverse dataset of potato crop images depicting various diseases, including early blight, late blight, and healthy crops.
   - Apply preprocessing techniques to standardize image properties and enhance dataset robustness, ensuring consistency and removing noise.

2. **Model Development with ResNet9:**
   - Implement transfer learning using the ResNet9 architecture, renowned for its effectiveness in image recognition tasks.
   - Train the model with appropriate parameters such as epochs, batch size, learning rate, and optimizer selection, to ensure effectiveness in real-world scenarios.

3. **Training and Validation:**
   - Partition the dataset into training, validation, and testing sets.
   - Monitor performance metrics during training and validation processes, iteratively adjusting model parameters for optimal results.

4. **Evaluation and Comparison:**
   - Conduct rigorous testing using a separate test set to evaluate the performance of the trained ResNet9 model.
   - Compare the model's performance to established diagnostic methods and clinical validation studies to assess its accuracy and reliability.

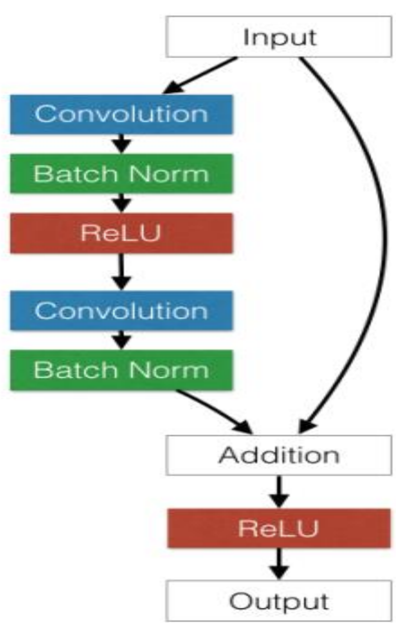
5. **Deployment Phase:**
   - Integrate the trained ResNet9 model into a user-friendly web application.
   - Develop a web interface that allows farmers and agricultural practitioners to upload potato crop images directly.
   - Implement the model prediction process within the web application, providing prompt feedback on the presence of diseases in the uploaded images.
   - Enable farmers to take timely interventions based on the application's feedback, such as targeted pesticide usage or adjustments to crop management techniques, to mitigate disease spread and minimize crop losses.

6.2 Dataset Collection

For our study on potato crop disease detection using ResNet9, we meticulously curated a comprehensive dataset comprising diverse scenarios encoun**tered in potato cultivation. This dataset encompasses 7,134 samples, including 2,436 images of healthy potato crops, 2,659 images exhibiting symptoms of early blight, and 2,660 images showcasing signs of late blight**. Each image was thoughtfully selected to represent various disease manifestations and environmental conditions commonly found in agricultural settings. The extensive dataset serves as a robust foundation for training and evaluating the ResNet9 model, enabling accurate disease detection across different stages of potato growth and disease progression.

6.3 Model Architecture

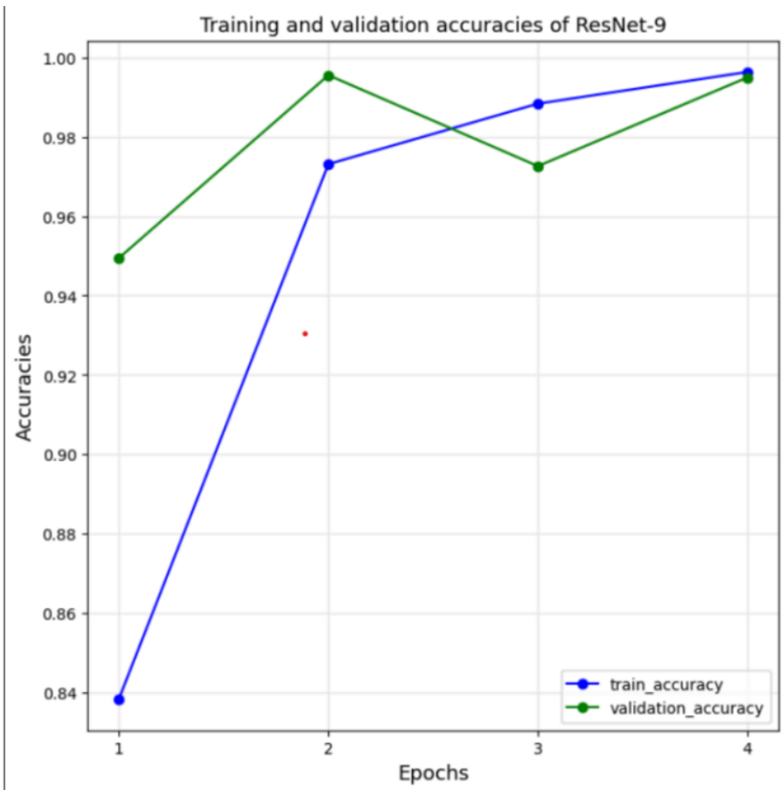
The model architecture chosen for potato crop disease detection is ResNet9, renowned for its depth and effectiveness in image recognition tasks. ResNet9 differs from traditional neural networks due to its unique approach, featuring residual blocks and direct connections between layers. These characteristics contribute to the model's robustness and accuracy in identifying complex patterns in images. Leveraging ResNet9's capabilities, including its ability to handle deep networks effectively, we aim to develop a system capable of accurately detecting various potato leaf diseases. Through the utilization of residual blocks, the model is expected to excel in detecting disease patterns, empowering farmers with early intervention opportunities to minimize crop losses.

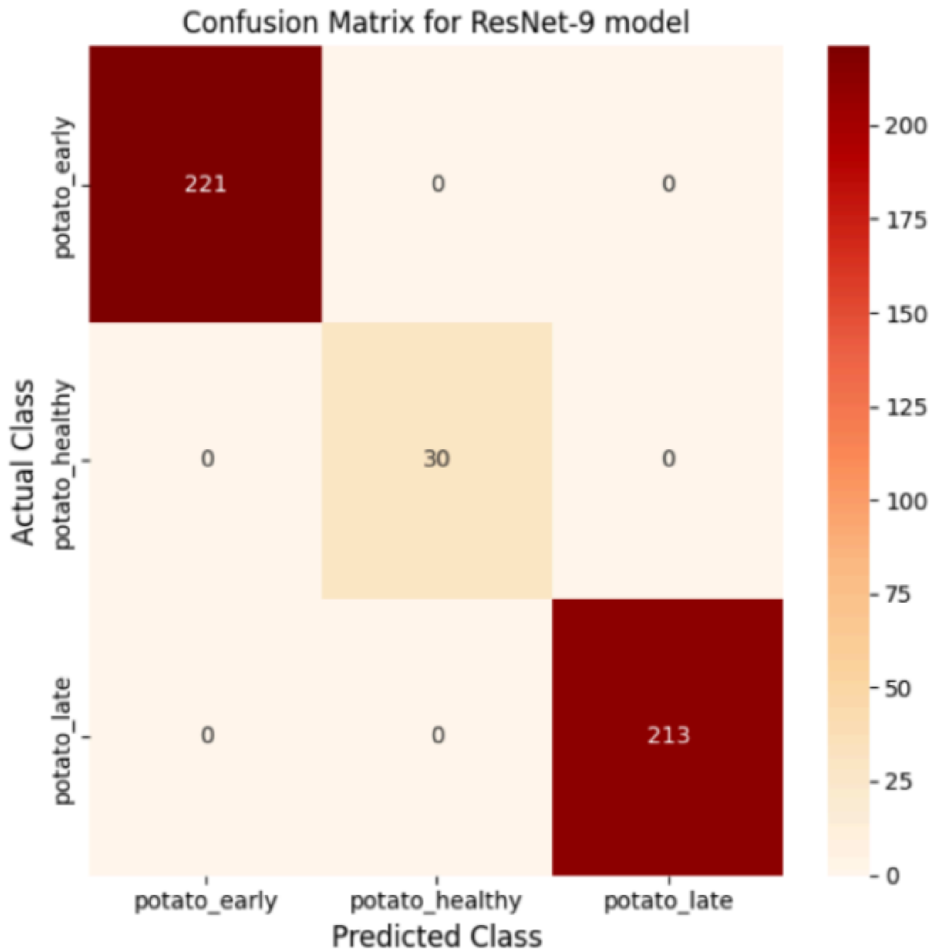


6.4 Training Procedure

1. Data Preparation: The dataset is split into training, validation, and testing sets. The images are preprocessed by resizing them to 256x256 pixels and normalizing the pixel values.
2. Model Training: The model is trained using the Adam optimizer with a learning rate of 0.0001 and a batch size of 32. The categorical cross-entropy loss function is used for multi-class classification. The model is trained for 4 epochs, with early stopping applied if the validation loss does not improve for five consecutive epochs.
3. Data Augmentation: Data augmentation techniques are applied during training, including random horizontal and vertical flips, rotations, and shifts. This helps to increase the diversity of the training data and reduce overfitting.
4. Model Evaluation: The model is evaluated on the testing set using the accuracy and F1 score metrics.

6.5 Model Evaluation






```
Classification Report :
                 precision    recall  f1-score   support

   potato_early       1.00      1.00      1.00       221
 potato_healthy       1.00      1.00      1.00        30
    potato_late       1.00      1.00      1.00       213

       accuracy                           1.00       464
      macro avg       1.00      1.00      1.00       464
   weighted avg       1.00      1.00      1.00       464
```

Chapter 7
Software Developement : Web Application

7.1 User Interface

It includes the following components:

- Upload button: Allows users to select an image file from their local device.
- Submit button: Initiates the process of sending the selected image to the server for prediction.
- Prediction display area: Displays the prediction results received from the server.
- Uploaded image display area: Shows the image uploaded by the user for reference.
- Disease Prediction and Recommendation display area: Provides information about the disease, how to cure it, and recommendations.

7.2 Functionality

- Image Upload: Users can upload an image of the potato leaf using the provided upload button. This image serves as input for the prediction model.
- Prediction Submission: Upon clicking the submit button, the selected image is sent to the server for processing and prediction.
- Prediction Display: The prediction results received from the server are displayed in the prediction display area. These results include information about the presence of early or late blight disease.
- Image Display: The uploaded image is displayed in the designated area for user reference and verification.

7.3 **Integration with Server**

- Interaction with Server-side Component: The web application interacts with the server-side component to perform image processing and prediction tasks.
- Submission Process: Upon submitting an image, the application sends a request to the server, which processes the image using a pre-trained machine learning model.
- Result Display: Once the prediction is made, the server sends back the results to the web application, which then displays them to the user.
- Image Display: The uploaded image is displayed in the designated area for user reference and verification.

7.4 **Code**

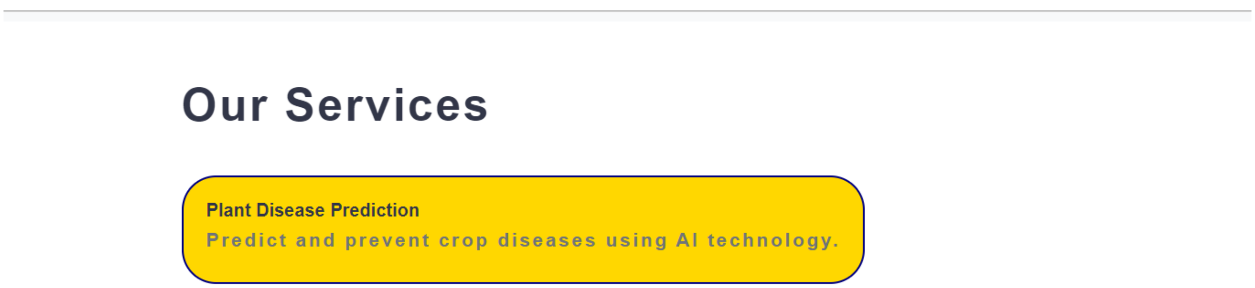
app.py

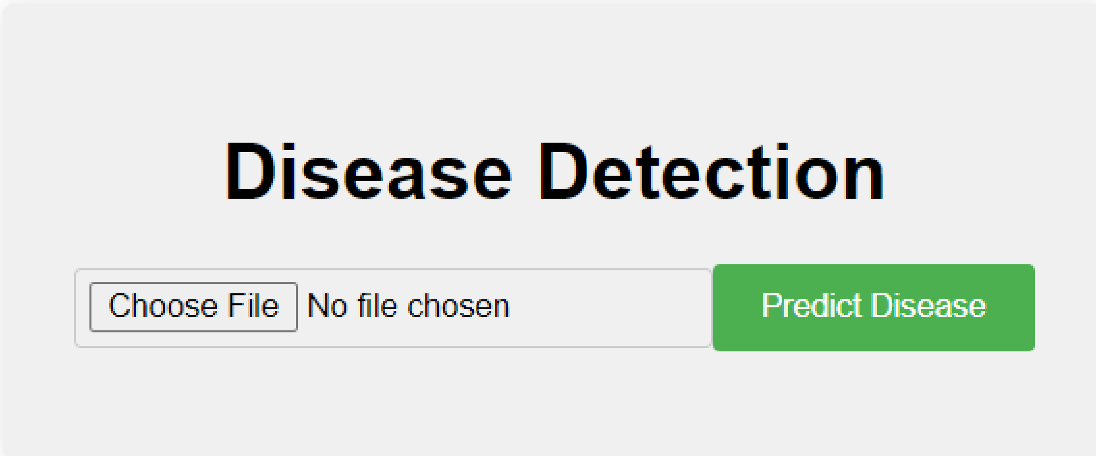
```
from markupsafe import Markup

from flask import Flask, render_template, request

import numpy as np

import pandas as pd

from utils.disease import disease_dic

import requests

import config

import pickle

import io

import torch

from torchvision import transforms

from PIL import Image

from utils.model import ResNet9

disease_classes = [

                'Potato___Early_blight',

                'Potato___Late_blight',

                'Potato___healthy',

                ]

disease_model_path = 'models/resnet9-mdlsd.pth'

disease_model = ResNet9(3, len(disease_classes))

disease_model.load_state_dict(torch.load(
```

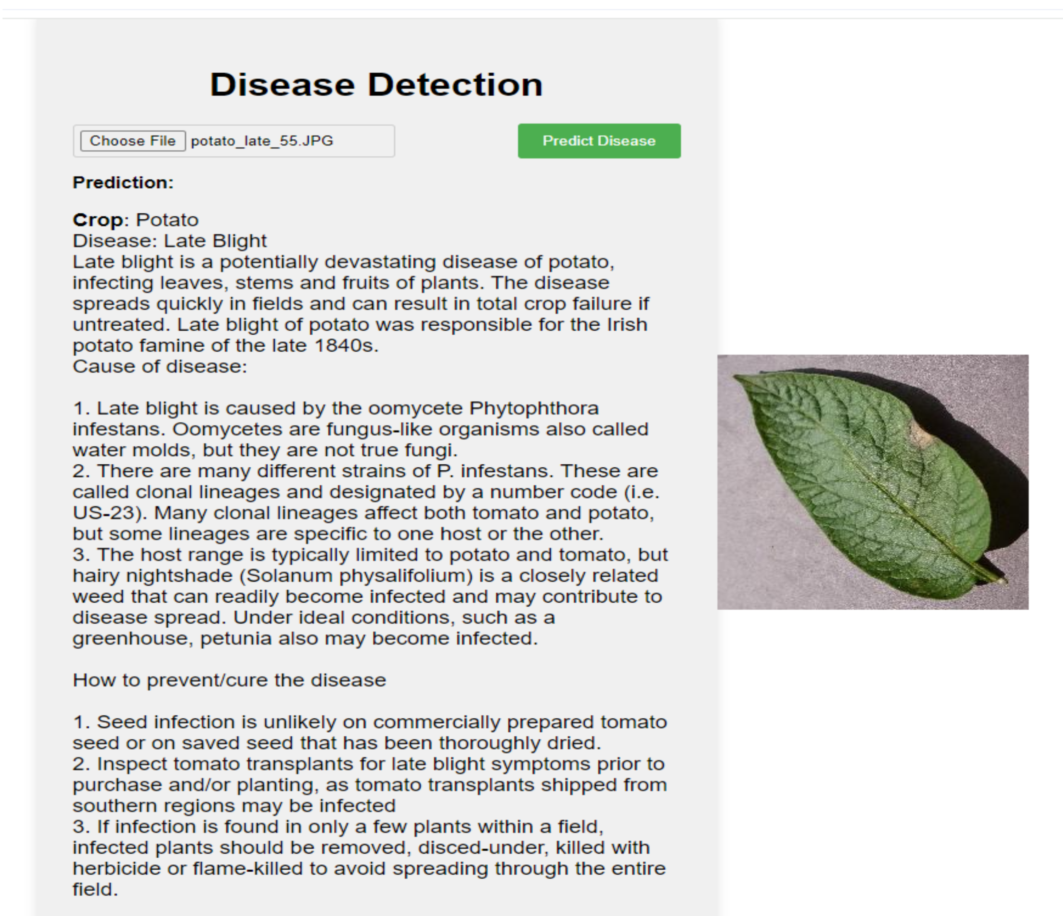
```
    disease_model_path, map_location=torch.device('cpu')))
disease_model.eval()
def predict_image(img, model=disease_model):


    transform = transforms.Compose([
        transforms.Resize(256),
        transforms.ToTensor(),
    ])
    image = Image.open(io.BytesIO(img))
    img_t = transform(image)
    img_u = torch.unsqueeze(img_t, 0)



    yb = model(img_u)
    _, preds = torch.max(yb, dim=1)
    prediction = disease_classes[preds[0].item()]
    return prediction


app = Flask(__name__)
@ app.route('/')
def home():
    title = 'LeafGuard- Home'
    return render_template('index.html', title=title)
```

```
@app.route('/disease-predict', methods=['GET','POST'])
def disease_prediction():
    title = 'LeafGuard - Disease Detection'
    if request.method == 'POST':
        print(request.files)
        file = request.files['files']
        # if not file:
        #     return render_template('disease.html', title=title)
        try:
            img = file.read()
            print("the image is:-",img)
            prediction = predict_image(img)
            print("the prediction is:-",prediction)

            prediction = Markup(str(disease_dic[prediction]))
            return render_template('disease-result.html', prediction=prediction, title=title)
        except:
            pass
    return render_template('disease.html', title=title)
if __name__ == '__main__':
    app.run(debug=False)
```

User Interface Screenshot(GUI)



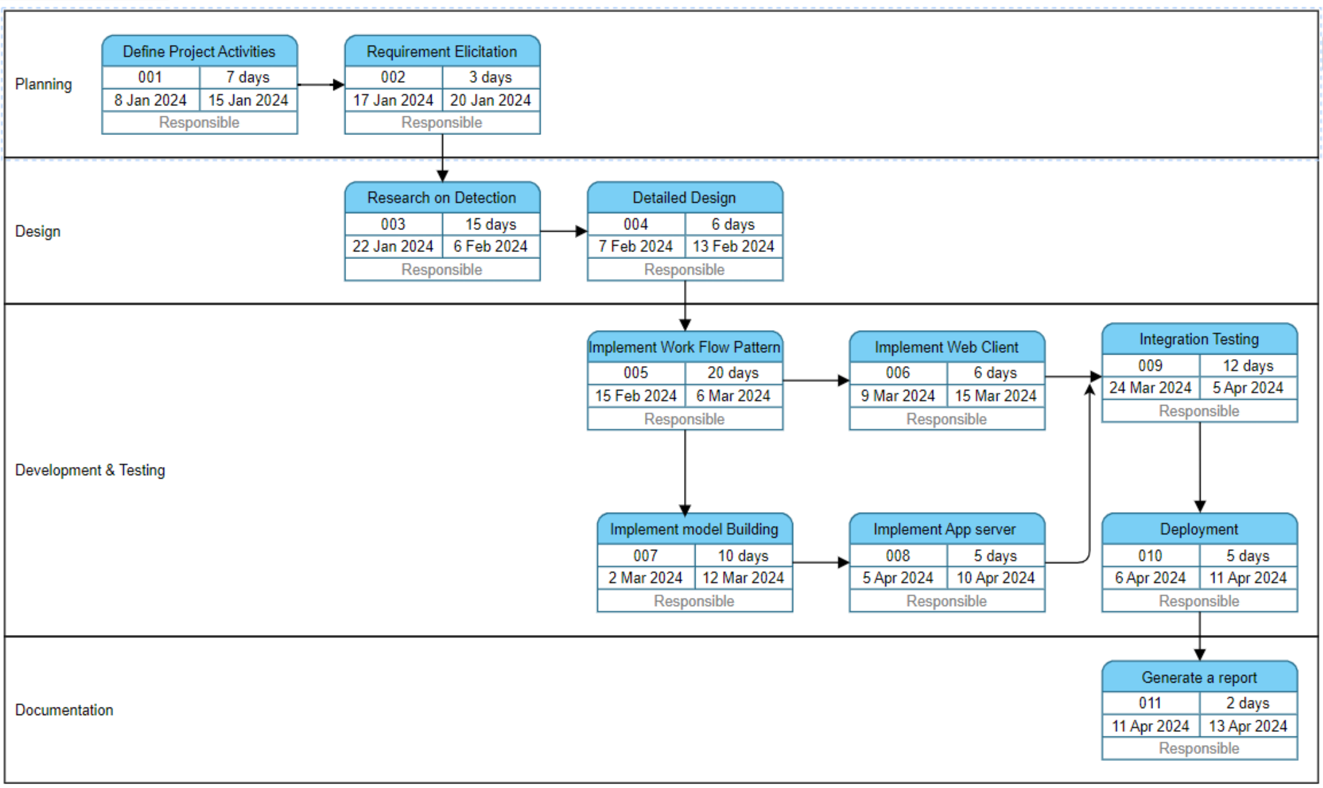

Disease Detection
Choose File No file chosen
Predict Disease


Disease Detection
Choose File potato_late_55.JPG
Predict Disease
Prediction:
Crop: Potato
Disease: Late Blight
Late blight is a potentially devastating disease of potato, infecting leaves, stems and fruits of plants. The disease spreads quickly in fields and can result in total crop failure if untreated. Late blight of potato was responsible for the Irish potato famine of the late 1840s.
Cause of disease:
1. Late blight is caused by the oomycete Phytophthora infestans. Oomycetes are fungus-like organisms also called water molds, but they are not true fungi.
2. There are many different strains of P. infestans. These are called clonal lineages and designated by a number code (i.e. US-23). Many clonal lineages affect both tomato and potato, but some lineages are specific to one host or the other.
3. The host range is typically limited to potato and tomato, but hairy nightshade (Solanum physalifolium) is a closely related weed that can readily become infected and may contribute to disease spread. Under ideal conditions, such as a greenhouse, petunia also may become infected.
How to prevent/cure the disease
1. Seed infection is unlikely on commercially prepared tomato seed or on saved seed that has been thoroughly dried.
2. Inspect tomato transplants for late blight symptoms prior to purchase and/or planting, as tomato transplants shipped from southern regions may be infected
3. If infection is found in only a few plants within a field, infected plants should be removed, disced-under, killed with herbicide or flame-killed to avoid spreading through the entire field.

Chapter 8
Project Planning

8.1 Pert Chart :

A PERT chart, which stands for Program Evaluation and Review Technique, is a project management tool used to plan and schedule tasks within a project. It visually represents the tasks involved in a project, their dependencies, and the estimated time required to complete each task. PERT charts help project managers and teams understand the sequence of activities, identify the critical path (the longest path through the project), allocate resources effectively, and estimate project completion time. PERT charts use nodes to represent tasks and arrows to represent dependencies between tasks, allowing for a clear visualization of the project timeline and workflow.



8.2 Cost estimation

The software cost estimation provides:

- The vital link between the general concepts and techniques of economic analysis and the particular world of software engineering.
- Software cost estimation techniques also provide an essential part of the foundation for good software management.

The cost in a project is due to:

- The requirements for software, hardware and human resources
- The cost of software development is due to the human resources needed
- Most cost estimates are measured in person-months (PM)
- The cost of the project depends on the nature and characteristics of the project .
- At any point, the accuracy of the estimate will depend on the amount of reliable information we have about the final product.

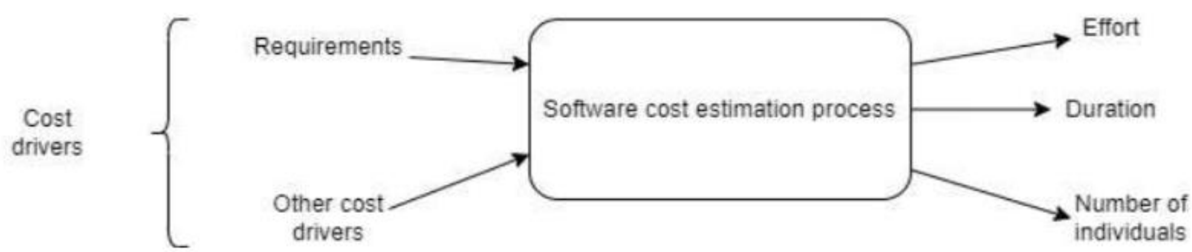


Fig . software estimation process view

The COCOMO model is a popular software estimation model that helps predict the effort and time required to complete a project based on its size and complexity. The text you sent me talks about factors that influence software development costs in general, such as human resources and project characteristics.

The text in the image is about the COCOMO models for software cost estimation. It lists three types of COCOMO models:

1. Basic/Organic
2. Intermediate/Semidetached
3. Detailed model/Detached

The most important factor contributing to a project's duration and cost is the Development Mode. There are 3 types of development modes:

- Organic Mode: The project is developed in a familiar, stable environment, and the product is similar to previously developed products. The product is relatively small, and requires little innovation.

- Semi Detached Mode: The project's characteristics are intermediate between Organic and Embedded.

- Embedded Mode: The project is characterized by tight, inflexible constraints and interface requirements. An embedded mode project will require a great deal of innovation.

The development mode of their project is Organic mode because of following reasons .

1. Project is small and simple.
2. Project team is small (3 members) with prior experience.
3. The problem is well understood and has been solved in the past.

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi - detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Table. The constant values a,b,c and d for the Basic Model for different categories of system.

BASIC SEMI-DETACHED MODEL

E = a * (KLOC)^b
DT = c x (Effort)d
P = effort/time

Where,

E: Effort applied in person-months
DT: Development time in months
P: Total number of persons required to complete the project

E = 3.0 * (5.5)^1.12 = 21 person-months (approx.)
DT = 2.5 * (21) ^ 0.35 = 18 months (approx.)
P = E / DT = 3 people

Chapter 9
Testing

To conduct the testing of the Potato Leaf Disease Detection website, manual testing was employed. Various test cases were developed to thoroughly evaluate the website's functionality. Manual testing proves to be a practical and efficient choice for small-scale projects like the Potato Leaf Disease Detection website. In such endeavors, where resources are often limited and the focus is specific, manual testing offers distinct advantages over automated testing methods. Its flexibility, adaptability, and human-centric approach align seamlessly with the needs of small teams striving for high-quality outcomes within constrained environments.

Manual testing offers several compelling benefits tailored to the dynamics of small projects. Firstly, its cost-effectiveness is noteworthy, as it requires minimal investment in tools and infrastructure. This aspect is particularly advantageous for projects operating within tight budgetary constraints. Additionally, the flexibility inherent in manual testing allows testers to promptly respond to changing requirements and evolving user needs. Unlike automated tests, which may necessitate extensive reconfiguration, manual tests can be easily adjusted as needed, enabling seamless adaptation throughout the project lifecycle.

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC_01	Testing a Healthy Potato Leaf	1. Navigate to LeafGuard Website 2. Click on Disease Detection in Our Services Section 3. Click on Choose File 4. Select the image to be tested 5. Click on Open 6. Click on Submit		The predicted label is : Healthy	The predicted label is: Healthy	Pass
TC_02	Testing a Late_Blight Potato Leaf	1. Navigate to LeafGuard Website 2. Click on Disease Detection in Our Services Section 3. Click on Choose File 4. Select the image to be tested 5. Click on Open 6. Click on Submit		The predicted label is : Late_Blight	The predicted label is : Late Blight	Pass
TC_03	Testing a Early_Blight Potato Leaf	1. Navigate to LeafGuard Website 2. Click on Disease Detection in Our Services Section 3. Click on Choose File 4. Select the image to be tested 5. Click on Open 6. Click on Submit		The predicted label is : Early_Blight	The predicted label is : Early Blight	Pass
TC_04	Testing Button for naviagtion to prediction page	1. Navigate to LeafGuard Website 2. Click on Disease Detection in Our Services Section	-	Redirected to Prediction Page	User is redirected to Prediction Page	Pass
TC_05	Testing files other than jpg	1. Navigate to LeafGuard Website 2. Click on Disease Detection in Our Services Section 3. Click on Choose File 4. Select the image to be tested 5. Click on Open 6. Click on Submit	Sample Text File	Error occurred while processing the image.	Error occurred while processing the image.	Pass
TC_06	Testing empty file	1. Navigate to LeafGuard Website 2. Click on Disease Detection in Our Services Section 3. Click on Choose File 4. Select the image to be tested 5. Click on Open 6. Click on Submit	-	Error occurred while processing the image. Please try again.	Error occurred while processing the image. Please try again.	Pass

Chapter 10
Conclusion and Future Work

10.1 Conclusion:

The development of potato leaf disease detection systems holds significant promise for agricultural management and food security. Through the utilization of machine learning algorithms, image processing techniques, and IoT devices, researchers have made substantial progress in automating the identification and classification of various potato leaf diseases. These advancements not only enhance the efficiency of disease monitoring but also enable timely intervention, leading to improved crop yield and quality.

10.2 Future Work:

1. **Enhanced Accuracy:** Continual refinement of machine learning models to improve accuracy and robustness in detecting and classifying various types of potato leaf diseases.
2. **Multimodal Fusion:** Integration of multiple data sources, such as hyperspectral imaging and drone-based monitoring, to provide comprehensive insights into plant health and disease progression.
3. **Real-Time Monitoring:** Development of real-time monitoring systems that leverage IoT devices and edge computing to provide instant feedback to farmers, enabling proactive disease management strategies.
4. **Data Augmentation Techniques:** Exploration of advanced data augmentation techniques to mitigate issues related to limited training data, thereby enhancing the generalization capability of models.
5. **Field Deployment and Validation:** Extensive field trials and validation studies to assess the practicality and effectiveness of potato leaf disease detection systems under real-world conditions across diverse geographical regions.
6. **User-Friendly Interfaces:** Designing user-friendly interfaces and mobile applications that facilitate easy adoption of disease detection technologies by farmers and agricultural practitioners.

Chapter 11
References and bibliography

[1] Predicting and Classifying Potato Leaf Disease using K-means Segmentation Techniques and Deep Learning Networks - ScienceDirect

[2] Recognition of early blight and late blight diseases on potato leaves based on graph cut segmentation - ScienceDirect

[3] PLDPNet: End-to-end hybrid deep learning framework for potato leaf disease prediction - ScienceDirect

[4]https://www.researchgate.net/publication/347449056_Potato_Leaf_Disease_Classification_Using_Deep_Learning_A pproach

[5] IRJET-V7I11227

[6] Application of Transfer Learning to Detect Potato Disease from Leaf Image (researchgate.net)

[7]https://www.researchgate.net/publication/351505791_Plant_disease_detection_using_hybrid_model_based_on_conv olutional_autoencoder_and_convolutional_neural_network

[8]Automated recognition of optical image based potato leaf blight diseases using deep learning - ScienceDirect

[9]Automated recognition of optical image based potato leaf blight diseases using deep learning - ScienceDirect

[10]Deep neural network features fusion and selection based on PLS regression with an application for crops diseases classification - ScienceDirect

[11]Potato diseases detection and classification using deep learning methods | Multimedia Tools and Applications (springer.com)

[12]High Efficiency Disease Detection for Potato Leaf with Convolutional Neural Network | SN Computer Science (springer.com)

[13]Plant diseases and pests detection based on deep learning: a review | Plant Methods | Full Text (biomedcentral.com)

[14]https://link.springer.com/article/10.1007/s11042-022-13 053-1

[15]https://plantmethods.biomedcentral.com/articles/10.118 6/s13007-021-00722-9

[16] https://www.mdpi.com/2077-0472/12/8/1192

[17]https://www.sciencedirect.com/science/article/abs/pii/S1 878818123001275

[18]https://www.sciencedirect.com/science/article/pii/S2772 375523000084

[19]https://www.sciencedirect.com/science/article/abs/pii/S0 88557652100182X

[20]https://www.sciencedirect.com/science/article/abs/pii/S1 574954121000807

[21]https://www.sciencedirect.com/science/article/abs/pii/S0 957417423012198