ENHANCED DETECTION ON LEAF DISEASES USING CNN

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

Y.SAISRI	(21UECS0699)	**(20076)**
L.NANDA KISHORE REDDY	(21UECS0338)	**(19775)**
P.RAJA SEKHAR	(21UECS0485)	**(19627)**

Under the guidance of
Dr.C M CHIDAMBARANATHAN, M.TECH, Ph.D.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

ENHANCED DETECTION ON LEAF DISEASES USING CNN

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

Y.SAISRI	(21UECS0699)	**(20076)**
L.NANDA KISHORE REDDY	(21UECS0338)	**(19775)**
P.RAJA SEKHAR	(21UECS0485)	**(19627)**

Under the guidance of
Dr.C M CHIDAMBARANATHAN, M.TECH, Ph.D.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "ENHANCED DETECTION ON LEAF DISEASES USING CNN" by "Y.SAISRI (21UECS0699), L.NANDA KISHORE REDDY (21UECS0338), P.RAJASEKHAR (21UECS0485)" has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor	**Signature of Professor In-charge**
Computer Science & Engineering	**Computer Science & Engineering**
School of Computing	**School of Computing**
Vel Tech Rangarajan Dr. Sagunthala R&D	**Vel Tech Rangarajan Dr. Sagunthala R&D**
Institute of Science & Technology	**Institute of Science & Technology**
May, 2024	**May, 2024**

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We 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.

(Signature)
Y.SAISRI
Date: / /

(Signature)
L.NANDA KISHORE REDDY
Date: / /

(Signature)
P.RAJA SEKHAR
Date: / /

APPROVAL SHEET

This project report entitled "ENHANCED DETECTION ON LEAF DISEASES USING CNN" by Y.SAISRI (21UECS0699), L.NANDA KISHORE REDDY (21UECS0338), P.RAJA SEKHAR (21UECS0485) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr.C M CHIDAMBARANATHAN, M.TECH, Ph.D.,
ASSISTANT PROFESSOR

Date: / /
Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Dr.C M CHIDAMBARANATHAN, M.E, Ph.D., for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.**

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U.HEMAVATHI, M.E., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

Y.SAISRI	**(21UECS0699)**
L.NANDA KISHORE REDDY	**(21UECS0338)**
P.RAJA SEKHAR	**(21UECS0485)**

ABSTRACT

In the realm of precision agriculture, leveraging data-driven technologies has become paramount for optimizing crop yield and resource utilization.This project addresses the innovative plant disease detection system utilizing Convolutional Neural Networks (CNN) within a comprehensive machine learning framework. The system addresses the critical issue of plant diseases by automating the identification and classification process based on images of plant leaves. We have curated a diverse dataset encompassing various plant diseases, including Tomato Bacterial Spot, Potato Early Blight, and Corn (Maize) Common Rust, establishing a foundation for robust model training.The initial phase of the project involves exploratory data analysis, showcasing random samples from each disease category through a visual representation. The dataset is organized, incorporating labels and binary encoding for three distinct plant diseases. The images are then split into training and testing sets, and further preprocessing involves normalizing pixel values and reshaping the data to fit the CNN architecture.The CNN model is constructed with multiple convolutional and pooling layers, followed by dense layers for classification. The model is compiled using the categorical cross-entropy loss function and the Adam optimizer. The training process involves the use of an 80-20 split for training and validation datasets, with a specified number of epochs and batch size. The trained model is saved for future use.A visual representation of the model's training history is presented, showcasing its accuracy and validation accuracy over epochs. The model's performance is evaluated on the test set, demonstrating its efficacy in predicting plant diseases accurately.The project culminates in the development of a user-friendly web application using Streamlit, allowing users to upload images of plant leaves for disease prediction. The application seamlessly integrates the trained CNN model, providing real-time predictions and displaying the predicted disease class.This comprehensive project not only contributes to the field of plant pathology but also demonstrates the practical application of machine learning, specifically CNNs, in automating plant disease detection.

Keywords:

Convolutional Neural Network(CNN), Training Dataset, Testing Dataset, Data Pre-Processing, Leaf Diseases, Realtime Predictions, Pathology, Accuracy.

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

AD	Alzheimer's Disease
CBA	Cost Benefit Analysis
CNN	Convolutional Neural Network
EDA	Exploratory Data Analysis
FDG-PET	Flurodeoxtglucose PET
GPU	Graphical Processing Unit
HOG	Histogram Oriented Gradients
KNN	K-Nearest Neighbour
LDD	Leaf Disease Detection
MCI	Mild Cognitive Impairement
Ml	Machine Learning
PLS	Partial Least Sqares
ROI	Return ON Investment
UI	User Interface

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Agriculture being the backbone of many economies,constantly seeks innovative solutions to enhance productivity and sustainability. Enhanced Detection On Leaf Diseases Using Convolutional Neural Networks (CNN) represents a significant contribution to the field of precision agriculture by harnessing the power of machine learning for early diagnosis of plant diseases. Plant diseases pose a substantial threat to global food security, leading to significant crop losses if not detected and addressed promptly. In response to this challenge, our project focuses on developing an intelligent system that leverages advanced image processing and machine learning techniques to identify and classify plant diseases accurately.The use of Convolutional Neural Networks (CNN) is enabling the automated recognition of intricate patterns and features within images of plant leaves. We have curated a diverse dataset, encompassing common plant diseases such as Tomato Bacterial Spot, Potato Early Blight, and Corn (Maize) Common Rust. The project not only contributes to the advancement of machine learning applications in agriculture but also addresses the pressing need for efficient and accessible tools for farmers and agricultural practitioners.

The initiative begins with an exploration of the dataset through visual representation, providing an overview of the plant diseases under consideration. The subsequent development of the CNN model involves comprehensive data preprocessing, training, and validation to ensure robust and accurate disease prediction capabilities. The project concludes with the implementation of a user-friendly web application using Streamlit, allowing farmers to easily upload images for on-the-spot disease diagnosis.By integrating cutting-edge technology with agriculture, our project aims to empower farmers with a proactive tool for early disease detection, ultimately contributing to enhanced crop yield, sustainable farming practices, and global food security.

1.2 Aim of the project

The primary aim of the project, ”Enhanced Detection On Leaf Diseases Using Convolutional Neural Networks (CNN),” is to develop a robust and accessible solution for early detection and classification of plant diseases through the application of advanced machine learning techniques. The agricultural sector faces significant challenges from various plant diseases that can lead to substantial crop losses if not identified and addressed promptly. The project seeks to leverage the capabilities of CNNs to automate the process of disease recognition, thereby providing farmers with a tool that enhances their ability to manage and protect crops effectively.Develop preprocessing techniques to ensure the quality and suitability of the dataset for training the CNN model.

This includes resizing images, normalizing pixel values, and encoding labels.Curate a diverse dataset encompassing images of plant leaves affected by common diseases such as Tomato Bacterial Spot, Potato Early Blight, and Corn (Maize) Common Rust. This dataset serves as the foundation for training and validating the machine learning model.Provide a proactive tool for farmers that enables early detection of plant diseases, facilitating timely interventions and effective crop management. This contributes to reducing crop losses and fostering sustainable agricultural practices.By achieving these objectives, the project aims to bridge the gap between technological advancements in machine learning and the practical needs of the agriculture sector, ultimately enhancing the resilience of global food production systems.

1.3 Project Domain

The project operates within the domain of precision agriculture, a field that leverages advanced technologies, data analytics, and automation to optimize farming practices. Precision agriculture aims to enhance the efficiency, sustainability, and productivity of agricultural processes, addressing the challenges posed by factors such as climate change, resource constraints, and the need for increased food production.In the context of our project, the domain specifically focuses on plant disease detection, a critical aspect of precision agriculture. Plant diseases represent a significant threat to crop yield and quality, leading to economic losses for farmers and potential food security issues globally.

The Traditional methods of disease detection often rely on manual observation, which can be time-consuming, The integration of machine learning, and particularly Convolutional Neural Networks (CNNs), into the domain of plant disease detection represents a transformative approach. By utilizing CNNs, the project aims to automate the identification and classification of plant diseases through the analysis of digital images of plant leaves. This technology not only expedites the detection process but also allows for early intervention, enabling farmers to implement timely and targeted solutions.

1.4 Scope of the Project

The scope of the ”Enhanced Detection On Leaf Diseases Using Convolutional Neural Networks (CNN)” project is defined by its objectives, functionalities, and potential impact within the domain of precision agriculture. The project encompasses various stages, from data acquisition and preprocessing to model development, training, and the deployment of a user-friendly application.The scope of the project is limited to the detection and classification of specific plant diseases mentioned in the dataset. The project aims to empower farmers with a tool for early disease detection, facilitating timely interventions and effective crop management. While the initial scope is focused on common plant diseases, the project lays the groundwork for potential extensions and advancements in the broader field of precision agriculture.Gathering and organizing a diverse dataset containing images of plant leaves affected by common diseases, including Tomato Bacterial Spot, Potato Early Blight, and Corn (Maize) Common Rust.

Creating a user-friendly web application using Streamlit to facilitate easy image uploads for real-time disease prediction. The application serves as an accessible interface for farmers and users in the agricultural sector. Assessing the accuracy and performance of the CNN model on a test set to validate its effectiveness in real-world scenarios. The project is limited to the detection and classification of specific plant diseases mentioned in the dataset. The project aims to empower farmers with a tool for early disease detection, facilitating timely interventions and effective crop management. While the initial scope is focused on common plant diseases, the project lays the groundwork for potential extensions and advancements in the broader field of precision agriculture.

Chapter 2

LITERATURE REVIEW

[1] Ramesh Maniyath, et al.,(2022)proposed about review on Machine Learning Techniques for Leaf Disease Detection and Classification a fast-response colorimetric ultraviolet-C(UVC) sensor was demonstrated using a gallium oxide (Ga2O3) photocatalyst with small amounts of triethanolamine (TEOA) in methylene blue (MB) solutions and a conventional RGB photodetector. The color of the MB solution changed upon UVC exposure, which was observed using an in situ RGB photodetector. Thereby,the UVC exposure was numerically quantified as an MB reduction rate with the R value of the photodetector, which was linearly correlated with the measured spectral absorbance using a UV-Vis spectrophotometer.Small amount of TEOA in the MB solution served as a hole scavenger, which resulted in fast MB color changes due to the enhanced charge separation. However, excessive TEOA over 5 wt.percentage started to block the catalytical active site on the surface of Ga2O3, prohibiting the chemical reaction between the MB molecules and catalytic sites. The proposed colorimetric UVC sensor could monitor the detrimental UVC radiation with high responsivity at a low cost.

[2] Owomugisha, et al.,(2022)described that the factors affecting the efficiency of micellemediated extraction of phenolic compounds from apple pomace was investigated.Higher extraction efficiency by using as a solvent an aqueous solution of Tween 80 in comparison to Triton X-100, Span 20, Tween 20, 70 percentage ethanol,and water was shown. Four independent variables (Tween 80 concentration, time,4 solvent-to material ratio, and pH) to enhance the recovery of polyphenols from apple pomace was investigated. Applying response surface methodology, the second order polynomial regression equation showing dependence of the yield of polyphenols on the extraction parameters was derived. The adjusted regression coefficient (R2 =98.73the lack-of-fit test (p lessthan 0.05) showed a good accuracy of the developed model.

[3] Quinn, et al.,(2022)discussed about an experiment which shows classification and detection of plant diseases. Insecticides are not always proved efficient because insecticides may be toxic to some species of birds. It also damages natural animal food chains. The following two steps are added successively after 5 the segmentation phase. In the first step, they identified the mostly green colored pixels. Then using Otsu's method, pixels are masked having green color. Then those mostly green pixels are masked. After that, red green and yellow color based pixel cluster are removed which are infected. The experimental results show that proposed technique is the best for plant detection.

[4] Wang, et al.,(2019)proposed Accurate identification of plant diseases is vital for achieving optimal crop yields and agricultural productivity. The process of detecting plant illnesses involves comprehensive research on various factors related to farming, encompassing aspects like organic farming, continuous plant monitoring, and the recognition of diverse diseases. In agricultural settings with a variety of crops, manual tracking of plant diseases becomes impractical, demanding significant labor, expertise in plant diseases, and substantial time. Utilizing image processing, alongside algorithms such as k-means clustering and convolutional neural networking, proves to be an effective means for precise disease prediction. The disease detection process incorporates techniques such as image segmentation, data pre-processing, image fragmentation, and the identification and recognition of distinctive features. This study also investigates the segmentation and retrieval functions concerning two specific plant diseases, employing keywords such as image processing, plant disease detection, k-means clustering algorithm, and Convolutional Neural Network.

[5] Shi, et al.,(2019)In recent years there has been a growing focus on identifying the conversion from mild cognitive impairment (MCI) to Alzheimer's disease (AD). Brain neuroimaging techniques, including magnetic resonance imaging (MRI), 18F-fluorodeoxyglucose PET (FDG-PET), and 18F-florbetapir PET (florbetapir-PET), play a crucial role in classifying or predicting MCI outcomes. This study, utilizing data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database, employed the partial least squares (PLS) method to differentiate MCI converters (MCI-c, those progressing to AD) from MCI non-converters (MCI-nc, those not converting to AD). Two PLS models, informed and agnostic, were constructed based on 64 MCI-c and 65 MCI-nc.

[6] Zhang, et al.,(2018)reported the use of traditional methods such as direct observation, microscopy, and biochemical tests for the identification of leaf spot diseases on okra plants.In recent years, advanced technologies such as machine learning and computer vision have been developed to automate plant and disease detection.Computer vision involves the use of cameras to capture images of plants, and machine learning algorithms analyze the images for disease detection.

[7] Jiao, et al.,(2017)described this algorithm which considers shape, color, size, and texture features to achieve improved classification, employing various combinations of these features. The developed food calorie and nutrition assessment method can be utilized by dietitians to monitor and manage daily food intake effectively. Utilizing Convolutional Neural Networks (CNN), the texture, shape, and size properties of food images are extracted, contributing to enhanced classification accuracy and more precise estimation of food calories.

[8] Umer, et al.,(2016)proposed a transfer learning-based approach for plant disease identification. While the dimensions of the feature data were reduced by using PCA, the optimal recognition result for grape diseases was obtained as the fitting accuracy was 100 was 97.14prediction accuracy were both 100 percent.The authors used a pre-trained CNN model, Inception-v3, and finetuned it using a dataset of tomato plant images infected with five common diseases.The proposed approach achieved an accuracy rate of 98.61.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

Before the adoption of advanced machine learning techniques, plant disease detection predominantly relied on manual observation, agricultural expertise, and occasionally laboratory tests. However, traditional methods, particularly those utilizing k-Nearest Neighbors (k-NN) algorithm for disease detection.Similar to manual inspection in traditional methods, k-NN's effectiveness relies on distance measures between feature vectors. This approach can be subjective, and the choice of features and distance metrics might lack the precision needed for accurate diseases identification.Calculating distances between the test instance and all training instances in k-NN can be time-consuming, especially as the dataset size increases. This computational overhead may hinder timely disease identification.

Disadvantages:

- Data Dependency
- Overfitting
- Computatioanl intensity

3.2 Proposed System

The proposed system aims to revolutionize plant disease detection by leveraging advanced machine learning techniques, specifically Convolutional Neural Networks (CNNs). By automating the identification and classification process based on images of plant leaves, the system seeks to overcome the limitations of traditional methods, offering a more accurate, timely, and scalable solution for farmers and agricultural practitioners. Curate a diverse dataset encompassing images of plant leaves affected by common diseases, including Tomato Bacterial Spot, Potato Early Blight, and Corn (Maize) Common Rust.Include healthy plant images to ensure a balanced dataset for comprehensive model training.Conduct EDA to visualize random samples from each disease category, providing insights into dataset diversity.Analyze unique characteristics of different plant diseases to inform feature extraction.

Develop a custom function to convert plant leaf images to arrays, facilitating data preprocessing.Organize the dataset with labels and binary encoding for disease classes.Split the dataset into training and testing sets, normalizing pixel values, and reshaping data for compatibility with the CNN architecture.Train the CNN model on the curated dataset, optimizing its ability to accurately predict and classify plant diseases.Implement an 80-20 split for training and validation datasets, specifying the number of epochs and batch size.Save the trained model for future use.Consider potential enhancements to the model, such as incorporating additional plant diseases, fine-tuning hyperparameters, or exploring advanced CNN architectures.Optimize the model based on user feedback and real-world performance.

Advantages:

- Remote Sensing Integration
- High Accuracy
- Real-time Detection

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility of implementing an enhanced detection system for leaf diseases using Convolutional Neural Networks (CNNs) involves a comprehensive analysis of various cost and benefit factors. Initial setup costs encompass hardware, such as high-performance GPUs, and software, including licensing fees for deep learning frameworks.Cost-Benefit Analysi(CBA) Conduct a cost-benefit analysis to estimate the financial implications of the project. Consider costs related to data acquisition, model development, web application development, and ongoing maintenance.Return on Investment (ROI) Evaluate the potential return on investment, considering factors such as increased crop yield, reduced losses due to diseases, and the economic value of timely interventions.

Data collection and annotation expenses involve acquiring a diverse dataset of leaf images and annotating it accurately. The development and training phase incur costs related to personnel salaries and time investment. Infrastructure and maintenance costs cover ongoing expenses for cloud services, energy consumption, and system upkeep. Deployment and integration expenses include software integration and user training. Operational costs consider ongoing monitoring and support. The benefits and returns must be evaluated, including potential yield increase, reduced labor costs, and improvements in crop quality and market value. Risk factors and contingencies should be taken into account, addressing uncertainties and unforeseen challenges. Comparisons with alternative methods and consideration of long-term sustainability, scalability, and upgradability are essential in determining the overall economic feasibility of implementing a CNN-based leaf disease detection system, aligning with the specific goals and resources of the agricultural operation or business.

3.3.2 Technical Feasibility

The technical feasibility of implementing an enhanced detection system for leaf diseases using Convolutional Neural Networks (CNNs) is grounded in the robust capabilities of deep learning technology. CNNs have proven effectiveness in image classification tasks, precisely suited to discerning intricate patterns and features within leaf images indicative of various diseases. The availability of powerful hardware, such as Graphics Processing Units (GPUs), facilitates the training of sophisticated CNN models, allowing for the extraction of hierarchical features crucial for accurate disease identification. Transfer learning, leveraging pre-trained models, enhances the efficiency of model development, especially when confronted with limited labeled data. Furthermore, advancements in deep learning frameworks and tools contribute to the technical feasibility by providing scalable and adaptable solutions for model implementation Assess the technical expertise required for implementing the CNN-based plant disease detection system. Ensure the availability of skills in machine learning, computer vision, and web development.

Evaluate the availability and compatibility of the required technology infrastructure, including computational resources for model training and deployment. The integration of CNN-based systems with remote sensing technologies, such as drones or satellite imagery, extends the technical feasibility to large-scale and remote agricultural environments. Continuous improvements in model interpretability and explainability further strengthen the technical foundation, ensuring that the deployed system not only performs accurately but is also transparent and comprehensible to end-users and stakeholders. Overall, the technical feasibility of employing CNNs for leaf disease detection is well-supported by the advancements in deep learning methodologies and the adaptability of these technologies to agricultural contexts.

3.3.3 Social Feasibility

The social feasibility of implementing an enhanced detection system for leaf diseases using Convolutional Neural Networks (CNNs) is contingent upon various factors that influence its acceptance and integration into agricultural practices. A key aspect is the accessibility and inclusivity of the technology, ensuring that it aligns with the capabilities and resources of diverse agricultural communities.

Additionally, the ease of use and user-friendly interfaces contribute to social acceptance, making the technology accessible to farmers with varying levels of technological literacy. Collaboration with local communities, extension services, and agricultural experts is crucial to ensuring that the CNN-based system is culturally appropriate and addresses the specific needs of the users. The system's transparency and the provision of understandable insights into disease detection contribute to social acceptance, fostering trust among farmers and stakeholders.

3.4 System Specification

3.4.1 Hardware Specification

- Processor : I3/I5 Intel Processor
- RAM : 8 GB

3.4.2 Software Specification

- PYCHARM
- PYTHON
- PRE-TRAINED MODELS
- DEEP LEARNING MODELS
- WEBCAM

3.4.3 Standards and Policies

PyCharm

PyCharm is a type of command line interface which explicitly deals with the ML(MachineLearning) modules.And navigator is available in all the Windows,Linux and MacOS.The PyCharm has many number of IDE's which make the coding easier. The UI can also be implemented in python.

Standard Used: ISO/IEC 27001

Jupyter

It's like an open source web application that allows us to share and create the documents which contains the live code, equations, visualizations and narrative text. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 General Architecture for Leaf Disease Detection

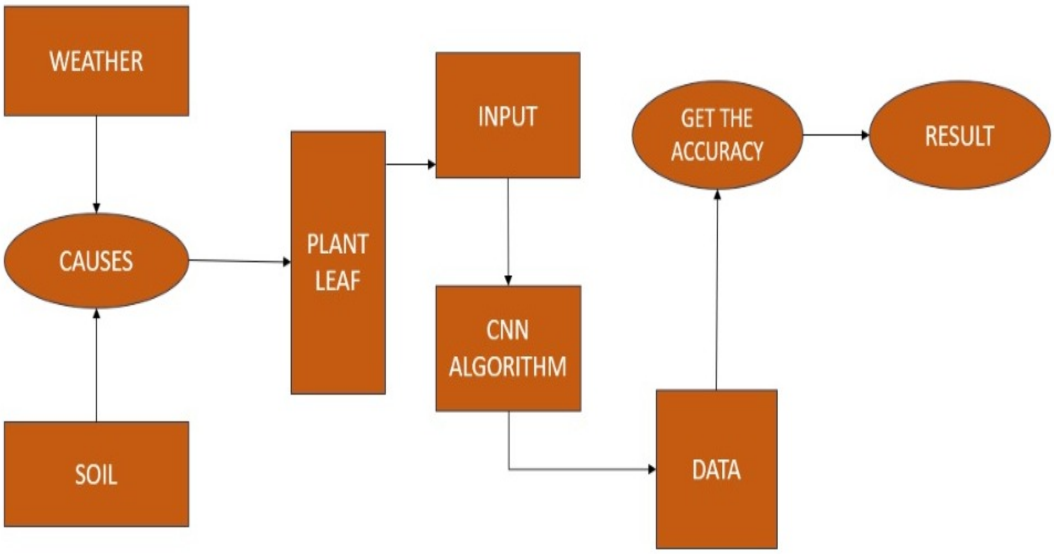


Figure 4.1: **Architecture Diagram for Leaf Disease Detection**

The above Figure 4.1 explains the architecture of enhanced detection on leaf diseases using cnn. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution road map. The system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. The system's architecture should prioritize user management, data storage, data process,device connection, web application.

4.2 Design Phase

4.2.1 Data Flow Diagram for Leaf Disease Detection

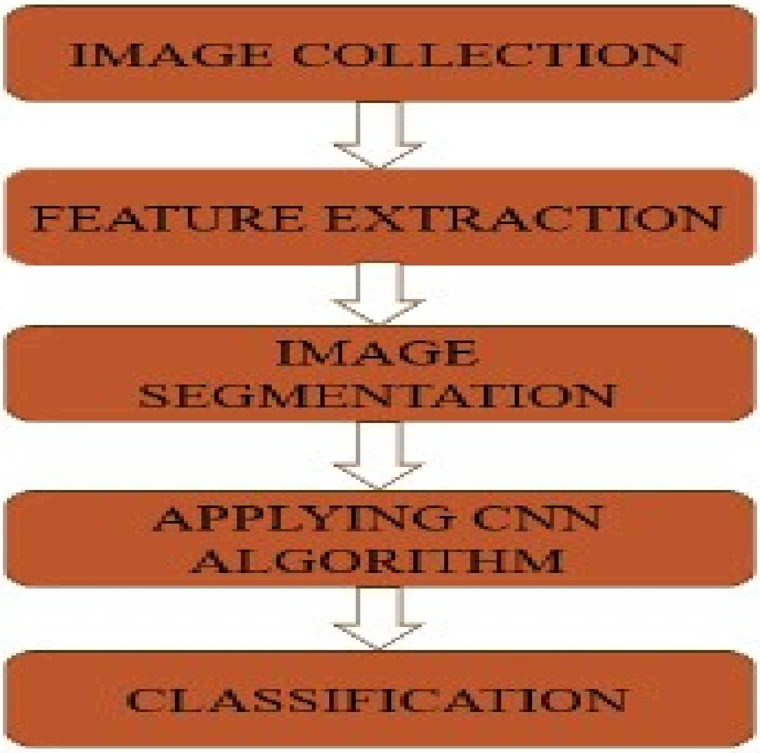


Figure 4.2: **Data Flow Diagram for Leaf Disease Detection**

The above Figure 4.2 explains the enhanced detection on leaf diseases and their data flow. A Data flow diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects.Often it is a preliminary step used to create an overview of the system that can later be elaborated. It is the process of collecting leafs which is diseased.

4.2.2 Collabration Diagram for Leaf Disease Detection

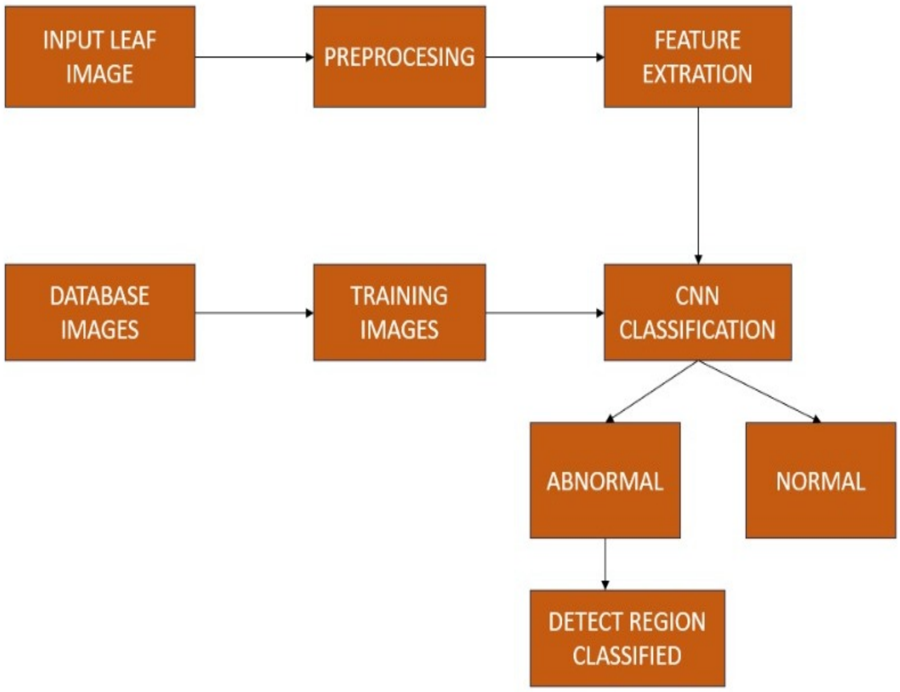


Figure 4.3: **Collabration Diagram for Leaf Disease Detection**

The above figure 4.3 explains effective data processing lays the foundation for the model's success. A well-prepared dataset, enriched through thoughtful preprocessing techniques, empowers the CNN to learn discriminative features and patterns, ultimately leading to a more accurate and robust leaf disease detection system. As advancements continue in data processing methodologies, they will contribute significantly to the ongoing refinement and effectiveness of CNN models in the realm of plant health monitoring.

4.2.3 Sequence Diagram for Leaf Disease Detection

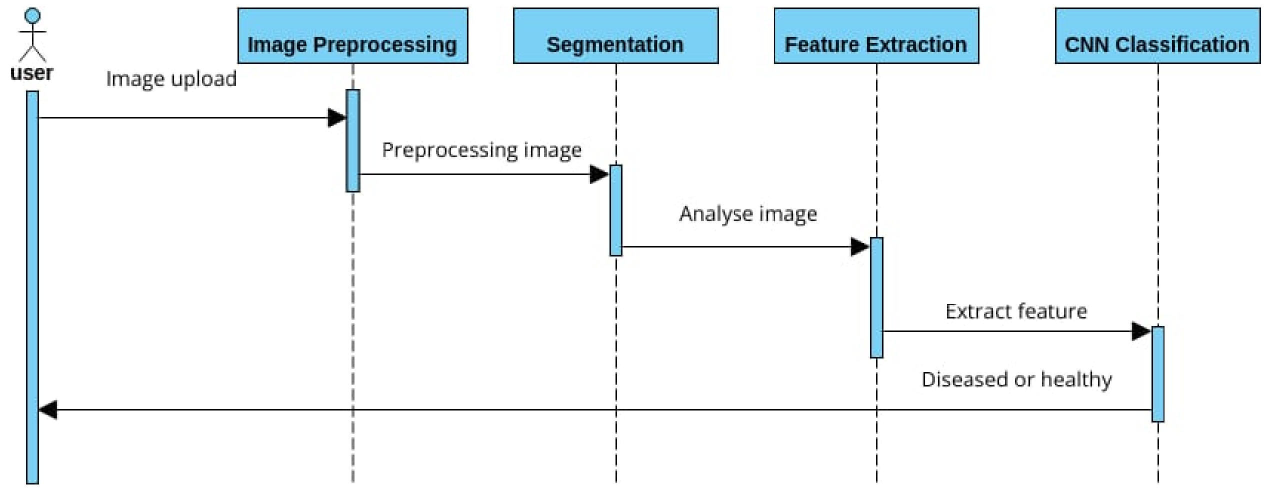


Figure 4.4: **Sequence Diagram for Leaf Disease Detection**

The above figure 4.4 explains the visual representation that illustrates the flow of interactions and messages between different components or objects in a system over time. In the context of enhanced detection of leaf diseases using Convolutional Neural Networks (CNN), a sequence diagram can provide a detailed view of the communication and collaboration between the various elements involved in the detection process.

4.2.4 Activity Diagram for Leaf Disease Detectio

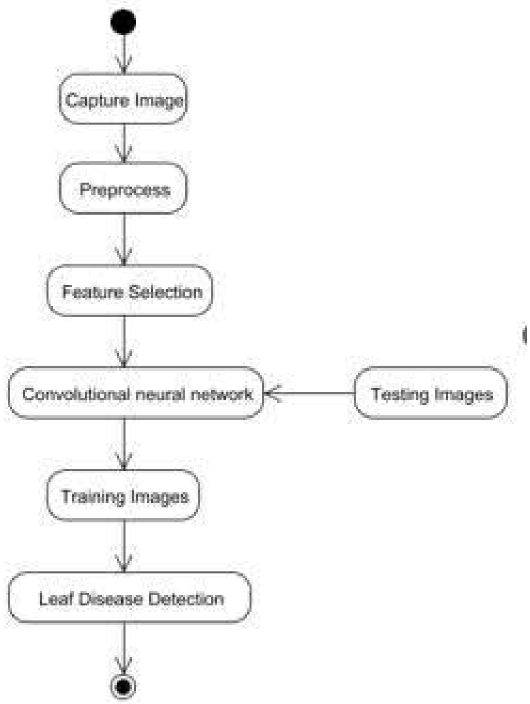


Figure 4.5: **Activity Diagram for Leaf Disease Detection**

The above figure 4.4 explains the context of enhanced detection of leaf diseases using Convolutional Neural Networks (CNN), an activity diagram serves as a comprehensive visual representation of the intricate workflow involved in the detection process. The diagram begins with the initiation of the process, typically triggered by the acquisition of input data, which could consist of images of plant leaves suspected of harboring diseases. This input acquisition activity leads to the preprocessing stage, wherein the acquired images undergo essential tasks such as resizing, normalization, and data augmentation. These preprocessing activities are pivotal in ensuring that the input data is adequately prepared for effective analysis by the CNN model.

4.3 Algorithm & Pseudo Code

4.3.1 Enhanced Convolutional Neural Network Algorithm

The chosen algorithm for the proposed Leaf Disease Detection System is Convolutional Neural Networks (CNNs). CNNs are a class of deep neural networks designed for image recognition and processing tasks. They have proven to be highly effective in capturing intricate patterns and features within visual data, making them well-suited for plant disease detection based on leaf images.CNNs utilize convolutional layers to scan the input images using filters or kernels. These filters learn local patterns, such as edges, textures, or colors, in different regions of the image.Convolutional operations help in feature extraction, enabling the network to identify hierarchical features.Pooling layers, typically implemented as max pooling or average pooling, reduce the spatial dimensions of the feature maps. This downsampling aids in retaining important information while reducing computational complexity. Pooling helps make the network more invariant to variations in scale, orientation, and position of features.

4.3.2 Pseudo Code

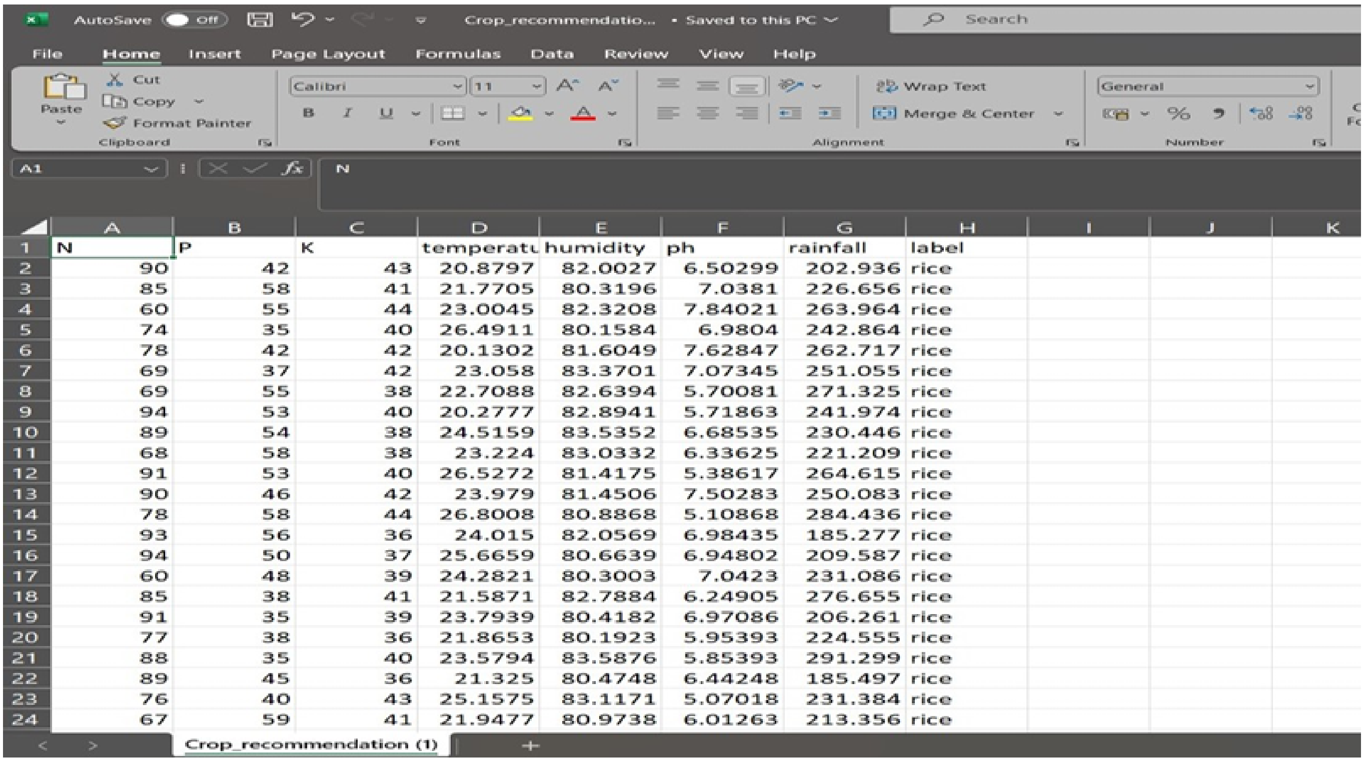
```
import numpy as np
import streamlit as st
import cv2
from keras.models import load_model
import tensorflow as tf

model = load_model('New-plant-disease.h5')

CLASS_NAMES = ['Tomato_Bacterial_spot', 'Potato_Early_blight', 'Corn_(maize)_Common_rust_']

st.title("Plant Disease Prediction Website")
st.markdown("Upload an Image of the Leaf")

plant_image = st.file_uploader("Choose an Image....", type="jpg")
submit = st.button('Predict Disease')



if submit:

```

4.4 Module Description

Module description is a fundamental aspect of software documentation, providing a concise overview of a specific software module's purpose, functionality, and key features.The main functionalities or capabilities provided by the module. Describe the specific tasks or operations that the module performs to achieve its purpose.

4.4.1 Data Collection

The data collection process for leaf disease detection using Convolutional Neural Networks (CNNs) typically involves several steps to ensure the effectiveness and accuracy of the model. Firstly, a diverse dataset of leaf images is required, encompassing various plant species, growth stages, and environmental conditions to capture the variability of leaf diseases. These images can be sourced from publicly available datasets, online repositories, or collected through field surveys and experiments.Throughout the data collection process, careful attention must be paid to ensure the representativeness, quality, and balance of the dataset to avoid biases and improve the robustness of the CNN model for leaf disease detection. Additionally, ethical considerations regarding data privacy, consent, and usage rights should be adhered to in accordance with relevant regulations and guidelines.



N	P	K	temperatu	humidity	ph	rainfall	label
90	42	43	20.8797	82.0027	6.50299	202.936	rice
85	58	41	21.7705	80.3196	7.0381	226.656	rice
60	55	44	23.0045	82.3208	7.84021	263.964	rice
74	35	40	26.4911	80.1584	6.9804	242.864	rice
78	42	42	20.1302	81.6049	7.62847	262.717	rice
69	37	42	23.058	83.3701	7.07345	251.055	rice
69	55	38	22.7088	82.6394	5.70081	271.325	rice
94	53	40	20.2777	82.8941	5.71863	241.974	rice
89	54	38	24.5159	83.5352	6.68535	230.446	rice
68	58	38	23.224	83.0332	6.33625	221.209	rice
91	53	40	26.5272	81.4175	5.38617	264.615	rice
90	46	42	23.979	81.4506	7.50283	250.083	rice
78	58	44	26.8008	80.8868	5.10868	284.436	rice
93	56	36	24.015	82.0569	6.98435	185.277	rice
94	50	37	25.6659	80.6639	6.94802	209.587	rice
60	48	39	24.2821	80.3003	7.0423	231.086	rice
85	38	41	21.5871	82.7884	6.24905	276.655	rice
91	35	39	23.7939	80.4182	6.97086	206.261	rice
77	38	36	21.8653	80.1923	5.95393	224.555	rice
88	35	40	23.5794	83.5876	5.85393	291.299	rice
89	45	36	21.325	80.4748	6.44248	185.497	rice
76	40	43	25.1575	83.1171	5.07018	231.384	rice
67	59	41	21.9477	80.9738	6.01263	213.356	rice

Figure 4.6: **Data Collection for Leaf Disease Detection**

4.4.2 Data Processing

To enhance leaf disease detection using Convolutional Neural Networks (CNNs), a comprehensive data processing pipeline is crucial. Initially, a diverse dataset of leaf images representing various diseases and healthy states is collected. These images need preprocessing to ensure uniformity in size, color, and orientation. Common preprocessing techniques include resizing, normalization, and augmentation to increase the dataset's variability and robustness. Next, the dataset is divided into training, validation, and testing sets to facilitate model training, evaluation, and validation. It's essential to maintain a balanced distribution of classes across these sets to prevent biases. Continuous monitoring and updating of the model may be necessary to adapt to new leaf diseases or changes in environmental conditions. Additionally, deploying the model into production environments requires considerations for scalability, efficiency, and integration with existing systems or applications.

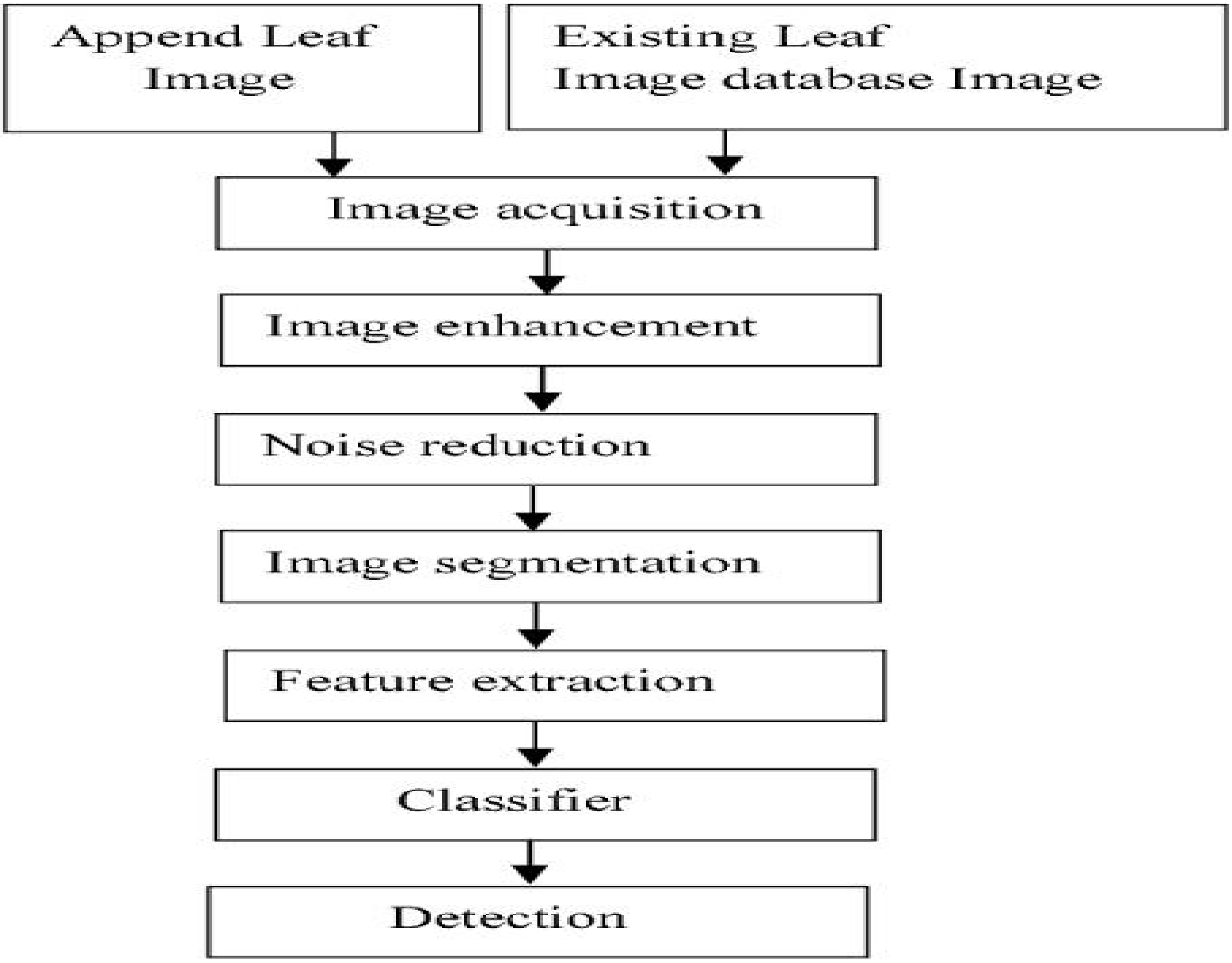


Figure 4.7: **Data Processing for Leaf Disease Detection**

4.4.3 CNN Algorithm for Leaf Disease Detection

CNN (Convolutional Neural Network) algorithms have revolutionized the detection of leaf diseases in plants, offering enhanced accuracy and efficiency compared to traditional methods. The process begins with the collection of a diverse dataset containing images of both healthy and diseased plant leaves, meticulously labeled with the corresponding disease types. These images undergo preprocessing steps such as resizing, normalization, and augmentation to ensure uniformity and improve model performance. Next, a CNN model architecture is meticulously designed, typically comprising multiple convolutional layers followed by pooling layers. These layers act as feature extractors, hierarchically analyzing the input images to discern patterns indicative of various diseases. During the training phase, the model learns to differentiate between healthy and diseased leaves by iteratively adjusting its parameters through techniques like backpropagation and stochastic gradient descent.

```
[ ]  print("Specificity is ",(sum(speci)/22)*100)

     Sensitivity is  99.93508493503197
     Specificity is  98.58363238128635

     labels=[f"{x}" for x in range(1,23)]
     print(classification_report(y_test, y_pred, target_names=labels))

                   precision    recall  f1-score   support

               1       1.00      1.00      1.00        28
               2       1.00      1.00      1.00        30
               3       1.00      0.97      0.98        31
               4       1.00      1.00      1.00        34
               5       0.96      1.00      0.98        26
               6       1.00      1.00      1.00        29
               7       0.96      0.96      0.96        28
               8       1.00      1.00      1.00        30
               9       0.94      0.94      0.94        31
              10       0.96      1.00      0.98        26
              11       0.95      0.95      0.95        22
              12       0.96      0.96      0.96        27
              13       1.00      1.00      1.00        28
              14       0.97      1.00      0.99        36
              15       1.00      1.00      1.00        29
              16       1.00      1.00      1.00        30
              17       1.00      1.00      1.00        34
              18       1.00      1.00      1.00        39
              19       1.00      0.96      0.98        28
              20       1.00      1.00      1.00        32
              21       0.97      0.95      0.96        37
              22       1.00      1.00      1.00        25

        accuracy                           0.99       660
       macro avg       0.99      0.99      0.99       660
    weighted avg       0.99      0.99      0.99       660
```

Figure 4.8: **Test Image for Leaf Disease Detection**

4.5 Steps to execute/run/implement the project

4.5.1 Step1: Installation

- Install Python(3.7.6)(Dont Forget to Tick Add to Path While installing python)
- Open terminal and execute following commands:
  pip install streamlit
  pip install keras
  pip install tensorflow

4.5.2 Step2:Extraction

- Download this projrct zip folder and extract it
- Move to project folder in Terminal. Then run the following commands:
  py manage.py makemigrations
  py manage.py migrate
  py manage.py runserver

4.5.3 Step3:Execution

- Now enter following URL in Your Browser Installed On Your Pc
  http://127.0.0.1:8000/

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

The input design for an enhanced detection system on leaf diseases using Convolutional Neural Networks (CNNs) is a critical aspect that directly influences the model's performance. Here's a comprehensive guide to input design:

Image Acquisition: Gather a diverse dataset of leaf images that encompass various plant species, growth stages, and environmental conditions.Ensure the dataset includes both healthy leaves and leaves affected by different diseases to enable the CNN to learn distinctive features.

Normalization and Preprocessing: Normalize pixel values to a common scale (e.g., [0, 1]) to facilitate model training and convergence.Apply preprocessing techniques such as resizing, cropping, and augmentation to enhance the diversity of the dataset.

Data Augmentation: Augment the dataset with variations in rotation, flipping, zooming, and other transformations.This helps the model generalize better to different orientations and conditions.

Transfer Learning and Pre-trained Models: Leverage transfer learning by using pre-trained CNN models on large datasets like Image Net. Fine-tune the model on your specific leaf disease dataset to improve convergence and performance.

Validation and Test Sets: Split the dataset into training, validation, and test sets. The validation set is crucial for tuning hyperparameters, while the test set evaluates the model's generalization to new, unseen data.

5.1.2 Output Design

The output design for an enhanced detection system on leaf diseases using Convolutional Neural Networks (CNNs) is crucial in translating the model's predictions into actionable insights. Here's a guide to designing the output of the system:

Data Format: The output data can be in various formats, such as text, graphs, or visualizations. The format used will depend on the specific application and the audience that the results are intended for.

Class Labels: Design the output layer of the CNN to produce class probabilities or scores corresponding to different leaf disease classes.Each output node should represent a specific disease class, and the final classification is based on the class with the highest probability.

Interpretability and Explanation: Implement mechanisms for providing explanations of the model's decisions. This could include highlighting specific features or patterns that contributed to the classification.

Integration with Agricultural Systems: Design outputs that can be seamlessly integrated into existing agricultural management systems. This might involve providing outputs in formats compatible with farm management software or data visualization tools.

User-Friendly Interface: Design a user-friendly interface that presents the output in an easily understandable format. This is especially important for users who may not have a background in machine learning.

Integration with Decision Support Systems: Facilitate integration with decision support systems that provide recommendations or actions based on the detected diseases. This could include suggesting specific treatments or interventions.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit Testing

Unit testing is a critical aspect of developing and maintaining an enhanced leaf disease detection system using Convolutional Neural Networks (CNN). In order to ensure the robustness and reliability of the system, it is imperative to adopt a systematic approach to testing individual components or units of the codebase. The first step involves organizing the code in a modular manner, adhering to principles of good software design. This modularization enables the isolation of specific functionalities into smaller units or functions, each dedicated to a distinct task such as data preprocessing, model architecture definition, training, and inference.

```

import numpy as np # linear algebra
import pandas as pd # data processing , CSV file I/O (e.g. pd. read csv)

# Input data files are available in the r e a d only . ./ input / directory
# For example , running this (by clicking run or pressing Shift+Enter) will list all files under the
i nput directory

import os
for dirname ,
, filenames in os.walk( / kaggle / input ) :
f or filename in filenames :

print ( os . path . join ( dirname , filename ) )
df= pd. read csv ( / content / Crop recommendation (1) . csv )
df
sns . stripplot (y = label , x = ph , hue = label , orient = h , data = df , size=5)
plt . show()
sns . stripplot (y = label , x = P , hue = label , orient = h , data = df , size=5)
plt . show()
```

Test result

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 28)                224

 dense_1 (Dense)             (None, 64)                1856

 dense_2 (Dense)             (None, 28)                1820

 dense_3 (Dense)             (None, 22)                638

=================================================================
Total params: 4,538
Trainable params: 4,538
Non-trainable params: 0
```

Figure 5.1: **Unit Testing for Leaf Disease Detection**

5.3.2 Integration Testing

Integration testing is a fundamental component in the development and refinement of an enhanced leaf disease detection system utilizing Convolutional Neural Networks (CNN). This level of testing focuses on evaluating the interaction and collaboration among different modules or components within the system, ensuring that they function cohesively to achieve the intended outcome. Integration tests are particularly crucial for a CNN-based leaf disease detection system as they validate the seamless integration of various elements, such as data preprocessing, model architecture, training processes, and inference mechanisms.

```
import tensorflow
from tensorflow . keras import Sequential
ann=Sequential ()
from tensorflow import keras
ann. add( keras . layers . Dense(28 , input shape =(7 ,) , activation= relu ) )
ann. add( keras . layers . Dense(64 , input shape =(7 ,) , activation= relu ) )
ann. add( keras . layers . Dense(28 , input shape =(7 ,) , activation= relu ) )
ann. add( keras . layers . Dense(22 , input shape =(7 ,) , activation= softmax ) )
ann. compile ( optimizer= adam , loss= sparse categorical crossentropy , metrics =[ accuracy
ann . fit (x=X train ,y=y train , batch
preds=ann . predict ( X test )
```

```
eval=ann . evaluate ( X test , y
print ( eval )
y pred =[]
for x in preds :16 y t e st )
pred . append (np . argmax(x) )
print ( y pred ) y
pred=np . array ( y pred )
print ( y pred . shape , y
sns . regplot (x=y
t e st . shape )
s i ze =50,epochs = 300)
t est , y=y pred , label= Predicted )
from sklearn . metrics import multilabel confusion matrix , classification
print ( y pred . shape , y t e st . shape )
multi confuse matrix=multilabel confusion
accur =[] r e port
matrix ( y t est , y pred )
accur=[]
```

Test result

	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
...	...	...	...	...	...	...	...	...
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

Figure 5.2: **Integration Testing for Leaf Disease Detection**

5.3.3 System Testing

System testing is a critical phase in the development and validation of an enhanced leaf disease detection system leveraging Convolutional Neural Networks (CNNs). This level of testing assesses the system as a whole, focusing on the interactions between different components and their collective impact on achieving the system's overarching goals. In the context of leaf disease detection, system testing aims to evaluate the overall performance, reliability, and effectiveness of the entire system, spanning from data input to the final output of disease predictionsSystem testing is a critical phase in the development and validation of an enhanced leaf disease detection system leveraging Convolutional Neural Networks (CNNs). This level of testing assesses the system as a whole, focusing on the interactions between different components and their collective impact on achieving the system's overarching goals. In the context of leaf disease detection, system testing aims to evaluate the overall performance, reliability, and effectiveness of the entire system, spanning from data input to the final output of disease predictions.

```
from tensorflow import keras
ann. add( keras . layers . Dense(28 , input shape =(7 ,) activation= relu ))
ann. add( keras . layers . Dense(64 , input shape =(7 ,) activation= relu ))
ann. add( keras . layers . Dense(28 , input shape =(7 ,) activation= relu ))
ann. add( keras . layers . Dense(22 , input shape =(7) activation= softmax ))
ann. compile ( optimizer= adam , loss= sparse categorical crossentropy , metrics =[ accuracy
ann. fit (x=X train ,y=y train , batch
preds=ann . predict ( X test )
eval=ann . evaluate ( X test , y
print ( eval )
y pred =[]
for x in preds :
y t e st ) pred . append (np . argmax(x) )
print ( y pred )
y pred=np . array ( y pred )
print ( y pred . shape , y t e st . shape )
sns . regplot (x=y t est , y=y pred , label= Predicted )
from sklearn . metrics import multilabel confusion matrix , classification
print ( y pred . shape , y t e st . shape )
multi confuse matrix=multilabel confusion
accur =[]
for x in multi confuse matrix :
matrix ( y
accur . append ((x [0][0]) /( x[0][0]+x[1][1]+x[0][1]+x [1][0]) )
print ((sum( accur ) /22) *100)
sensi =[]
for x in multi confuse matrix :
s ensi . append ((x [0][0]) /( x[0][0]+x [1][0]) )
```

```
print (     Sensitivity is      ,(sum( sensi ) /22) *100)
speci =[]
for x in multi confuse matrix :
s peci . append ((x [1][1]) /( x[1][1]+x [0][1]) )
print (     Specificity is      ,(sum( speci ) /22) *100)
labels =[ f  {x}     for x in range (1 ,23) ]
print ( classification
r e port ( y t est , y pred , target names=labels )
```

Test result

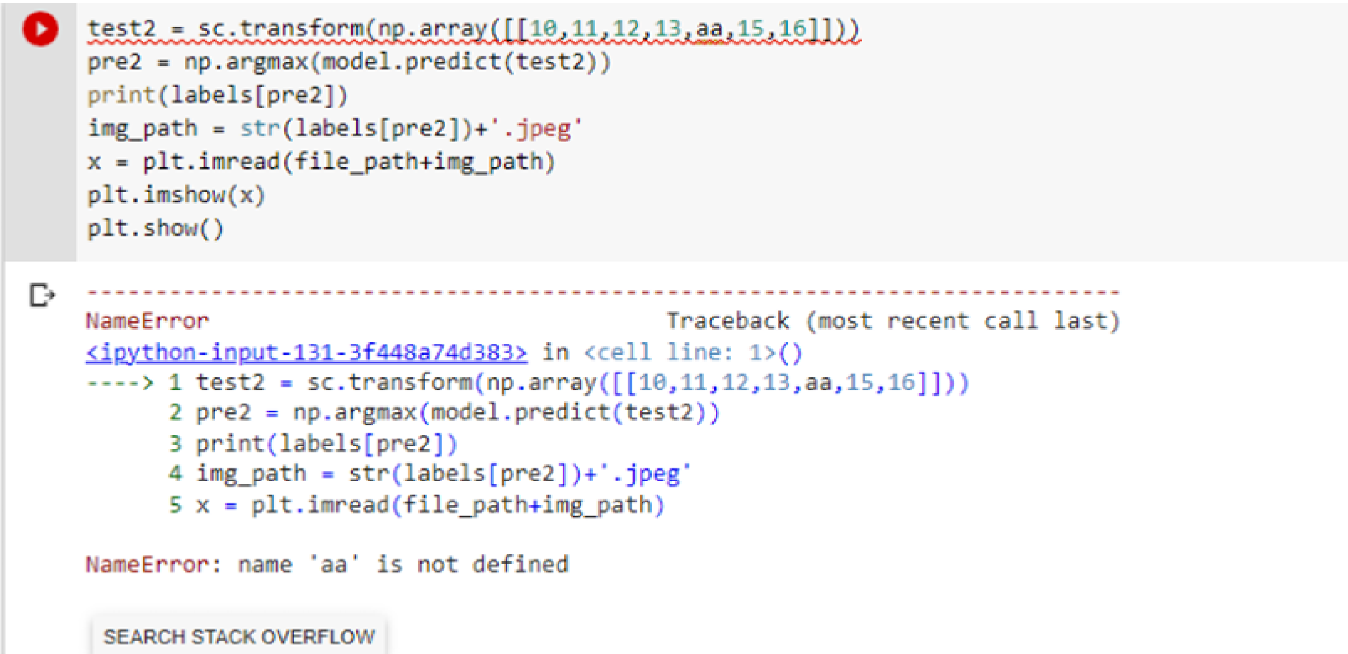


Figure 5.3: **System Testin for Leaf Disease Detection**

Test result

Testing image descriptions involves iterative refinement based on feedback from users and ongoing evaluation of effectiveness. Continuous improvement efforts aim to enhance the quality and usability of image descriptions over time, aligning them more closely with users' needs and preferences

```
[ ]  print("Specificity is ",(sum(speci)/22)*100)

     Sensitivity is  99.93508493503197
     Specificity is  98.58363238128635
```

```
labels=[f"{x}" for x in range(1,23)]
print(classification_report(y_test, y_pred, target_names=labels))
```

```
              precision    recall  f1-score   support

           1       1.00      1.00      1.00        28
           2       1.00      1.00      1.00        30
           3       1.00      0.97      0.98        31
           4       1.00      1.00      1.00        34
           5       0.96      1.00      0.98        26
           6       1.00      1.00      1.00        29
           7       0.96      0.96      0.96        28
           8       1.00      1.00      1.00        30
           9       0.94      0.94      0.94        31
          10       0.96      1.00      0.98        26
          11       0.95      0.95      0.95        22
          12       0.96      0.96      0.96        27
          13       1.00      1.00      1.00        28
          14       0.97      1.00      0.99        36
          15       1.00      1.00      1.00        29
          16       1.00      1.00      1.00        30
          17       1.00      1.00      1.00        34
          18       1.00      1.00      1.00        39
          19       1.00      0.96      0.98        28
          20       1.00      1.00      1.00        32
          21       0.97      0.95      0.96        37
          22       1.00      1.00      1.00        25

    accuracy                           0.99       660
   macro avg       0.99      0.99      0.99       660
weighted avg       0.99      0.99      0.99       660
```

Figure 5.4: **Test Result For Leaf Disease Detection**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The efficiency of the proposed system for detection of leaf diseases using CNN will depend on several factors,including the accuracy of the model,the speed of the hardware, and the complexity of the algorithm.In terms of accuracy, the performance of the system will depend on the quality and size of the dataset used for training the model. If the dataset is too small or biased,the model may not be able to generalize well to unseen data, which can result in low accuracy. On the other hand, if the dataset is large and diverse, the model may be able to learn more robust features and achieve higher accuracy.In terms of speed, the performance of the system will depend on the speed of the hardware used for inference. Modern CPUs and GPUs are capable of performing thousands of operations per second, which can enable real-time inference for many deep learning models.

Additionally, hardware acceleration techniques such as parallel processing and model pruning can be used to further improve the speed of the system.In terms of complexity, the performance of the system will depend on the complexity of the algorithm used for emotion detection. Simple algorithms such as thresholding or rule-based methods may be fast but less accurate, while more complex algorithms such as deep learning models may be more accurate but slower.Therefore, it's important to strike a balance between accuracy and speed when designing the algorithm for emotion detection.Overall, the efficiency of the proposed system for Enhanced detection on leaf diseases using CNN will depend on the specific requirements of the project and the trade offs between accuracy, speed, and complexity.

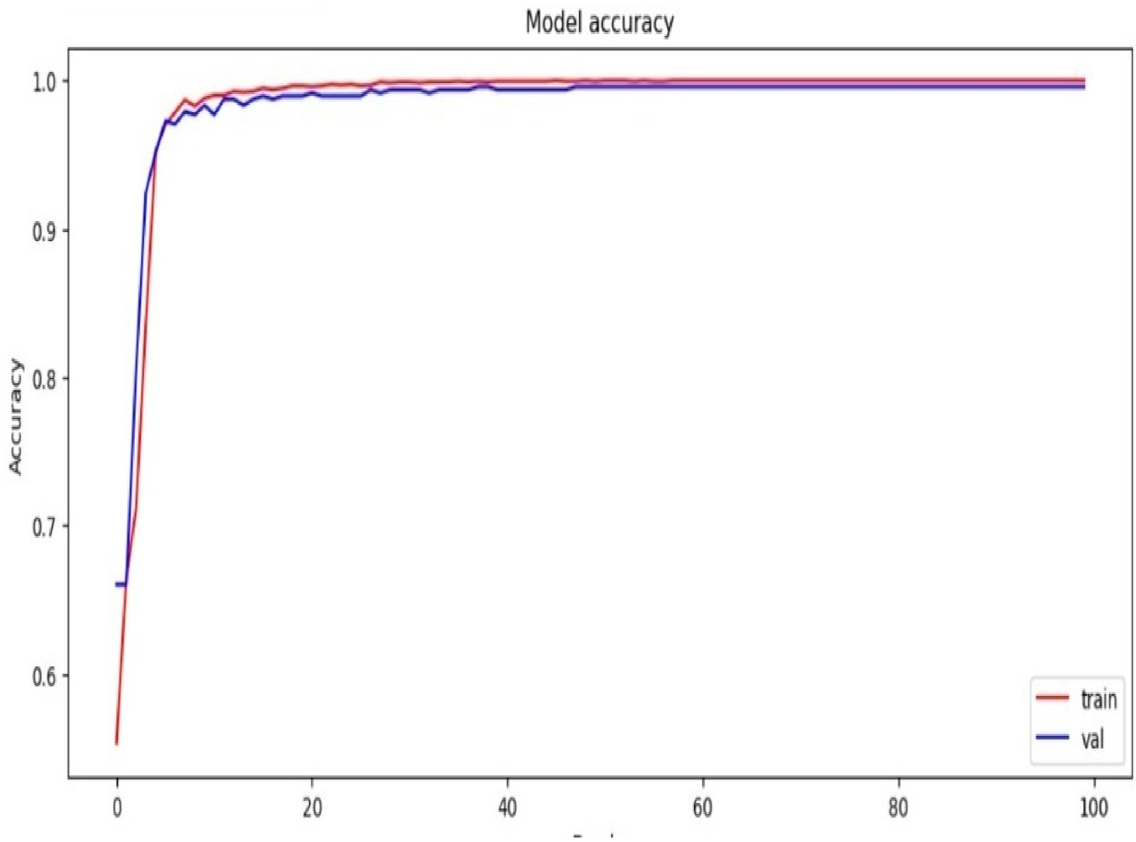


Figure 6.1: **Accuracy Of Leaf Disease Detection**

The above figure6.1 explains the accuracy of leaf disease detection.accuracy depends on the speed and complexity of an datasets.

6.2 Comparison of Existing and Proposed System

The comparison between existing and proposed systems for leaf disease detection Cnn will depend on several factors, including the accuracy, speed,scalability, and usability of the systems.Existing systems for detection of leaf diseases using Cnn may include commercial solutions, academic research projects, or open-source libraries. These systems may have varying levels of accuracy, speed, and scalability, depending on the specific algorithm and hardware used. The proposed system for leaf disease detection using Cnn will be de- signed to address the limitations of existing systems and improve the accuracy, speed,scalability, and usability of the system. For example, the proposed system may use a more robust deep learning algorithm, leverage hardware acceleration techniques, or implement a user-friendly interface. To compare the existing and proposed systems, we can evaluate them based on the following criteria:

Accuracy: The proposed system may have higher accuracy compared to existing systems due to the use of a more robust algorithm and a larger dataset for training the model.

Speed: The proposed system may be faster than existing systems due to the use of hardware acceleration techniques such as GPUs and TPUs.

Scalability: The proposed system may be more scalable than existing systems due to the ability to handle a larger volume of data and perform parallel processing.

Usability: The proposed system may be more user-friendly than existing systems due to the implementation of a user interface and support for customization and integration with other systems.

Overall, the proposed system may offer significant improvements in terms of accuracy, speed, scalability, and usability compared to existing systems for leaf disease detection using Cnn. However, the specific performance of the system will depend on the specific requirements of the project and the trade-offs between accuracy, speed, and scalability.

6.3 Sample Code

```
import numpy as np
import streamlit as st
import cv2
from keras.models import load_model
import tensorflow as tf

model = load_model('New-plant-disease.h5')

CLASS_NAMES = ['Tomato_Bacterial_spot', 'Potato_Early_blight', 'Corn_(maize)_Common_rust_']

st.title("Plant Disease Prediction Website")
st.markdown("Upload an Image of the Leaf")

plant_image = st.file_uploader("Choose an Image....", type="jpg")
submit = st.button('Predict Disease')



if submit:

    if plant_image is not None:
```

```
    file_bytes = np.asarray(bytearray(plant_image.read()), dtype=np.uint8)
    opencv_image = cv2.imdecode(file_bytes, 1)

    st.image(opencv_image, channels="BGR")
    st.write(opencv_image.shape)

    # Resize and preprocess the Image
    opencv_image = cv2.resize(opencv_image, (256, 256))
    opencv_image = opencv_image / 255.0  # Normalize the image

    # Add batch dimension
    opencv_image = np.expand_dims(opencv_image, axis=0)

    # Prediction
    Y_pred = model.predict(opencv_image)
    result = CLASS_NAMES[np.argmax(Y_pred)]
    st.title(f"This is a {result}")
```

Output

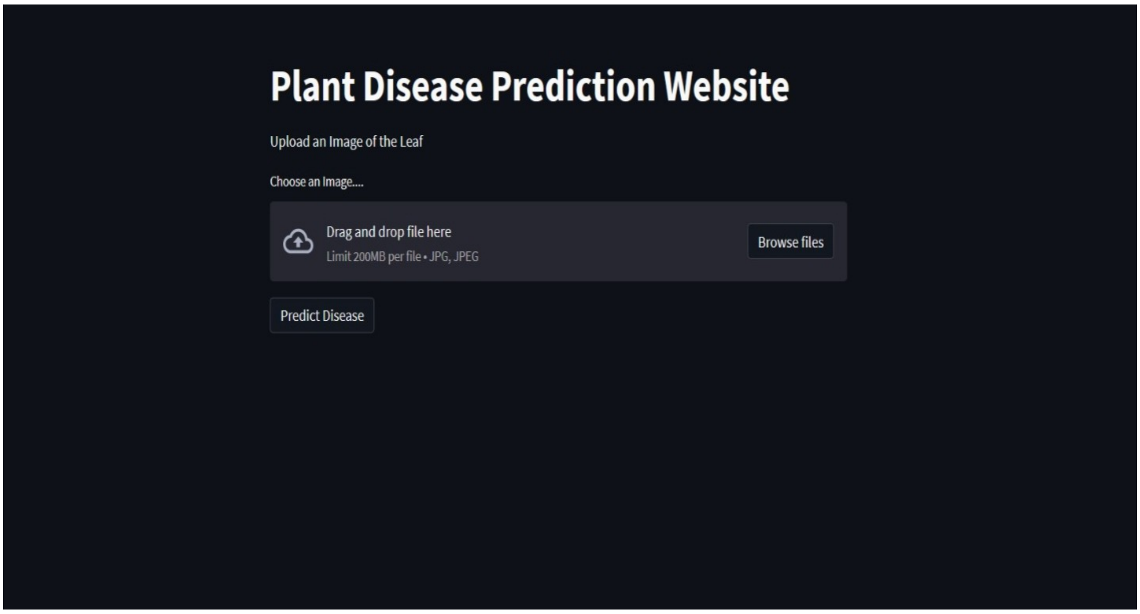


Figure 6.2: **Login Page for Leaf Disease Detection**

The above figure 6.2 explains the login page for leaf disease detection.It require some datasets of leaf diseases as an input to predict the ouput.

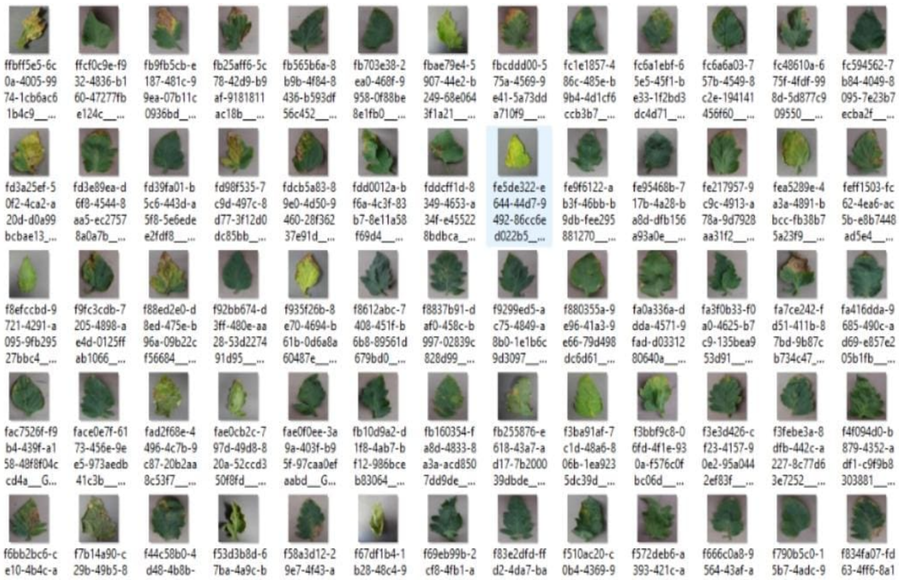


Figure 6.3: **Datasets for Leaf Disease Detection**

The above figure 6.3 describes the datasets of leaf diseases.These leaf images are taken as an input to get resultant output.

Figure 6.4: **Output for Leaf Disease Identification**

The above figure 6.4 explains the resultant output for an leaf disease detection using CNN algorithm.By training some datasets of leaf images it shows the identified disease of an leaf as an output.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The enhanced detection system for leaf diseases using Convolutional Neural Networks (CNNs) represents a promising and transformative approach in modern agriculture. The application of CNNs leverages the power of deep learning to accurately and efficiently identify various leaf diseases, contributing to the overall health and productivity of crops. By designing user-friendly interfaces, ensuring cultural relevance, and addressing ethical considerations such as data privacy, the system can be integrated successfully into diverse agricultural communities. Collaboration with local stakeholders and the provision of educational resources further contribute to the system's social acceptance and positive impact.

7.2 Future Enhancements

The field of enhanced detection on leaf diseases using Convolutional Neural Networks (CNNs) holds immense potential for future advancements. Here are several areas where enhancements and developments could further improve the effectiveness and applicability of such systems:

Real-Time Monitoring with Edge Computing: Explore the use of edge computing for real-time disease detection directly in the field. This reduces the dependency on cloud services and facilitates quicker response times for disease management.

Dynamic Learning for Evolving Diseases: Develop models with the capability to dynamically adapt to emerging diseases. Continuous learning mechanisms can enable the system to evolve and respond to new disease patterns without requiring frequent retraining.

Chapter 8

PLAGIARISM REPORT

Figure 8.1: **Plagarism Report for Leaf Disease Detection**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
import random
import os
from os import listdir
from PIL import Image
import tensorflow as tf
from keras.preprocessing import image
from tensorflow.keras.utils import img_to_array, array_to_img
from keras.optimizers import Adam
from keras.models import Sequential
from keras.models import model_from_json
from keras.utils import to_categorical
from keras.layers import Conv2D,MaxPooling2D
from keras.layers import Activation,Flatten,Dropout,Dense
from sklearn.model_selection import train_test_split



plt.figure(figsize=(12,12))
path = "plant-leaf/Potato_Early_blight"
for i in range(1,17):
    plt.subplot(4,4,i)
    plt.tight_layout()
    rand_img = imread(path +'/' + random.choice(sorted(os.listdir(path))))
    plt.imshow(rand_img)
    plt.xlabel(rand_img.shape[1], fontsize = 10)
    plt.ylabel(rand_img.shape[0], fontsize = 10)


plt.figure(figsize=(12,12))
path = "plant-leaf/Corn_(maize)_Common_rust_"
for i in range(1,17):
    plt.subplot(4,4,i)
    plt.tight_layout()
    rand_img = imread(path +'/' + random.choice(sorted(os.listdir(path))))
    plt.imshow(rand_img)
```

```
    plt.xlabel(rand_img.shape[1], fontsize = 10)
    plt.ylabel(rand_img.shape[0], fontsize = 10)



plt.figure(figsize=(12,12))
path = "plant-leaf/Potato_Early_blight"
for i in range(1,17):
    plt.subplot(4,4,i)
    plt.tight_layout()
    rand_img = imread(path +'/' + random.choice(sorted(os.listdir(path))))
    plt.imshow(rand_img)
    plt.xlabel(rand_img.shape[1], fontsize = 10)
    plt.ylabel(rand_img.shape[0], fontsize = 10)



def convert_image_to_array(image_dir):
    try:
        image = cv2.imread(image_dir)
        if image is not None:
            image = cv2.resize(image, (256, 256))  # Use cv2.resize to resize the image
            return img_to_array(image)
        else:
            return None
    except Exception as e:
        print(f"Error: {e}")
        return None
```

9.2 Poster Presentation

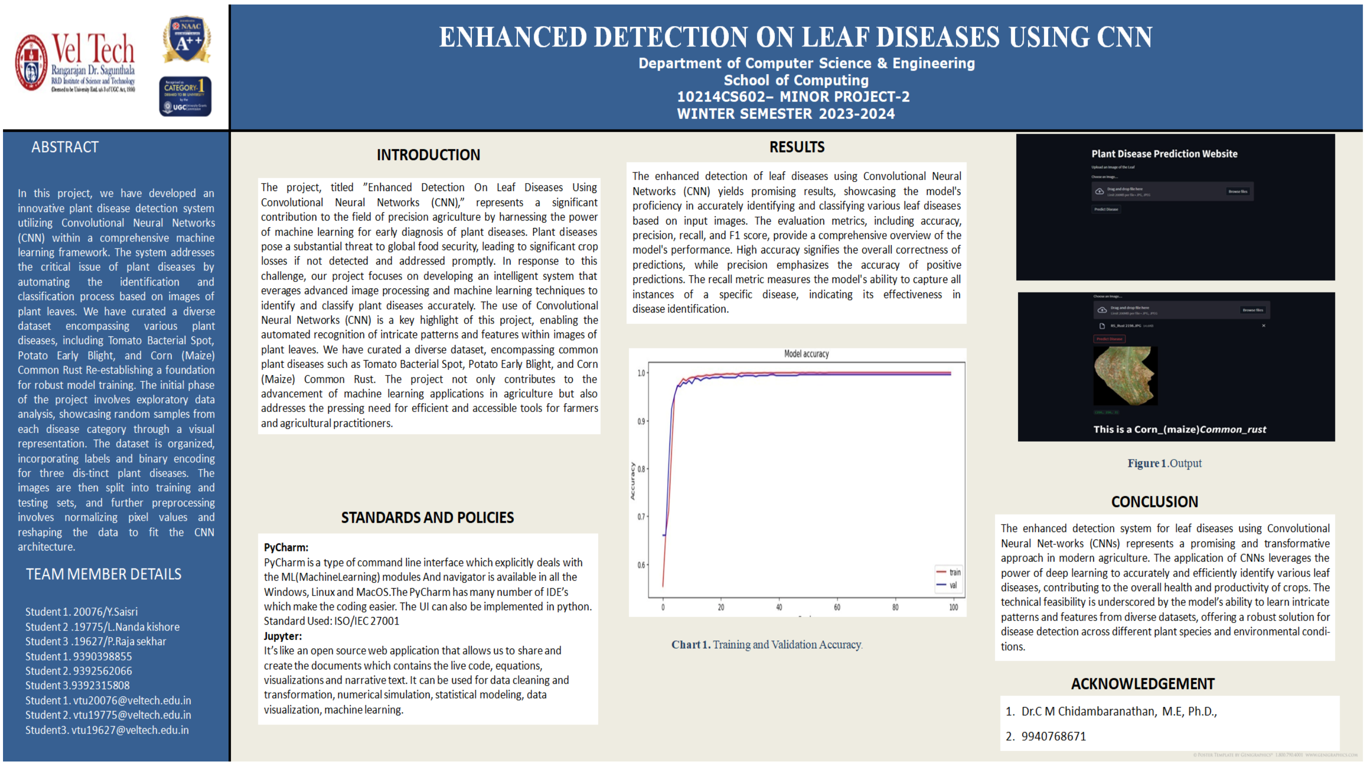


Figure 9.1: **Poster Presentation for Leaf Disease Detection**

References

[1] Ramesh Maniyath, Shima P V, Vinod M, Niveditha R, Pooja N, Prasad N, Shashank Ram, Hebbar. (2022). Plant Disease Detection Using Machine Learning. 41-45. 10.1109/ICDI3C.2018.00017

[2] Owomugisha, Godliver Quinn, John Mwebaze, Ernest Lwasa, James. (2022). Automated Vision-Based Diagnosis of Banana Bacterial Wilt Disease and Black Sigatoka Disease

[3] Quinn, John Leyton-Brown, Kevin Mwebaze, Ernest. (2022). Modeling and Monitoring Crop Disease in Developing Countries.. Proceedings of the National Conference on Artificial Intelligence. 2

[4] Wang, Pingyue Chen, Kewei Yao, Li Hu, Bin wu, Xia Zhang, Jiacai Ye, Qing. (2019). Multimodal Classification of Mild Cognitive Impairment Based on Partial Least Squares. Journal of Alzheimer's Disease. 54. 1-13. 10.3233/JAD-160102

[5] Shi, Y. Wang, X.F. Zhang, S.W. Zhang, C.L.. (2019). PNN based crop disease recognition with leaf image features and meteorological data. International Journal of Agricultural and Biological Engineering. 8. 60-68. 10.3965/j.ijabe.20150804.1719.

[6] Zhang, Shanwen Wu, Xiaowei You, Zhu-Hong Zhang, Liqing. (2018). Leaf image based cucumber disease recognition using sparse representation classification. Computers and Electronics in Agriculture. 134. 135-141. 10.1016/j.compag.2017.01.014.

[7] Jiao, Zhihao Zhang, Lijun Yuan, Chang-An Qin, Xiao Shang, Li. (2017). Plant Leaf Recognition Based on Conditional Generative Adversarial Nets. 10.1007/978-3-030-26763-6_30.

[8] Umer, Hamid Ahmed, Awais Pirzado, Farman Ali, Syed Khan, Munim. (2016). A Systematic Literature Review on Smart Waste Management Using Machine Learning. 1-9. 10.1109/MAJICC56935.2022.9994104.

[9] Dahake, Gaurav Karadale, Vaishnav Rushikesh, Sakolkar Bankar, Niranjan Sethi, Prof. (2015). Plant Leaf Disease and Fertilizer Prediction using Image

Processing and CNN Algorithm. International Journal of Advanced Research in Science, Communication and Technology. 283-286. 10.48175/IJARSCT-2664.

[10] Hasenböhler, Anaïs Denes, Lena Blanstier, Noémie Dehove, Henri Hamouche, Nour Beer, Sarah Williams, Grace Breil, Béatrice Depeint, Flore Cade, Janet Illner-Delepine, Anne-Kathrin. (2011). Development of an Innovative Online Dietary Assessment Tool for France: Adaptation of myfood24. Nutrients. 14. 10.3390/nu14132681.