A Project Report on

Agrolife: ML Enabled Plant Disease Classification System

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Anagha Rai(19104030)
Ishika Sharma(19104061)
Ruta Mhaskar(19104013)

Under the Guidance of

Dr. Sameer S. Nanivadekar
Prof. Sonal Jain
Prof. Charul Singh

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2022-2023

Approval Sheet

This Project Report entitled ***"Agrolife: ML Enabled Plant Disease Classification System"*** Submitted by ***"Anagha Rai"(19104030), "Ishika Sharma"(19104061), "Ruta Mhaskar"(19104013)*** is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Sonal Jain
Co-Guide

Dr. Sameer S. Nanivadekar
Guide

Prof. Charul Singh
Co-Guide

Dr. Kiran Deshpande
Head Department of Information Technology

Place: A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Agrolife: ML Enabled Plant Disease Classification System"*** submitted by ***"Anagha Rai" (19104030), "Ishika Sharma" (19104061), "Ruta Mhaskar" (19104013)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2022-2023.

Prof. Sonal Jain
Co-Guide

Dr. Sameer S. Navivadekar
Guide

Prof. Charul Singh
Co-Guide

Dr. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place: A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

I have great pleasure in presenting the report on **Agrolife: ML Enabled Plant Disease Classification System.** I take this opportunity to express my sincere thanks towards my guide **Dr. Sameer S. Nanivadekar**, Co-Guide **Prof. Sonal Jain** & **Prof. Charul Singh** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. I would like to express my gratitude towards their constant encouragement, support and guidance through the development of project.

I thank **Dr. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

I thank **Prof. Sonal Jain** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

I also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. I wish to express my deep gratitude towards all our colleagues of APSIT for their encouragement.

Anagha Rai
(19104030)

Ishika Sharma
(19104061)

Ruta Mhaskar
(19104013)

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Anagha Rai(19104030)

Ishika Sharma(19104061)

Ruta Mhaskar(19104013)

Date:

Abstract

In growing nations like India, agriculture plays a huge role, but there are still major concerns about food security. Plant diseases are the primary cause of squandered crops and their early detection is crucial since they have an impact on how much food is produced. This project proposes a system to visualize different plant illnesses and offer appropriate advice for curing these problems. Our goal is to determine whether the plant is sick or not and if it is, to identify the type of illness that caused it. Along with this, the system gives a general overview of the illness, suggests a few straightforward home cures (if necessary), and directs the user toward getting the appropriate consultation. Convolutional Neural Network (CNN), a deep learning model is employed to recognize and categorize plant leaf diseases, to display the type of ailment as well as potential treatments after receiving input from the user in the form of picture data. Other technologies utilized include Google Cloud and TensorFlow which is being used for data preprocessing. Applying the CNN algorithm, it is investigated that an average classification accuracy of 91% is being achieved to perform the classification of the leaf diseases. Also by utilizing machine learning algorithms, the system facilitates farmers in identifying the optimal crop for their farmland based on inputs such as soil type and other relevant factors, as well as recommending suitable fertilizers. This would ultimately enhance productivity and diminish crop losses.

Index Terms—Plant Disease, TensorFlow, Convolution Neural Network (CNN), Deep Learning, Machine Learning, Fertilizer Recommendation, Crop Recommendation, Farming, Agriculture.

Contents

List of Figures

List of Abbreviations

ML:	Machine Learning
DL:	Deep Learning
CNN:	Convolutional Neural Network
SVM:	Support Vector Machine
ReLU:	Rectified Linear Unit
LVQ:	Learning Vector Quantization
NPK:	Nitrogen, Phosphorus, Potassium

Chapter 1

Introduction

Farming produces the majority of the world's food and textiles. More than 15 percent of crops in India are lost due to disease, making it one of the most pressing issues to address. Thus in this research paper, we present a novel plant leaf disease classification system using CNNs. Our system is trained on a large dataset of plant leaf images and employs advanced deep-learning techniques for feature extraction and classification. The aim of this study is to demonstrate the effectiveness and potential of using CNNs in a plant disease classification and to contribute to the development of more reliable and efficient solutions for crop management The ability to produce enough food to meet the needs of more then 7 billion people are still preserved by modern technology. However, a number of factors, such as temperature change, the decline in pollinators, plant diseases, and others continue to threaten food security[1]. The Deep Learning (DL) approach is a subcategory of Machine Learning (ML), was introduced in 1943 when threshold logic was introduced to build a computer model closely resembling the biological pathways of humans[2]. Also, many developed/modified DL architectures are implemented along with several visualization techniques like the box plots, histograms, charts, heat maps, etc. to detect and classify the symptoms of plant diseases[4]. DL has been widely used in the image and video processing, voice processing, and natural language processing. Along with this, it has also become a research hotspot in the field of agricultural plant protection, such as plant disease recognition and pest range assessment, etc. The application of deep learning in plant disease recognition can avoid the disadvantages caused by artificial selection of disease spot features, also make plant disease feature extraction more objective, and improve the efficiency of research and technology transformation speed[6]. In addition to plant leaf disease classification, our project also includes crop and fertilization recommendation modules utilizing machine learning algorithms. These modules are designed to assist farmers in identifying the most suitable crops and fertilizers based on various factors such as soil type, climate, and crop yield, ultimately leading to improved crop management and increased productivity.

Chapter 2

Literature Review

The goal of the literature review is to understand the knowledge that is already available on the Plant Disease Classification System. The study of the research helped in choosing the best algorithm and feature extraction technique for effective results.

Marzougui, F., Elleuch, M., and Kherallah, M. [1], used pre-trained trained weights as a starting point to avoid a very long treatment. Followed by this, the proposed approach was compared to several artisanal shallow structure approaches based on machine learning. The proposed system achieved promising precision results on the plant leave dataset, demonstrating the effectiveness of its approach for the detection of diseases.

The review by Liu, J., Wang, X. [2] provided a definition of plant diseases and pest detection problems and put forward a comparison with traditional plant diseases and pest detection methods. According to the difference in network structure, this study outlined the research on plant diseases and pests detection based on deep learning in recent years from three aspects of classification network, detection network, and segmentation network, and the advantages and disadvantages of each method were summarized. Common datasets were introduced, and the performance of existing studies was compared. On this basis, possible challenges in practical applications of plant diseases and pest detection based on deep learning were discussed. In addition, many possible solutions and research ideas were proposed for the challenges, and several suggestions were given.

Pandian JA, Kumar VD, Geman O, Hnatiuc M, Arif M, Kanchanadevi K [3], proposed a fourteen-layered 'deep convolutional neural network' (14-DCNN) to detect diseases in plant leaves using leaf images. A new dataset was created using numerous open datasets. 'Data Augmentation techniques were then used to balance out the individual class sizes of the dataset. Three image augmentation techniques were used namely basic image manipulation (BIM), deep convolutional generative adversarial network (DCGAN), and neural style transfer (NST). The dataset consisted of about 147,500 images of 58 different 'healthy' and 'diseased' plant leaf classes along with one 'no-leaf' class. The proposed DCNN model was trained in multi-graphics processing units (MGPUs) environment for exactly 1000 epochs. The random search with the coarse-to-fine searching technique was used to select the best hyper-parameter values in order to improve the training performance of the proposed model.

Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif in [4] provided a comprehensive explanation of Deep Learning models are used to visualize various plant diseases. In addition to this, some research gaps were identified from which to obtain greater transparency for detecting diseases in plants, even before their symptoms appear clearly. The paper also stated that plant diseases affect the growth of their respective species, thus making their early identification very important. Several Machine Learning models had been employed for the detection and classification of plant diseases but, after the advancements in a subdivision of ML i.e Deep Learning (DL), this area of research appeared to have huge potential in terms of increased accuracy. Several developed DL architectures were implemented along with several visualization techniques to detect and classify ailments. Also, several performance metrics were used for the assessment of these architectures or techniques.

Reference [5] dealt with a replacement approach to the development of a disease recognition model supported by leaf image classification, by the utilization of deep convolutional networks. All the essential steps required for implementing this disease recognition model was completely described throughout the paper, starting from gathering images to making a database, assessed by agricultural experts, and a deep learning framework to perform the deep Convolutional Neural Network training. The advance and novelty of the developed model illustrated its simplicity: healthy leaves and background images are in line with other classes, enabling the model to distinguish between diseased leaves and healthy ones, or from the environment by using CNN.

Reference [6] provided the research progress of deep learning technology in the field of crop leaf disease identification in recent years. The application of deep learning in a plant disease recognition can avoid the disadvantages caused by the artificial selection of disease spot features makes plant disease feature extraction more objective as well as improve the research efficiency and technology transformation speed.In this paper, the authors L. Li, S. Zhang, and B. Wang, presented the current trends and challenges for the detection of plant leaf disease using deep learning (DL) and advanced imaging techniques with the hopes that this research will be a valuable asset for those researchers who study the detection of plant diseases and pests. At the same time, they also discussed some of the current challenges and hurdles that need to be resolved.

A. KP and J. Anitha in [7], trained a deep learning model i.e. Convolutional Neural Network (CNN) to classify the different plant diseases. This model was used due to its massive success in image-based classification. The deep learning model provided a faster and more accurate prediction as compared to manual observations of the plant leaf. Here, the CNN model and pre-trained models namely VGG, ResNet, and DenseNet were trained using the dataset. Among them, the DenseNet model was observed to have achieved the highest accuracy.

Sai Reddy, B., Neeraja, S. [8] presented an automatic plant leaf damage detection and disease identification system. The first stage of the proposed method identified the type of disease based on the plant leaf image using a method known as DenseNet.Stage two identified the damage in the leaf using deep learning-based semantic segmentation. Each 'RGB' pixel value combination in the image was extracted followed by supervised training that was performed on the pixel values using the 1D Convolutional Neural Network (CNN). The third stage suggested a remedy for the disease based on the disease type and the damage state observed. This gave an accuracy of 97 %.

Reference [9] proposed a more concise method of separating the crop and disease identification and classifying them independently, and demonstrated that it is more effective than the traditional crop-disease pair approach. Meanwhile, a tri-linear convolutional neural network (T-CNN) model using bi-linear pooling was constructed and used images obtained in a real-world environment for the study of crop disease identification. The crop and disease identification accuracy achieved 99.99 % and 99.7 % on the test set in a controlled laboratory environment, and 84.11 % and 75.58 % on the test set in a real-world environment, respectively.

Geetharamani G., Arun Pandian J.[10] put forth a novel plant leaf disease identification model following the 'deep convolutional neural network' (Deep CNN) concept. Six types of data augmentation methods were used namely: image flipping, gamma correction, noise injection, principal component analysis (PCA) color augmentation, rotation, and scaling. Compared to popular transfer learning approaches, the proposed model achieved better performance when using the validation data. The model achieved 96.46 % classification accuracy.

G. Shrestha, Deepsikha, M. Das, and N. Dey in [11] proposed a 'CNN' based method for plant disease detection. Simulation study and analysis were done on dummy images in terms of 'time complexity' and the 'area of the infected region' using an image processing technique. A total of 15 cases had been fed to the model, out of which 12 cases were of 'diseased' plant leaves which were Bell Paper Bacterial Spot, Potato Early Blight, Potato Late Blight, also Tomato Target Spot, Tomato Mosaic Virus, Tomato Yellow Leaf Curl Virus, Tomato Bacterial Spot, Tomato Early Blight, Tomato Late Blight, Tomato Leaf Mold, Tomato Septoria Leaf Spot and Tomato Spider Mites, and rest 3 cases were of 'healthy' leaves namely, Bell Paper Healthy, Potato Healthy, and Tomato Healthy. Test accuracy was 88.80 %. Different performance matrices were derived for the same.

Reference [12] investigated a potential solution to the problem of disease detection models' performance drop by using segmented image data to train the convolutional neural network (CNN) models. As compared to the F-CNN model trained using full images, the S-CNN model trained using segmented images was more than double in performance and gave up to 98.6 percent accuracy when tested on independent data previously unrevealed by the models even with 10 disease classes. Not only this, by using tomato plant and target spot disease type as an example, it showed that the confidence of self-classification for the S-CNN model improves significantly over the F-CNN model. Hence, this research work brings the applicability of automated methods closer to non-experts for the timely detection of diseases.

In [13], convolutional neural network(CNN) models were developed to perform plant disease detection and identification using simple leaves images of healthy and diseased plants, through deep learning methodologies. Training of the models was performed with the use of an open database of about 87,848 images, containing 25 different plants in a set of 58 distinct classes of [plant, disease] combinations, including healthy plants. Several model architectures were trained, with the best performance reaching a 99.53 percent success rate in identifying the corresponding [plant, disease] combination (or healthy plant). This significantly high success rate makes the model a very useful advisory or early warning tool, and an approach that could be further expanded to support an integrated plant disease identification system in order to operate in real cultivation conditions.

M. Sardogan, A. Tuncer, and Y. Ozen [14] presented a 'Convolutional Neural Network (CNN) model and Learning Vector Quantization (LVQ) algorithm' based methodology for tomato leaf disease detection and categorization. The dataset contained about 500 images of tomato leaves with four symptoms of diseases. CNN for automatic feature extraction and categorization was modeled and 'Color' information was actively used for plant leaf disease research. The filters were applied to three channels based on RGB components. Lastly, the LVQ had been fed with the output feature vector of the convolution part for training the network.

Objectives:

After conducting an in-depth literature survey, we found that CNN is the most feasible methodology to implement a plant leaf disease classification system. Therefore, we implemented a CNN-based methodology for our project, as it yields higher accuracy and more accurate results. In our project, we implemented the objectives identified from the literature survey to develop a plant leaf disease classification system that accurately classifies plant leaves based on their visual features. Thus, to ensure the success of our project, we have implemented the following objectives in our project:

- To detect the crop disease using a Convolutional Neural Network algorithm.
- To give the right knowledge to the users about which disease the crop has used the image classification.
- To help predict disease and display the actions to be taken at the right time.
- To give the right knowledge to the users about which crop to sow and which fertilizer to use in their field based on the soil data using Machine Learning.

Chapter 3

Project Design

3.1 Existing System

- The existing method for plant disease detection is simply naked eye observation by experts through which identification and detection of plant diseases is done.

- For doing so, a large team of experts as well as continuous monitoring of plant is required, which costs very high when we do with large farms.

- At the same time, in some countries, farmers do not have proper facilities or even idea that they can contact to experts. Due to which consulting experts even cost high as well as time consuming too. In such conditions, the suggested technique proves to be beneficial in monitoring large fields of crops.

- Recently, automatic detection of the diseases are done by just seeing the symptoms on the plant leaves which makes it easier as well as cheaper. The existing methods studies are for increasing throughput and reduction subjectiveness which comes due to naked eye observation through which identification and detection of plant diseases is done.

3.2 Proposed system

- This project aims to investigate and identify whether a plant is diseased, and if so, provide the user with a summary of the disease, including potential home remedies, and guidance on seeking appropriate professional advice.

- The deep learning technique utilized in this project is based on convolutional neural networks (CNNs), which receive input from the user and display information about the type of disease, along with a detailed description and potential remedies.

- Ultimately, this system can aid in preventing crop loss and enhancing production growth.

- In addition to identifying plant diseases and providing remedies, the proposed system also incorporates a fertilizer and crop recommendation system. This feature utilizes machine learning algorithms to analyze soil and environmental data and make personalized recommendations for fertilizers and crops based on the specific needs of each user's plant. This functionality can improve crop yields and support sustainable farming practices.

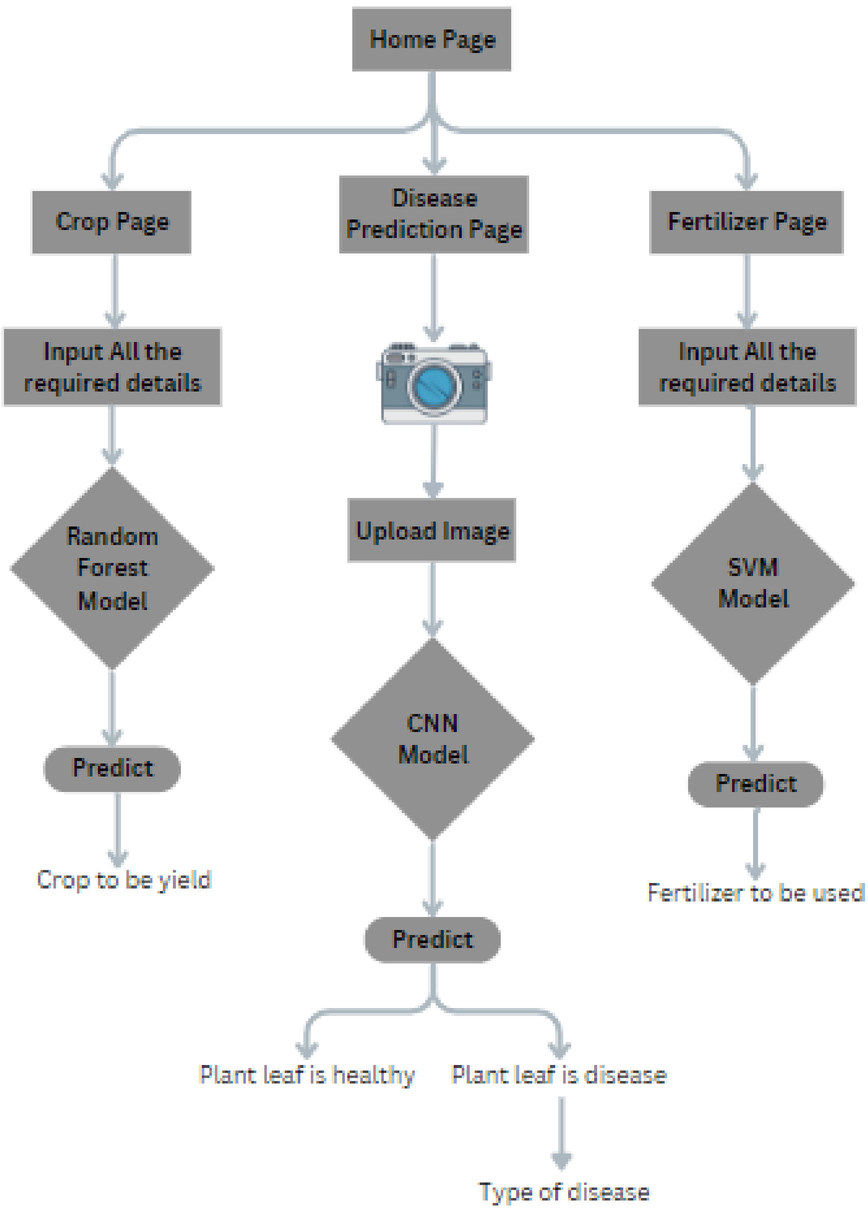


Figure 3.1: System Architecture of our Website

3.3 Mathematical Formulas

CNN Convolutional Neural Network:
Suppose that we have a square neuron layer of size $N \times N$ which is followed by a convolutional layer. If we use an $m \times m$ filter ω, the convolutional layer output will be of size $(N-m+1) \times (N-m+1)$. In order to compute the pre-nonlinearity input to some unit $x_{ij}^{(l)}$ in our layer, we need to add the contributions (weighted by the filter components) from the previous layer cells.

$$x_{ij}^{(l)} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{l-1}(i+a, j+b)$$

This is just a convolution, which we can express in 'Matlab' using `conv2(x, w, 'valid')`. Then, the convolutional layer applies its nonlinearity:

$$y_{ij}^{(l)} = \sigma(x_{ij}^{(l)})$$

Rectified Linear Unit (ReLU):
ReLU is an activation function used in CNNs that introduces non-linearity into the model. It applies the following formula:

$$\text{f(x)=max(0,x)}$$

Pooling Layer:
The pooling layer reduces the spatial dimensionality of the input image by downsampling it. It applies the following formula:

$$\text{f(x)} = \text{max(x)}$$

where x is the input values within the pooling window, and the formula for average pooling is f(x) = mean(x), where x is the input values within the pooling window. The pooling window size and stride length are typically specified as hyperparameters.

Softmax:
The Softmax function is commonly used in the output layer of neural networks for multiclass classification tasks. Given a vector of scores z, the softmax function calculates the probability p of each class as follows:

$$p_j = \frac{e^{z_j}}{\sum_k e^{z_k}}, \text{ for all } k$$

where j is the index of the current class, k is the index of all classes, e is the exponential function, and $\sum_k e^{z_k}$ is the sum of the exponential values over all classes.

Sequential Layer:
A Sequential layer is a type of layer used in neural networks to define a sequence of other layers:

$$y = f(W,b)(x) = f_L(f_{L-1}(\dots f_2(f_1(x; W_1, b_1); W_2, b_2)\dots))$$

Here, x is the input to the layer, y is the output, and f denotes the activation function used in each layer, W and b denote the weights and biases of each layer, and L is the number of layers in the sequence.

3.4 System Diagram

3.4.1 Activity Diagram

An activity diagram is used to model the behavior of a system or process, including the interaction between users, components, and other systems. They help to visualize the entire process and identify any possible flaws or improvements. Following Figure 3.2 is the activity diagram of our system.

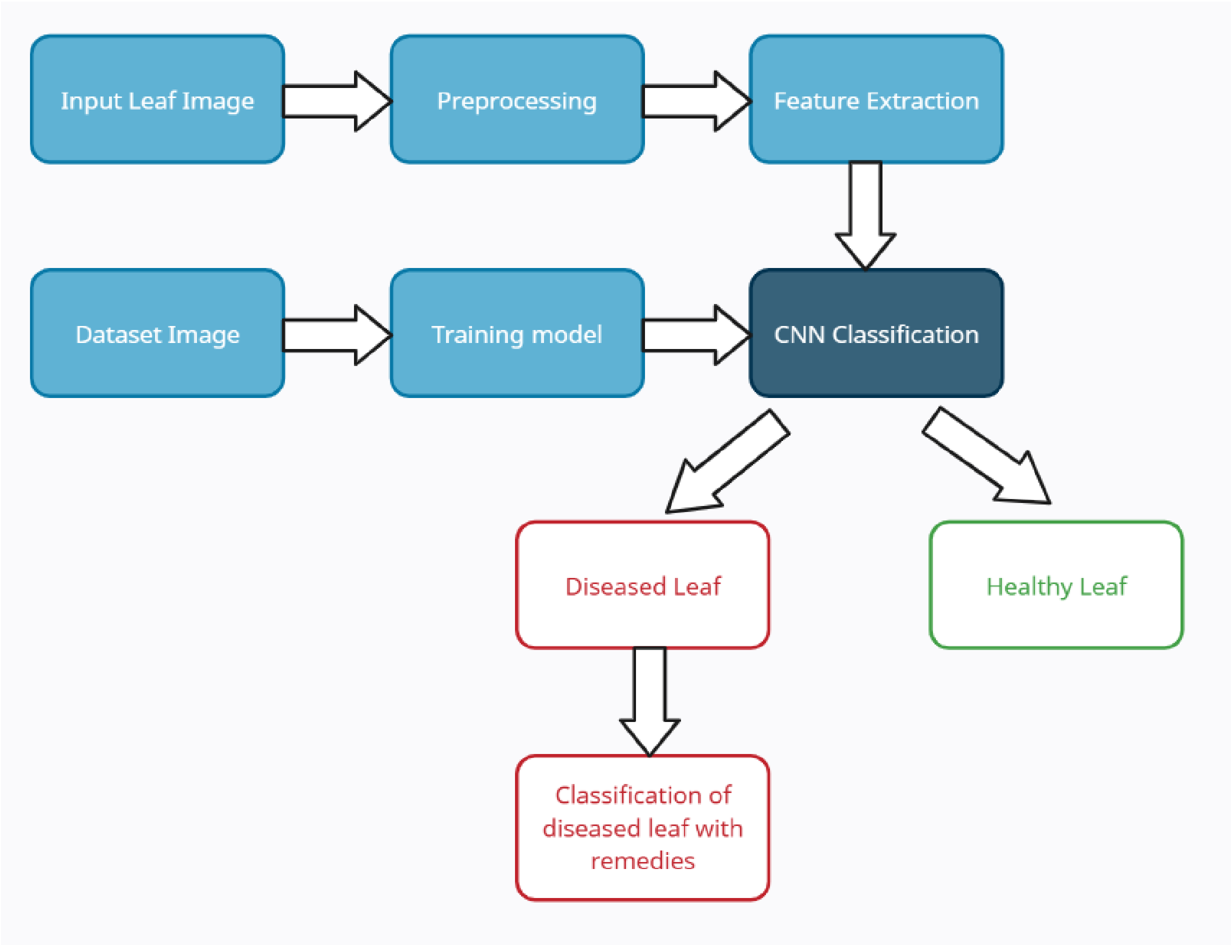


Figure 3.2: Activity Diagram for Plant Disease Classification

As shown in Figure 3.2, The activity diagram begins with the user selecting an image of a plant leaf to classify its health status. The input image is then preprocessed and fed into a CNN model for classification.
Once the image is classified, the system determines whether the plant is diseased or not. If it is, the system provides information about the disease along with possible home remedies to the user.

3.4.2 Use Case Diagram

A use case diagram is a visual representation of the interactions between the system and its users, highlighting the various use cases and scenarios that the system can handle. They provide a clear understanding of the system's functionalities and requirements, helping to identify potential issues and ensuring that the system meets the needs of its users. The use

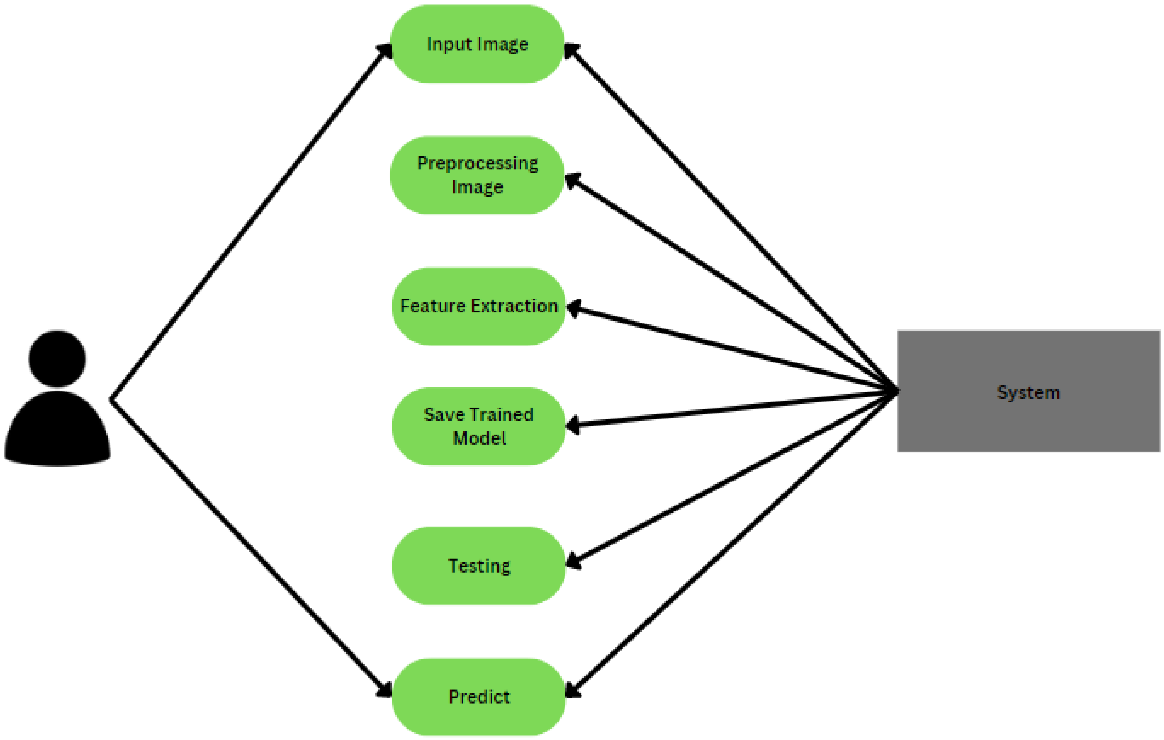


Figure 3.3: Use Case Diagram for Plant Disease Classification

case diagram for a plant leaf disease classification system using deep learning techniques involves several components. The user starts by providing an input image of a plant leaf, which undergoes preprocessing to remove any distortions. Relevant features of the preprocessed image are then extracted using deep learning techniques like CNNs. The system then utilizes a pre-trained model to identify the disease present in the input image. Testing is done to ensure the accurate identification, and the system provides the user with information and remedies about the disease. This use case diagram provides a clear outline of the process involved in plant leaf disease classification using deep learning techniques.

3.4.3 Sequence Diagram

A Sequence diagram is a powerful tool for understanding and modeling the interactions between components in a system in a time-ordered sequence. They help to visualize the flow of messages and events over time and can be used to design and analyze complex systems.

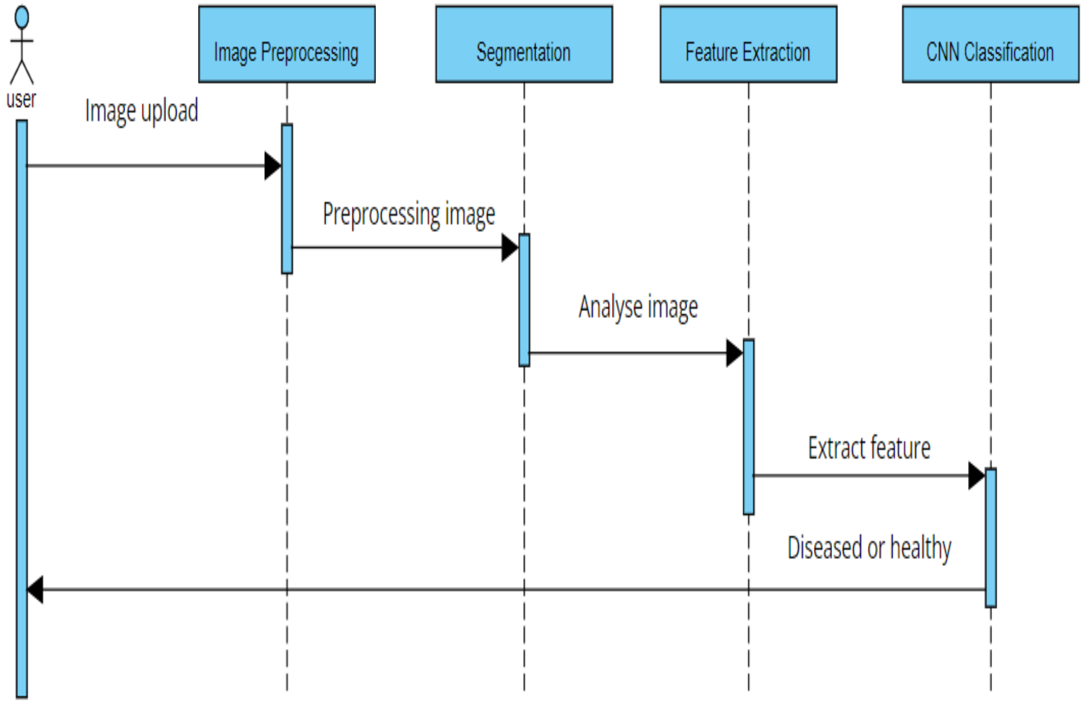


Figure 3.4: Sequence Diagram for Plant Disease Classification

The sequence diagram shown in Figure 3.4 is the interaction between different components of the plant leaf classification system. The user selects an image of a plant leaf, which is then passed on to the image processing module for preprocessing.
Once the image is preprocessed, it is fed into the CNN model for classification. The CNN model processes the image and returns the classification result to the classification module. The classification module then determines whether the plant is diseased or not. If it is, the system provides information about the disease and possible remedies to the user.

Chapter 4

Project Implementation

Project implementation consists of visions and plans with which we are supposed to build the end product. This includes the logical conclusion, after evaluating, deciding, visioning, planning, and finding the other resources for the project. Technical implementation is one of the major aspects of executing a project.

4.1 Module 1: Plant Leaf Disease Classification System

Import Requirements: Import requirements refer to the external libraries, modules, or packages that need to be imported into a project to enable certain functionalities or operations.

```
!pip install torchsummary
```

```
Collecting torchsummary
  Downloading torchsummary-1.5.1-py3-none-any.whl (2.8 kB)
Installing collected packages: torchsummary
Successfully installed torchsummary-1.5.1
```

```
import os                       # for working with files
import numpy as np              # for numerical computationss
import pandas as pd             # for working with dataframes
import torch                    # Pytorch module
import matplotlib.pyplot as plt # for plotting informations on graph and images using tensors
import torch.nn as nn           # for creating  neural networks
from torch.utils.data import DataLoader # for dataloaders
from PIL import Image           # for checking images
import torch.nn.functional as F # for functions for calculating loss
import torchvision.transforms as transforms   # for transforming images into tensors
from torchvision.utils import make_grid       # for data checking
from torchvision.datasets import ImageFolder  # for working with classes and images
from torchsummary import summary              # for getting the summary of our model

%matplotlib inline
```

The following code snippet Counts the number of diseases in different plants considered.

So we have images of leaves of 14 plants and while excluding healthy leaves, we have 26 types of images that show a particular disease in a particular plant.

```
# Number of images for each disease
nums = {}
for disease in diseases:
    nums[disease] = len(os.listdir(train_dir + '/' + disease))

# converting the nums dictionary to pandas dataframe passing index as plant name and number of images as column

img_per_class = pd.DataFrame(nums.values(), index=nums.keys(), columns=["no. of images"])
img_per_class
```

	no. of images
Tomato___Late_blight	1851
Tomato___healthy	1926
Grape___healthy	1692
Orange___Haunglongbing_(Citrus_greening)	2010
Soybean___healthy	2022
Squash___Powdery_mildew	1736
Potato___healthy	1824
Corn_(maize)___Northern_Leaf_Blight	1908
Tomato___Early_blight	1920
Tomato___Septoria_leaf_spot	1745
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	1642
Strawberry___Leaf_scorch	1774
Peach___healthy	1728

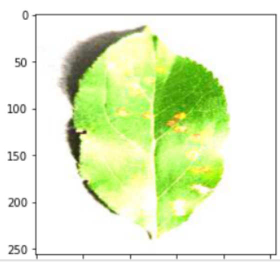
Image prediction: Converts the image to an array and returns the predicted class with the highest probability.

```
def predict_image(img, model):
    """Converts image to array and return the predicted class
        with highest probability"""
    # Convert to a batch of 1
    xb = to_device(img.unsqueeze(0), device)
    # Get predictions from model
    yb = model(xb)
    # Pick index with highest probability
    _, preds  = torch.max(yb, dim=1)
    # Retrieve the class label

    return train.classes[preds[0].item()]
```

```
# predicting first image
img, label = test[0]
plt.imshow(img.permute(1, 2, 0))
print('Label:', test_images[0], ', Predicted:', predict_image(img, model))
```

```
Label: AppleCedarRust1.JPG , Predicted: Apple___Cedar_apple_rust
```



4.2 Module 2: Crop Recommendation System

Import Libraries: The process of including external pre-written code modules that provide additional functionalities to the project, allowing developers to save time and focus on implementing the core logic.

```
# Importing libraries

from __future__ import print_function
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn import tree
import warnings
warnings.filterwarnings('ignore')
```

```
df = pd.read_csv('../Data-processed/crop-recommendation.csv')
```

```
df.head()
```

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

Predict the suitable crop based on input.

```
data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
prediction = RF.predict(data)
print(prediction)
```

```
['coffee']
```

```
data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])
prediction = RF.predict(data)
print(prediction)
```

```
['jute']
```

4.3 Module 3: Fertilizer Recommendation System

Import Libraries: The act of adding pre-written code modules from outside sources that give the project extra functionality while freeing up developers' time to construct the project's main logic.

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import imblearn
from imblearn.over_sampling import SMOTE
from collections import Counter
```

+ Code + Markdown

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split
```

```
from sklearn.neighbors import  KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
import xgboost
from xgboost import XGBClassifier
```

```
from sklearn.metrics import accuracy_score, confusion_matrix
import pickle
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

Predict the suitable Fertilizer.

```
DATA_PATH = "Fertilizer Prediction.csv"
```

```
data = pd.read_csv("Fertilizer Prediction.csv")
data.head(10)
```

	Temparature	Humidity	Moisture	Soil Type	Crop Type	Nitrogen	Potassium	Phosphorous	Fertilizer Name
0	26	52	38	Sandy	Maize	37	0	0	Urea
1	29	52	45	Loamy	Sugarcane	12	0	36	DAP
2	34	65	62	Black	Cotton	7	9	30	14-35-14
3	32	62	34	Red	Tobacco	22	0	20	28-28
4	28	54	46	Clayey	Paddy	35	0	0	Urea
5	26	52	35	Sandy	Barley	12	10	13	17-17-17
6	25	50	64	Red	Cotton	9	0	10	20-20
7	33	64	50	Loamy	Wheat	41	0	0	Urea
8	30	60	42	Sandy	Millets	21	0	18	28-28
9	29	58	33	Black	Oil seeds	9	7	30	14-35-14

Exploring the dataset

```
data["Fertilizer Name"].unique()
```

```
array(['Urea', 'DAP', '14-35-14', '28-28', '17-17-17', '20-20',
       '10-26-26'], dtype=object)
```

Save the model.

```
#save the model
pickle.dump(svm_pipeline, open("svm_pipeline.pkl", "wb"))
```

```
loaded_model = pickle.load(open("svm_pipeline.pkl", 'rb'))
sipt = np.array([[36, 60, 43,  4,  4, 15,  0, 41]])
predd = loaded_model.predict(sipt)
predicted_Fertizer=predd[0]
print((fertname_dict[predicted_Fertizer]))
```

```
DAP
```

Chapter 5

Testing

5.1 Software Testing

Software testing is the process of verifying and validating software to ensure that it meets the specified requirements and works as expected. It involves different types of testing such as functional testing, performance testing, security testing, and usability testing. Selenium is a popular open-source testing tool that is widely used for web application testing. It provides a suite of tools for automating web browsers and is used for testing web-based applications across different platforms and browsers. Selenium supports different programming languages such as Java, Python, C#, and Ruby, making it a flexible and versatile tool for testing.
For performing software testing on our system we have used the Selenium tool. Selenium is primarily used for functional testing, where it automates user interactions with the web application, simulating real user behavior. To perform software testing on a plant leaf disease classification system project using Selenium, one needs to set up the testing environment, create test cases, automate the tests, verify the results, and generate reports. Efficient software testing using Selenium can enhance the quality of the software, decrease the number of errors and issues, and ultimately increase customer satisfaction. A detailed description of our software testing is shown in Table 5.1.

Table 5.1: Software Testing for Plant Leaf Classification System

Test Case	Test Description	Expected Outcome	Actual Outcome	Pass/Fail
TC-01	Upload image	User uploads an image of a plant leaf	Image is successfully uploaded	Pass
TC-02	Preprocessing	Preprocessing module processes the image	Preprocessed image is displayed	Pass
TC-03	Classification	CNN model classifies the preprocessed image	Correct disease classification is displayed	Pass
TC-04	Disease information	System provides information about the disease	Accurate disease information is displayed	Pass
TC-05	Remedy information	System provides remedies for the disease	Appropriate remedies are displayed	Pass
TC-06	Crop recommendation	System recommends a crop based on location and soil properties	Recommended crop is suitable for the location and soil type	Pass
TC-07	Fertilizer recommendation	System recommends a fertilizer based on the recommended crop	Recommended fertilizer is suitable for the recommended crop	Pass
TC-08	Usability	System is user-friendly	User can easily navigate through the system	Pass

Chapter 6

Result

The images presented below are the outputs of our system.

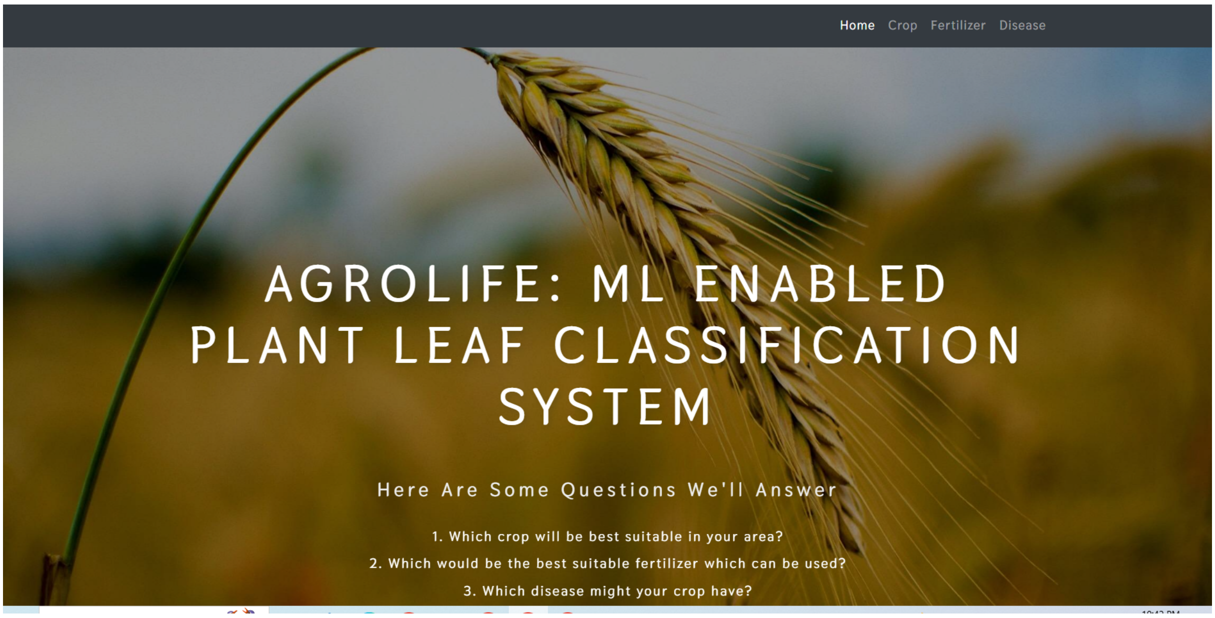


Figure 6.1: Home Page of Plant Leaf Disease Classification System

The Homepage of the plant disease detection system is shown in Fig 6.1. It consists of the features and the facilities offered by the system. From the home page, we can navigate to the other two pages. This is where the user lands when they first visit the website. Overall, the homepage serves as the introductory point for users to familiarize themselves with the system.

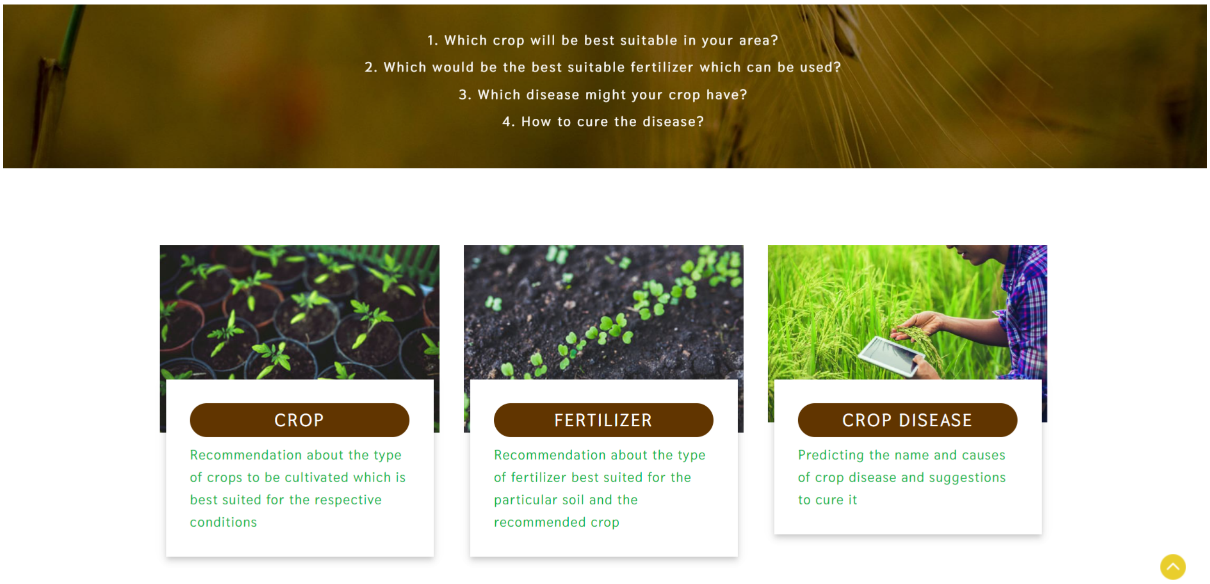


Figure 6.2: Home Page of Plant Leaf Disease Classification System

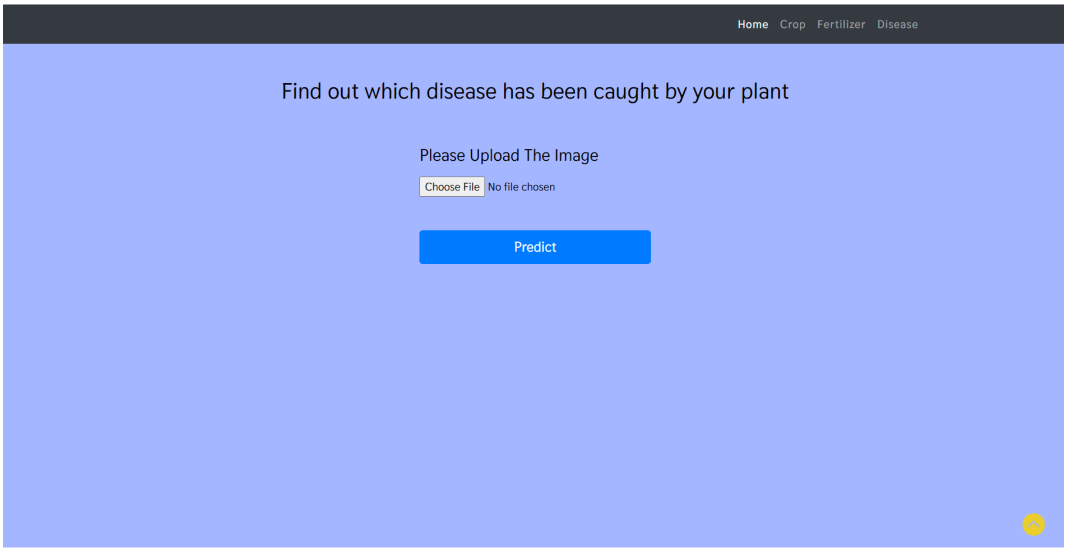


Figure 6.3: User interface of Disease Classification Page

The UI of a plant leaf disease classification system using CNN (Figure 6.3) typically includes an interface that allows users to upload an image of a plant leaf, a button to initiate the prediction process, and a display area that shows the results of the prediction, indicating whether the leaf is healthy or diseased.

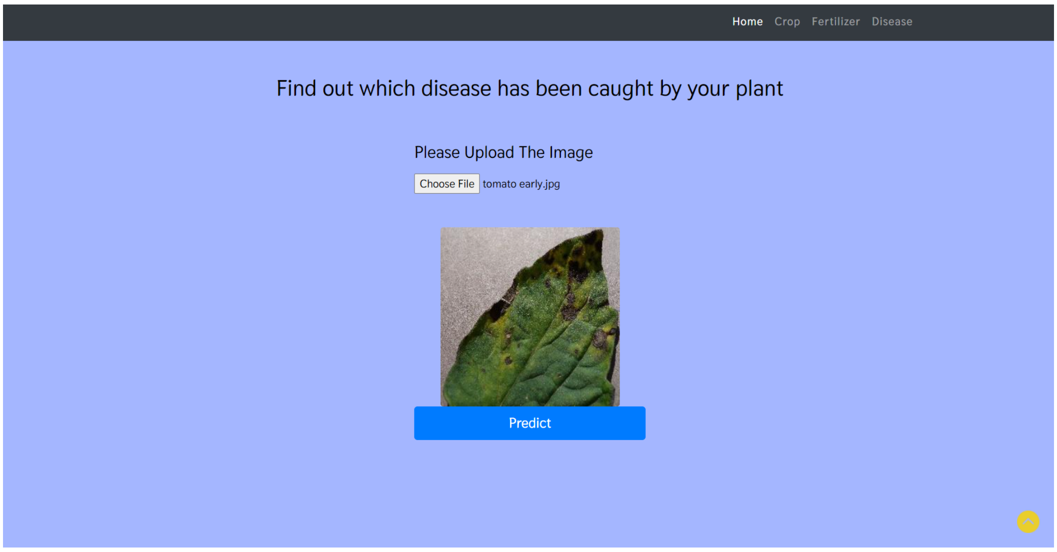


Figure 6.4: Upload image page of Plant Leaf

The interface in Figure 6.4 above allows the user to upload an image of a leaf and receive a prediction of whether or not the plant is diseased.

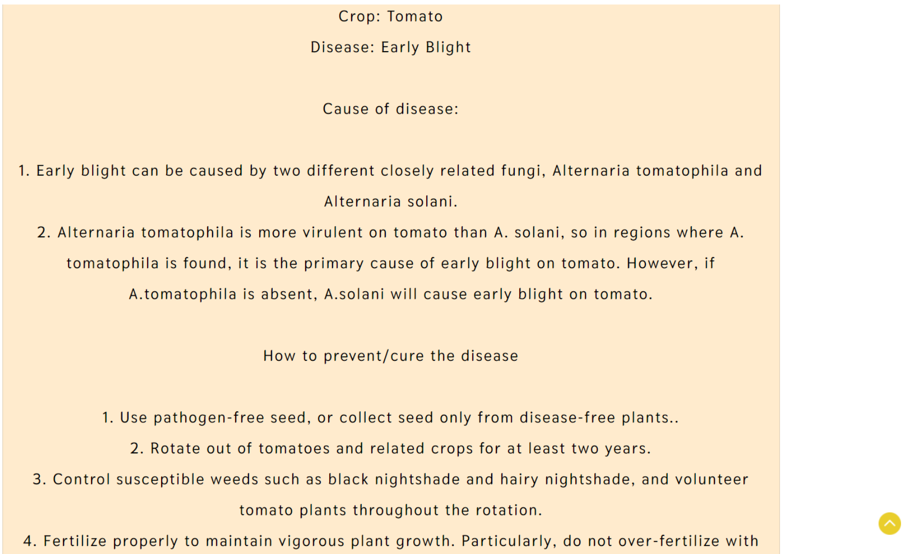


Figure 6.5: Result for Plant Disease Classification

The above figure 6.5 shows the disease detected for the uploaded image. The causes of the disease are also predicted with the remedy for the disease.

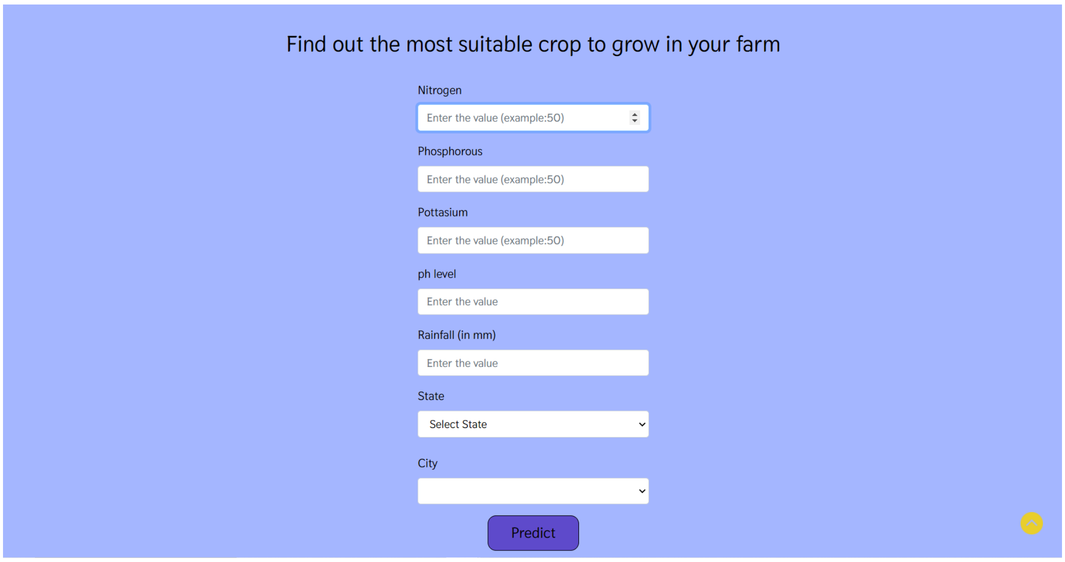


Figure 6.6: User Interface for Crop Recommendation Page

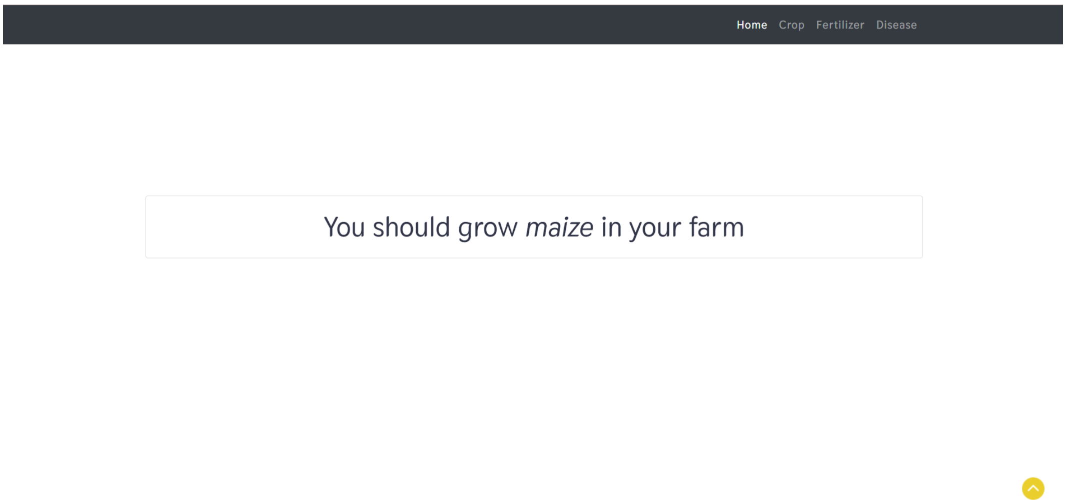


Figure 6.7: Result for Crop Recommendation Page

As shown in Figure 6.6 Farmers can input their soil and climate data using the user-friendly platform provided by the Crop Recommendation system's user interface (UI). Based on variables like soil type, nutrient content, and weather patterns, the algorithm then analyses the data and suggests the optimum crop varieties. The recommended crop variety is displayed on the user interface to aid farmers in making wise choices as shown in figure 6.7. The UI aims to boost farmer productivity by streamlining the crop recommendation process.

Figure 6.8: User Interface for Fertilizer Recommendation Page

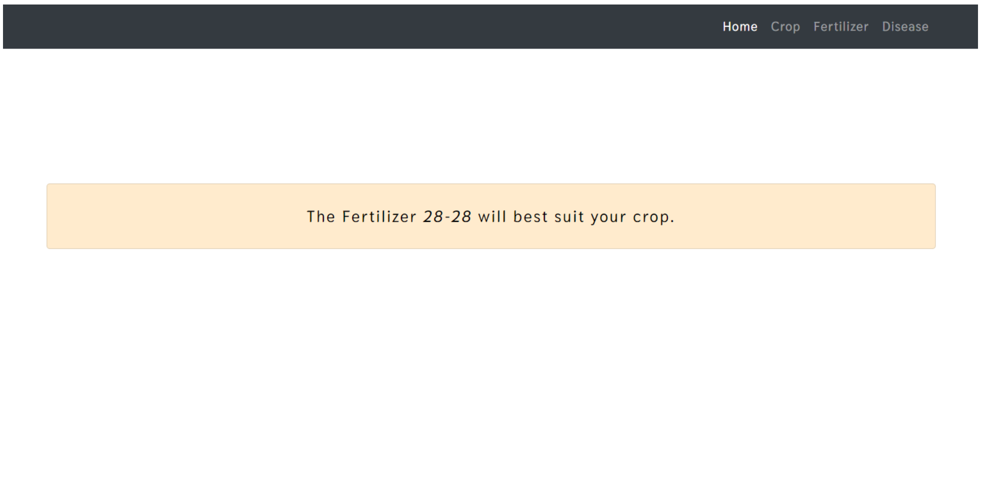


Figure 6.9: Result for Fertilizer Recommendation Page

Figure 6.8, the UI of the fertilizer recommendation system suggests the best fertilizer type based on soil analysis and crop requirements. In order to maximize crop productivity while reducing fertilizer waste and environmental effects, it intends to offer farmers a tailored solution. The suggested fertilizer is displayed on the user interface of the system as shown in figure 6.9.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

In this project, plant leaf disease detection, and classification method are presented based on Convolutional Neural Networks using Deep Learning. The dataset consists of 70,295 plant leaf images which consist of potato, tomato, apple, pepper, and 10 more types of plant leaves. The analysis has been carried out on healthy and diseased leaf images to perform classification. It is concluded that the proposed method effectively recognizes different types of leaf diseases for the respective type of leaf.

Along with disease classification, we have also implemented crop recommendation and fertilizer recommendation modules. The crop recommendation and the fertilizer recommendation use regression models whereas crop disease identification uses a convolutional neural network model. This would overall help farmers to solve queries regarding which crop to yield, what fertilizer to be used, and check whether the plant leaf is diseased or not, if yes then what type of disease.

7.2 Future Scope

Future studies will concentrate on problems with real-time data collection and multi-object deep learning models that can identify plant diseases from a cluster of leaves rather than a single leaf. Furthermore, we are working to incorporate the developed trained model into a mobile application. In the future, it will help farmers and the agricultural sector by instantly detecting leaf illnesses and providing guidance or herbal remedies.
Our future work is aimed at an improved data set with a large number of attributes and also implements yield prediction. As a result, the model will be improved and agricultural output will rise.

Bibliography

[1] Marzougui, F., Elleuch, M., Kherallah, M. (2020). A Deep CNN Approach for Plant Disease Detection. 2020 21st International Arab Conference on Information Technology (ACIT). doi:10.1109/acit50332.2020.9300072

[2] Liu, J., Wang, X. Plant diseases and pests detection based on deep learning: a review. Plant Methods 17, 22 (2021). https://doi.org/10.1186/s13007-021-00722-9.

[3] Pandian, J.A.; Kumar, V.D.; Geman, O.; Hnatiuc, M.; Arif, M.; Kanchanadevi, K. Plant Disease Detection Using Deep Convolutional Neural Network. Appl. Sci. 2022, 12, 6982. https://doi.org/10.3390/ app12146982.

[4] Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant Disease Detection and Classification by Deep Learning. PlanNetworks, 8, 468. https://doi.org/10.3390/plants8110468

[5] Srivastava, Prakanshu / Mishra, Kritika / Awasthi, Vibhav / Sahu, Vivek / Pal, Pawan Kumar. (2021). PLANT DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK. International Journal of Advanced Research. 09.691-698.10.21474/IJAR01/12346.

[6] L. Li, S. Zhang and B. Wang, ”Plant Disease Detection and Chasification by Deep Learning—A Review,” in IEEE Access, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646.

[7] A. KP and J. Anitha, ”Plant disease classification using deep learning,” 2021 3rd International Conferethe nce on Signtypesrocessing and Communithe cation (ICPSC), Coimthe batore, India, 2021, pp. 407-411, doi: 10.1109/ICSPC51sizes2021.9451696

[8] Sai Reddy, B., Neeraja, S. Plant leaf disease classification and damage detection system using deep learning models. Multimed Tools Appl 81, 24021–24040 (2022). https://doi.org/10.1007/s11042-022-12147-0

[9] Dongfang Wang, Jun Wang, Wenrui Li, Ping Guan, T-CNN: Trilinear convolutional neural networks model for visual detection of plant diseases, Computers and Electronics in Agriculture,Volume 190,2021,106468,ISSN 0168-1699,https://doi.org/10.1016/j.compag.2021.106468. (https://www.sciencedirect.com/science/article/pii/S0168169921004853)

[10] Geetharamani G., Arun Pandian J., Identification of plant leaf diseases using a nine-layer deep convolutional neural network, Computers Electrical Engineering, Volume 76,2019,Pages 323-338,ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2019.04.011. (https://www.sciencedirect.com/science/article/pii/S0045790619300023)

[11] G. Shrestha, Deepsikha, M. Das and N. Dey, ”Plant Disease Detection Using CNN,” 2020 IEEE Applied Signal Processing Conference (ASPCON), Kolkata, India, 2020, pp. 109-113, doi: 10.1109/ASPCON49795.2020.9276722.

[12] Parul Sharma, Yash Paul Singh Berwal, Wiqas Ghai,Performance analysis of deep learning CNN models for disease detection in plants using image segmentation,Information Processing in Agriculture,Volume 7, Issue 4,2020,Pages 566-574,ISSN 2214-3173,https://doi.org/10.1016/j.inpa.2019.11.001. (https://www.sciencedirect.com/science/article/pii/S2214317319301957)

[13] Konstantinos P. Ferentinos,Deep learning models for plant disease detection and diagnosis,Computers and Electronics in Agriculture,Volume 145,2018,Pages 311-318,ISSN 0168-1699,https://doi.org/10.1016/j.compag.2018.01.009. (https://www.sciencedirect.com/science/article/pii/S0168169917311742)

[14] M. Sardogan, A. Tuncer and Y. Ozen, ”Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm,” 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, Bosnia and Herzegovina, 2018, pp. 382-385, doi: 10.1109/UBMK.2018.8566635

Appendices

- cd Downloads
- cd B.E-Major-project-Master
- pip install virtualenv
- Install Anaconda
- conda create -n venv python=3.6.12
- conda activate venv
- pip install -r requirements.txt
- pip install numpy
- pip install pandas
- pip install Flask
- pip install scikit-learn
- pip install pytorch
- pip install requests
- pip install pillow
- gunicorn == 20.0.4
- pip install xgboost==1.2.1

Publication

Paper entitled **"Agrolife: Machine Learning Enabled Plant Disease Classification System"** is accepted at **"The 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering"** by **"Anagha Rai, Ishika Sharma, Ruta Mhaskar, Dr. Sameer Nanivadekar, Prof. Sonal Jain and Prof. Charul Singh".**