A

Mini Project

On

PLANT LEAF DISEASE PREDICTION USING DEEP LEARNING

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

V. Sindhu Bhargavi (217R1A05R7)

S. Sruthi (227R5A0525)

Under the Guidance of

M. MADHUSUDHAN

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2021-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
GROUP OF INSTITUTIONS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled "**PLANT LEAF DISEASE PREDICTION USING DEEP LEARNING"** being submitted by **V. SINDHU BHARGAVI (217R1A05R7), S. SRUTHI (227R5A0525)** in partial fulfillment of the requirements for the awardof the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2024-25.

The results embodied in this project have not been submitted to any other University orInstitute for the award of any degree or diploma.

M. Madhusudhan
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. N. Bhaskar
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ______________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on theencouragement and guidelines of many others. We take this opportunity to express ourgratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to our guide **Mr. M. Madhusudhan,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project ReviewCommittee (PRC) Coordinators: **Dr. J. Narasimha Rao, Dr. K. Maheshwari, Mr. K. Ranjith Reddy, Mrs. K. Shilpa** for their cordial support, valuable information and guidance, which helped using completing this task through various stages.

We are also thankful to **Dr. N. Bhaskar,** Head of the Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy,**Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus**who contributed to the completion of the project. We are grateful for their constant support andhelp.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

V. SINDHU BHARGAVI (217R1A05R7)

S. SRUTHI (217R1A0525)

ABSTRACT

Plant diseases threaten the world's farmers, consumers, ecosystems, and economy. In India, farmers lose 35% of their crops due to pests and diseases. Another significant health risk is the reckless use of pesticides, as many of them are toxic and biomagnified. Additional measures to mitigate these negative impacts might include early disease detection, crop monitoring, and individualized treatments. Experts in agriculture can usually identify a disease by looking at its symptoms. However, farmers have a hard time getting in touch with professionals. Our technology is the first of its kind to allow for the automated analysis, monitoring, and prediction of diseases via collaboration and integration. A smartphone app allows farmers to take pictures of sick plant portions, which helps with diagnosis and treatment. The use of state-of-the-art AI algorithms for analyzing images in the cloud enables real-time prediction. Using user-uploaded photographs and expert advice, the AI model is continuously learning to improve its accuracy. Also, local experts may be contacted by farmers via the website. To aid in the development of preventive measures, maps of infection density with unfold predictions are produced using a cloud-based database of geo-tagged images and micro-climatic information. Using a web-based interface, experts may do disease analytics with geographical visualizations. We implemented convolutional neural networks (CNNs) trained on large disorder datasets consisting of plant images collected autonomously from several farms over the course of seven months into aour analysis. Plant pathologists double-checked the outcomes after using the automatic CNN model to diagnose test images. A level of disease detection accuracy higher than 95% was achieved. Our solution is a provider in the cloud that can help experts and farmers manage agricultural plant diseases in an eco-friendly way. It is unique, extensible, and easy to operate.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The Plant Leaf Disease Prediction project aims to develop a machine learning model that detects plant diseases from leaf images. The process involves collecting and labeling a dataset of healthy and diseased leaf images, preprocessing them through techniques like resizing and augmentation, and training models such as CNNs or using transfer learning. The model's performance will be evaluated using metrics like accuracy and precision. The project will also include deploying the model in a web or mobile app for real-time disease prediction.

1.2 PROJECT PURPOSE

The purpose of the Plant Leaf Disease Prediction project is to develop a system that enables early detection of diseases in plants by analyzing leaf images. This will help farmers and agricultural professionals to quickly identify diseases, allowing for timely interventions that can reduce crop damage and improve yield. By using machine learning models, the project aims to automate the detection process, making it faster, more accurate, and accessible through a web or mobile application. Ultimately, the project seeks to enhance agricultural productivity and minimize losses due to plant diseases, supporting sustainable farming practices.

1.3 PROJECT FEATURES

The Plant Leaf Disease Prediction project will enable users to upload leaf images and receive real-time disease detection results. It will feature a machine learning model trained on a large dataset, optimized for high accuracy, and deployed through a web or mobile interface. The system will preprocess images to improve detection, provide disease information, and offer cloud-based scalability for quick predictions. User-friendly documentation and potential for future integration into agricultural systems are also key features.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The problem of plant leaf disease detection is critical in agriculture, as early identification and treatment of diseases are essential to preventing crop loss and ensuring food security. Traditional methods of disease detection are often manual, time-consuming, and reliant on expert knowledge, which may not be readily available to all farmers, particularly in remote areas. This leads to delays in identifying and managing plant diseases, resulting in reduced crop yields, increased use of pesticides, and higher costs for farmers. The goal of this project is to develop an automated system that can accurately detect diseases in plant leaves using machine learning models. By analyzing images of plant leaves, the system will be able to identify signs of diseases at an early stage, helping farmers and agricultural professionals take timely action. The model needs to be able to classify different types of plant diseases basedon visual symptoms and provide real-time predictions.

2.2 EXISTING SYSTEM

In the existing landscape of plant leaf disease prediction, traditional methods heavily rely on manual visual inspection by agricultural experts to detect and diagnose diseases. These methods involve physically examining individual plant leaves for symptoms such as discoloration, lesions, spots, or other abnormalities indicative of disease presence. While human expertise can be valuable, this approach is time-consuming, labor - intensive, and subject to human error. Moreover, it may not always be feasible to accurately diagnose diseases at early stages or in large-scale agricultural fields. Some existing automated systems for plant disease detection utilize image processing techniques to analyze digital images of plant leaves and identify disease symptoms. These systems often employ simple image processing algorithms to extract features such as colour, texture, and shape from leaf images and classify them into different disease categories using predefined rules or thresholds. While these approaches represent a step towards automation, they may lack the robustness and accuracy required for reliable disease prediction, particularly in the presence of varying environmental conditions and disease manifestations.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- High Human Error
- Limited Early Detection
- Scalability Issues
- Time Consuming
- Implementation and Maintenance
- Computational Cost

2.3 PROPOSED SYSTEM

In our proposed system for plant leaf disease prediction, we aim to address the limitations of existing methods by leveraging advanced computer vision techniques and machine learning algorithms. Our system is designed to automate the process of disease detection and prediction, enabling early and accurate diagnosis of plant diseases to facilitate timely intervention and management practices in agriculture. The core component of our proposed system is the use of deep learning models, particularly convolutional neural networks (CNNs), for analyzing plant leaf images and identifying disease symptoms. CNNs have demonstrated remarkable capabilities in image recognition tasks and can automatically learn and extract relevant features from raw image data, making them well-suited for plant disease prediction. Additionally, we employ transfer learning techniques to fine-tune pre-trained CNN models on plant leaf disease datasets, allowing for efficient training and adaptation to specific disease patterns. Our system incorporates a comprehensive database of labeled plant leaf images, covering a wide range of disease types and severity levels. This dataset serves as the foundation for training and evaluating our deep learning models, ensuring robust performance across diverse disease manifestations. We also integrate data augmentation techniques to enhance model generalization and robustness, allowing our system to accurately predict diseases even in the presence of variations in lighting conditions, leaf orientation, and background clutter.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- High Accuracy
- Scalability
- Early Detection
- User-Friendly Interface
- Minimum Time Required
- Real-Time Moni

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and developmentof the system is limited. The expenditures must be justified. Thus the developed system as wellwithin the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system; instead, they must accept it as a necessity. The level of acceptance by the users solely depends on the methods employed to educate them about the system and to make them familiar with it. Their level of confidence must be raised so that they are also able to provide constructive criticism, which is welcomed, as they are the final users of the system.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Dual Core@ CPU 2.90GHz.
- Hard disk : 16GB and Above.
- Memory : 4GB RAM and Above.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and softwarecomponents of the system. The following are some software requirements.

- Operating system : Windows 8 and Above.
- Languages : Python (Version 3.7.0)

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for plant leaf diseaseprediction using deep learning, starting from input to final classification.

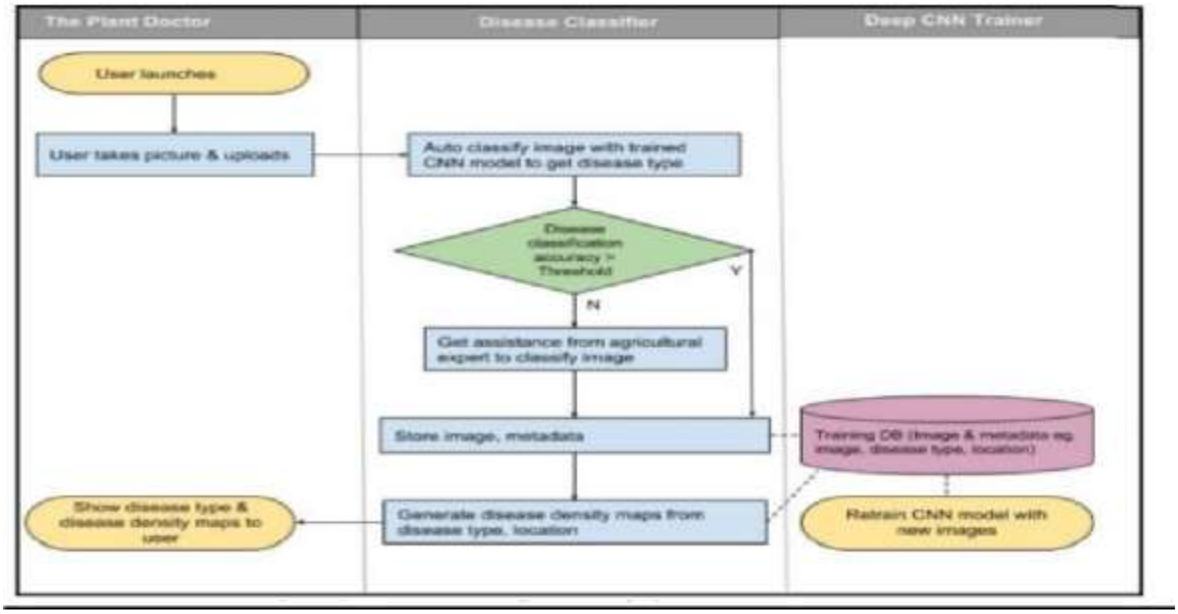


Figure 3.1: Project Architecture of Plant Leaf Disease Prediction Using Deep Learning

3.2 DESCRIPTION

The Plant Doctor: The user launches the application. The user takes a picture of the plant and uploads it to the system. After classification, the system shows the disease type and generates disease density maps to the user.

Feature Extraction: Feature extraction in plant leaf disease prediction involves identifying and isolating key characteristics, such as color, texture, and patterns, from leaf images to help the model accurately classify diseases.

Disease Classifier: The system uses a trained CNN model to automatically classify the plant image and identify the disease type. If the classification accuracy is below a certain threshold, an agricultural expert assists in classifying the image manually. The image and related metadata (like disease type and location) are stored in the system.

Deep CNN Trainer: The system uses the stored images and metadata to update its database (DB). It retrains the CNN model with the new data to improve the accuracy of classification.

3.3 USE CASE DIAGRAM

In A use case diagram represents the functional aspects of a system, showing how users (or external systems) interact with it. For a plant leaf disease prediction based on deep learning, the primary actors (users)

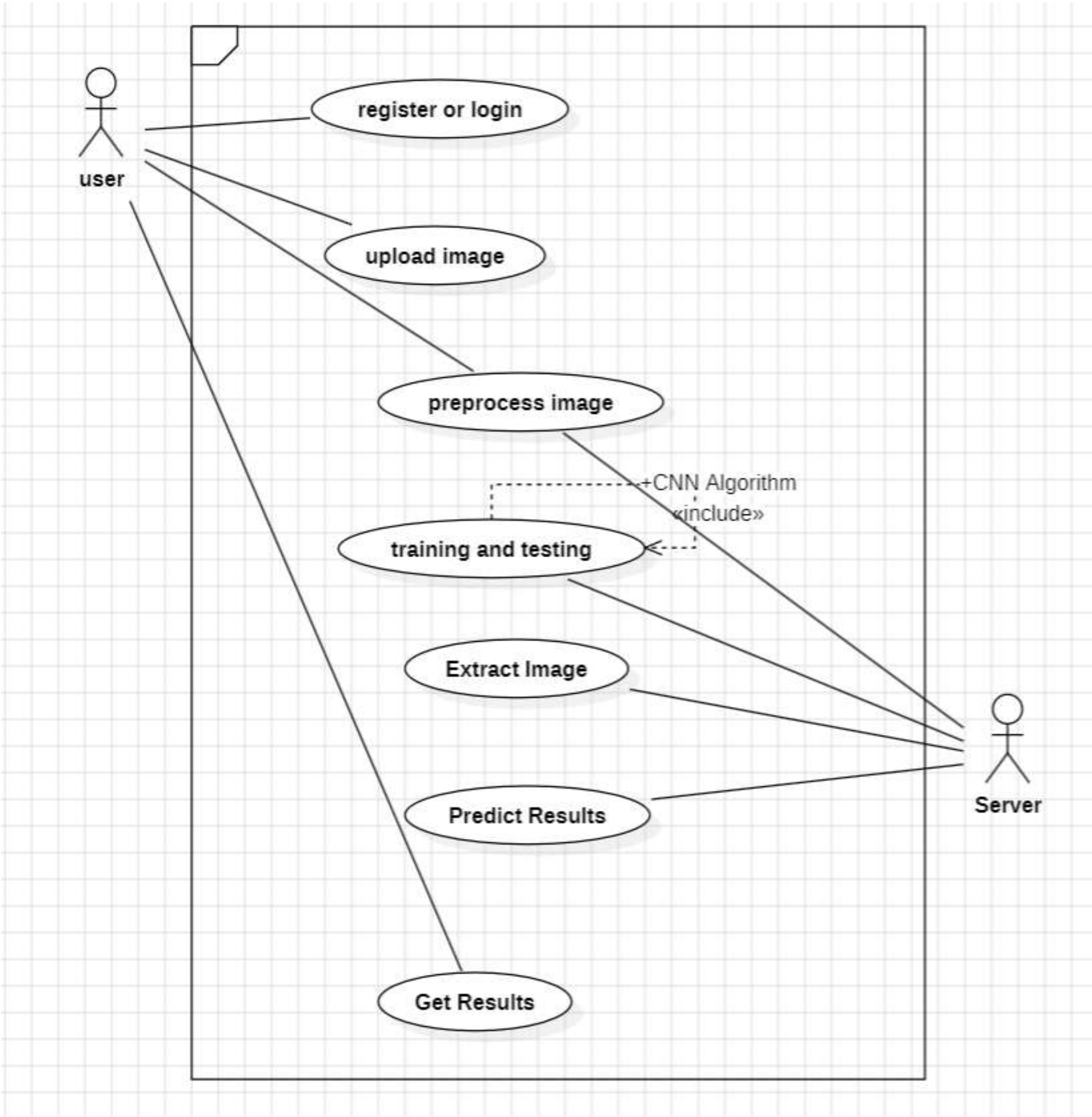


Figure 3.3: Use Case Diagram for Plant Leaf Disease Prediction Using Deep Learning

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects. In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

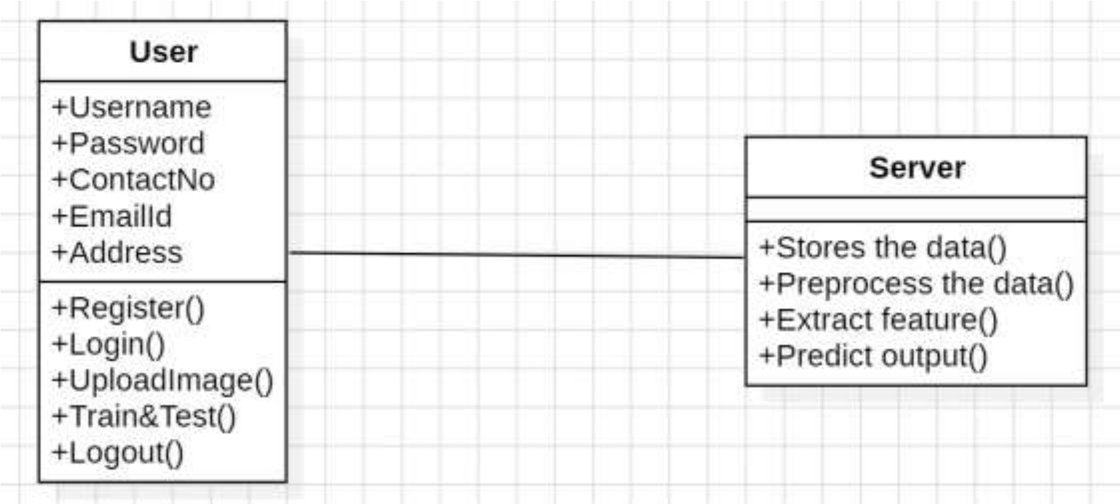


Figure 3.4: Class Diagram for User and Server for Plant Leaf Disease Prediction Using Deep Learning

3.5 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagramthat shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

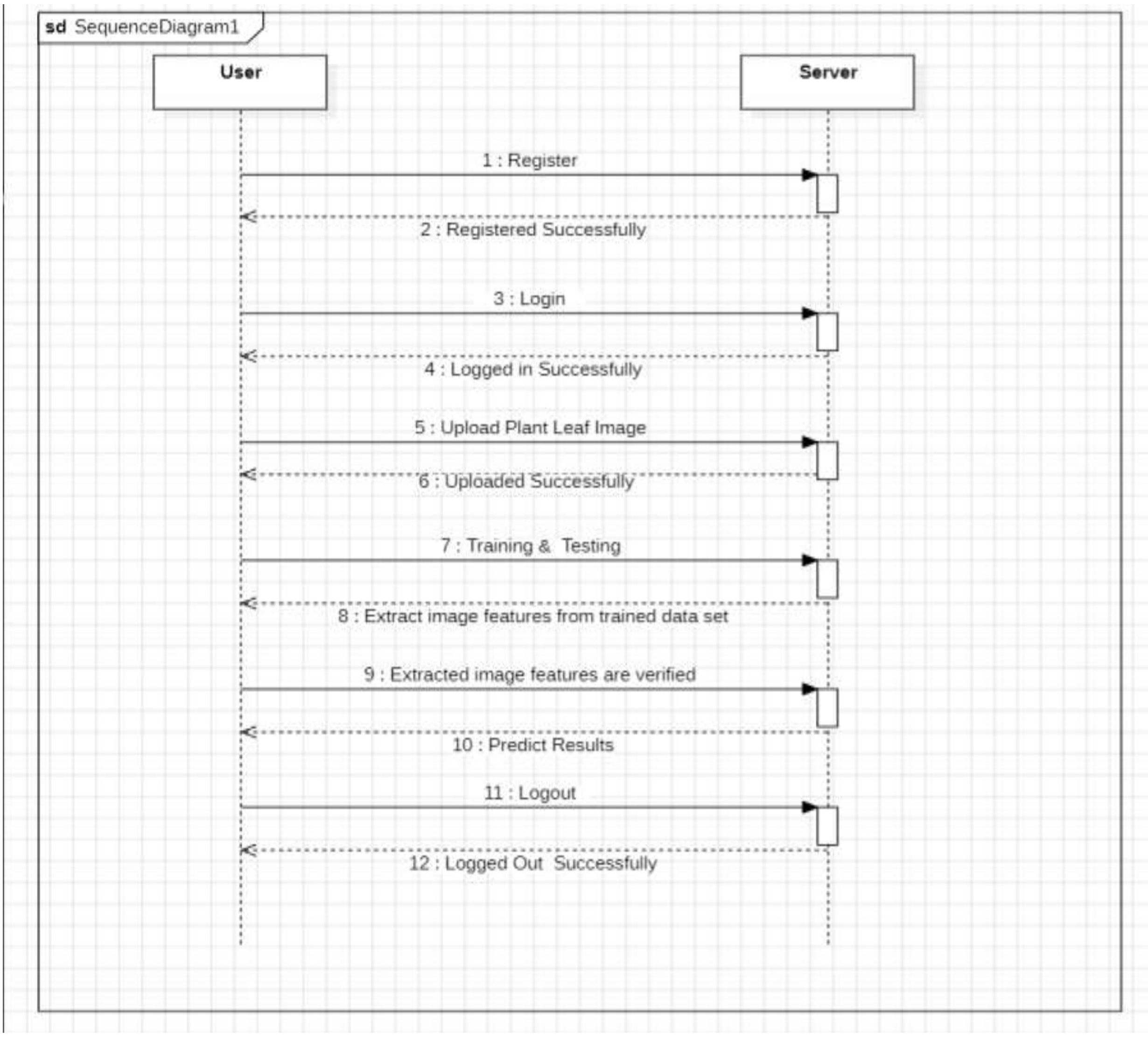


Figure 3.5: Sequence Diagram for User and Server for Plant Leaf Disease Prediction Using Deep Learning

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and the actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflowsof components in a system. An activity diagram shows the overall flow of control.

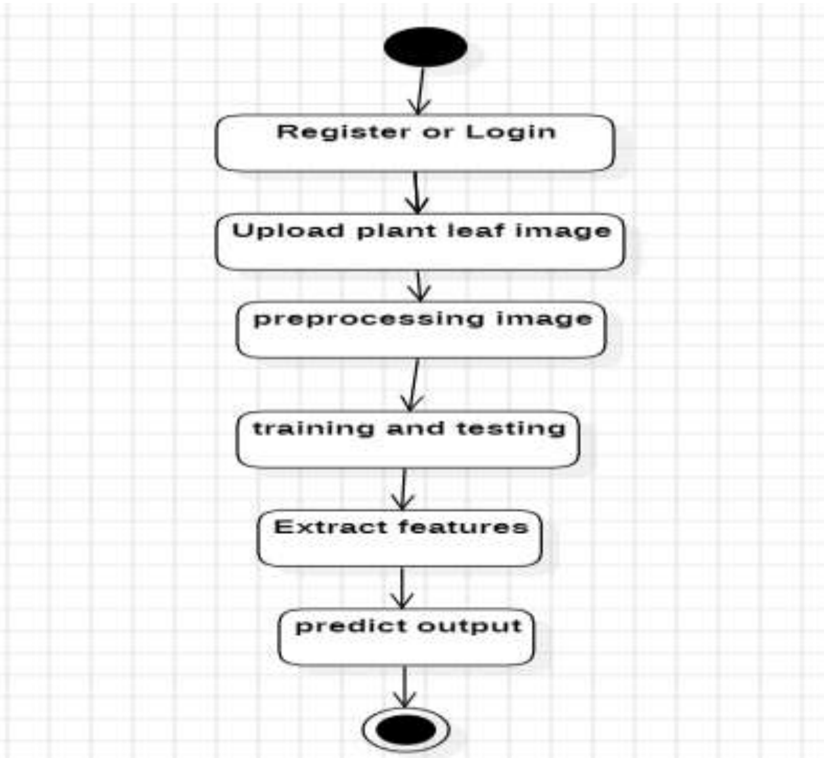


Figure 3.6: Activity Diagram for User and System for Plant Leaf Disease Prediction Using Deep Learning

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
import numpy as np
import os
from keras.models import model_from_json
from keras.utils.np_utils import to_categorical
import cv2
from sklearn.model_selection import train_test_split
from keras.layers import Input
from keras.models import Model
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
import keras

plants = ['Pepper__bell___Bacterial_spot', 'Pepper__bell___healthy', 'Potato___Early_blight',
'Potato___healthy', 'Potato___Late_blight', 'Tomato_Bacterial_spot', 'Tomato_Early_blight',
'Tomato_healthy', 'Tomato_Late_blight', 'Tomato_Leaf_Mold','Tomato_Septoria_leaf_spot',
'Tomato_Spider_mites_Two_spotted_spider_mite', 'Tomato_Target_Spot','Tomato_
Tomato_mosaic_virus', 'Tomato_Tomato_YellowLeaf_Curl_Virus']

X = []
Y = []
'''
for i in range(len(plants)):
    for root, dirs, directory in os.walk('PlantVillage/'+plants[i]):
        for j in range(len(directory)):
            img = cv2.imread('PlantVillage/'+plants[i]+"/"+directory[j])
            img = cv2.resize(img, (64,64))
            im2arr = np.array(img)
            im2arr = im2arr.reshape(64,64,3)
```

```
        X.append(im2arr)
        Y.append(i)
        print('PlantVillage/'+plants[i]+"/"+directory[j])

np.save("model/myimg_data.txt",X)
np.save("model/myimg_label.txt",Y)

X = np.load("model/myimg_data.txt.npy")
Y = np.load("model/myimg_label.txt.npy")

X = np.asarray(X)
Y = np.asarray(Y)
print(X.shape)
Y = to_categorical(Y)
print(Y.shape)
img = X[20].reshape(64,64,3)
cv2.imshow('ff',cv2.resize(img,(250,250)))
cv2.waitKey(0)
print("shape                ==
"+str(X.shape)) print("shape ==
"+str(Y.shape))print(Y)
X = X.astype('float32')
X = X/255

indices = np.arange(X.shape[0])
np.random.shuffle(indices)
X = X[indices]
Y = Y[indices]
classifier = Sequential() #alexnet transfer learning code here
classifier.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))
classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))
```

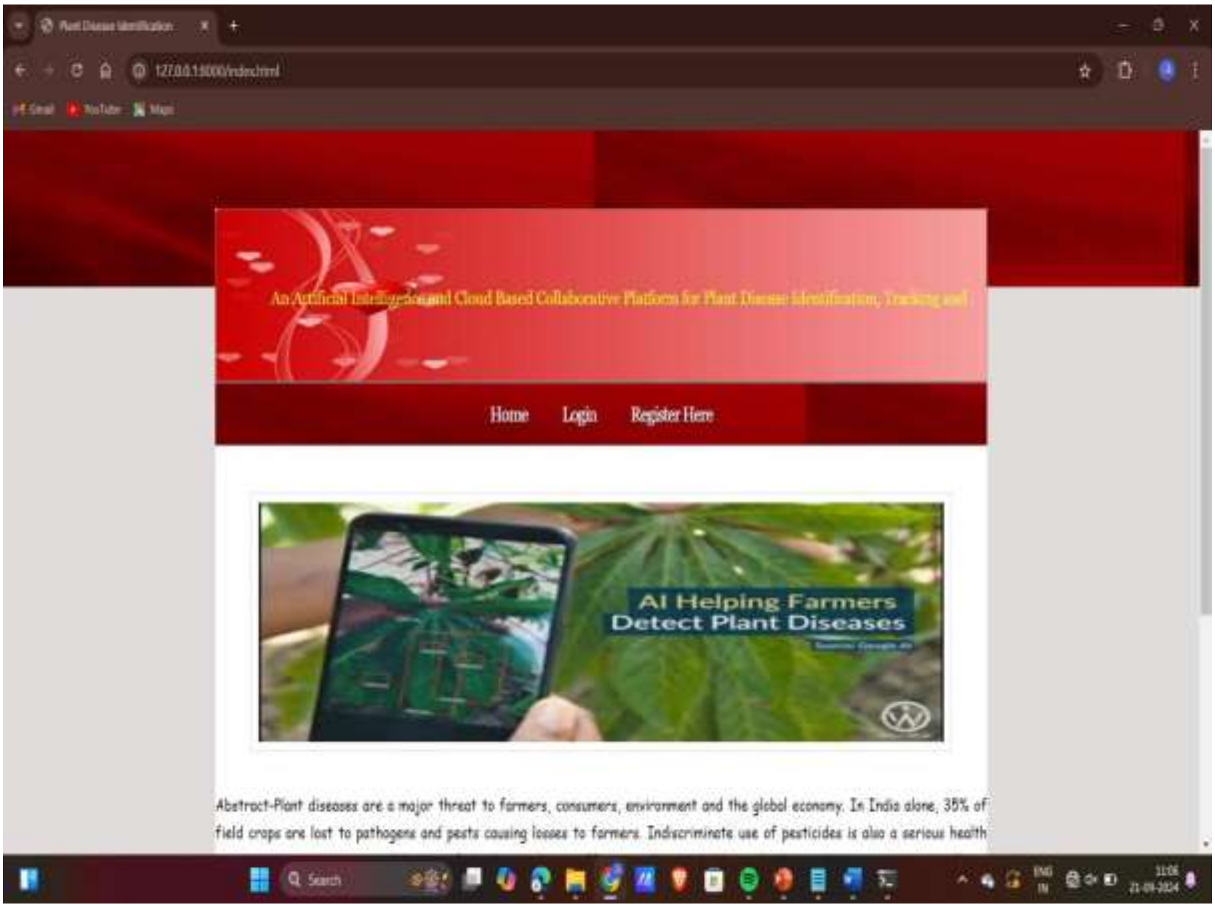
```
classifier.add(Dense(output_dim = 15, activation = 'softmax'))
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
print(classifier.summary())
classifier.fit(X, Y, batch_size=16, epochs=10, validation_split=0.2, shuffle=True, verbose=2)
classifier.save_weights('model/model_weights.h5')
model_json = classifier.to_json()
with open("model/model.json", "w") as json_file:json_file.write(model_json)
'''
with open('model/model.json', "r") as json_file:
    loaded_model_json = json_file.read()
 loaded_model = model_from_json(loaded_model_json)

 loaded_model.load_weights("model/model_weights.h5")
 loaded_model._make_predict_function()
 print(loaded_model.summary())

img = cv2.imread('download.jpg')
img = cv2.resize(img, (64,64))
im2arr = np.array(img)
im2arr = im2arr.reshape(1,64,64,3)
X = np.asarray(im2arr)
X = X.astype('float32')
X = X/255
preds = loaded_model.predict(X)
print(str(preds)+" "+str(np.argmax(preds)))
predict = np.argmax(preds)
print(plants[predict])
img = im2arr.reshape(64,64,3)
cv2.imshow(plants[predict],cv2.resize(img,(250,250)))
cv2.waitKey(0)
```

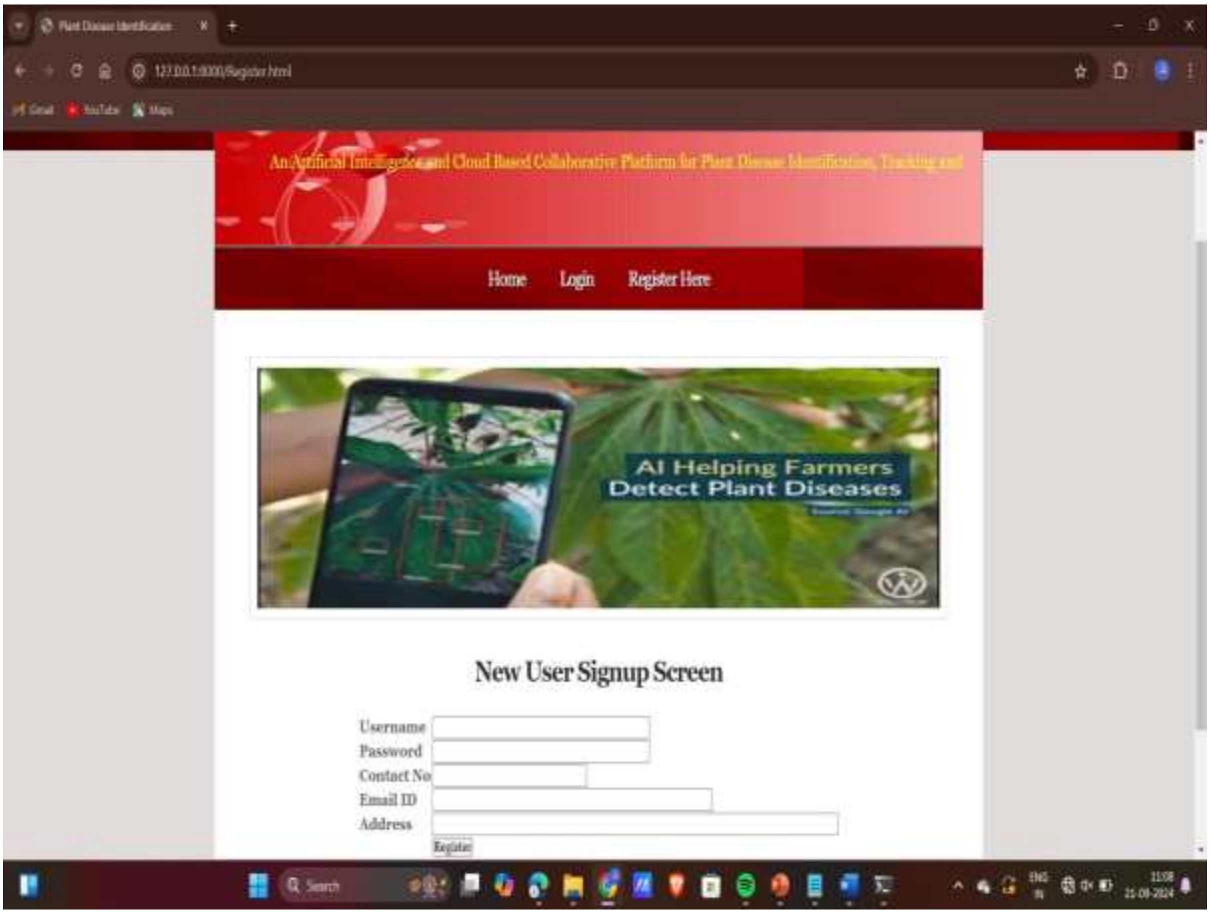
5. SCREENSHOTS

5.1 Index.html



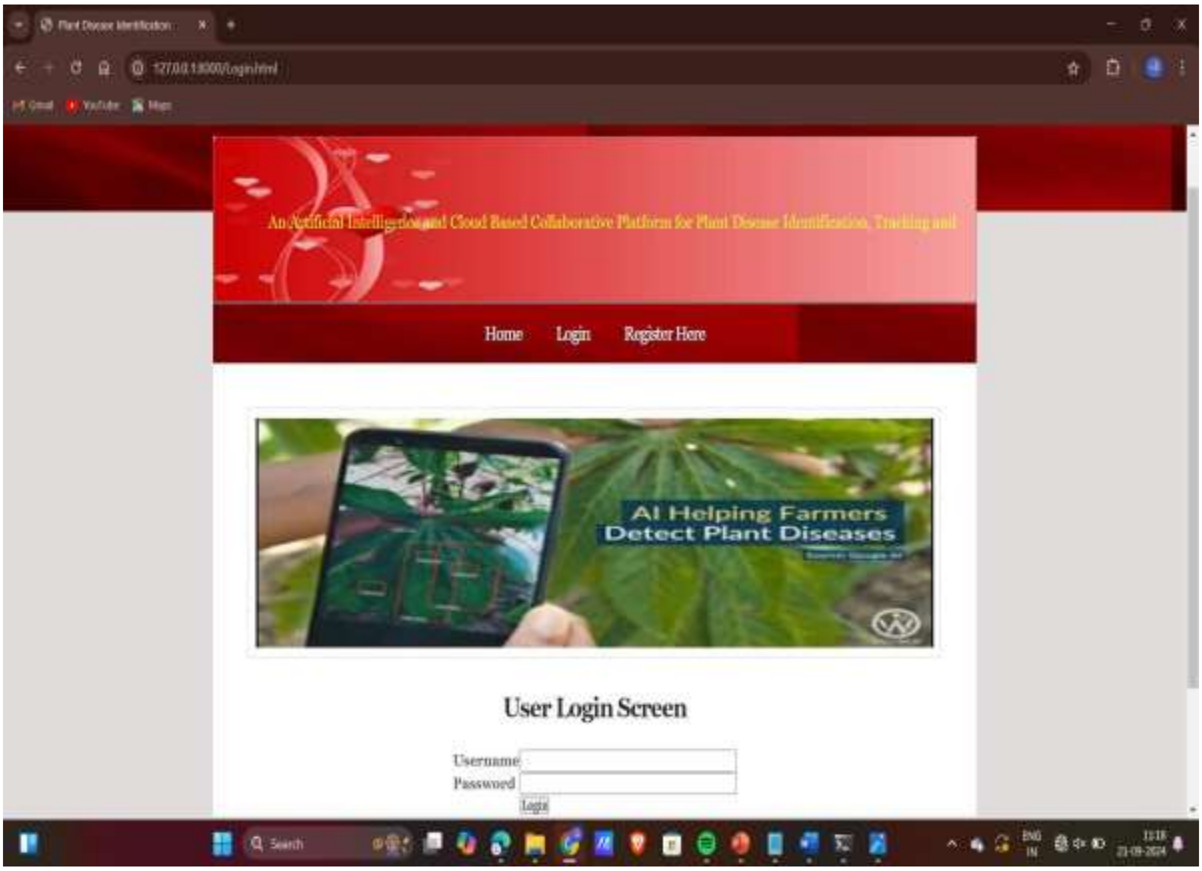
Screenshot 5.1: Index Screen of Plant Leaf Disease Prediction Using Deep Learning

5.2 Register.html



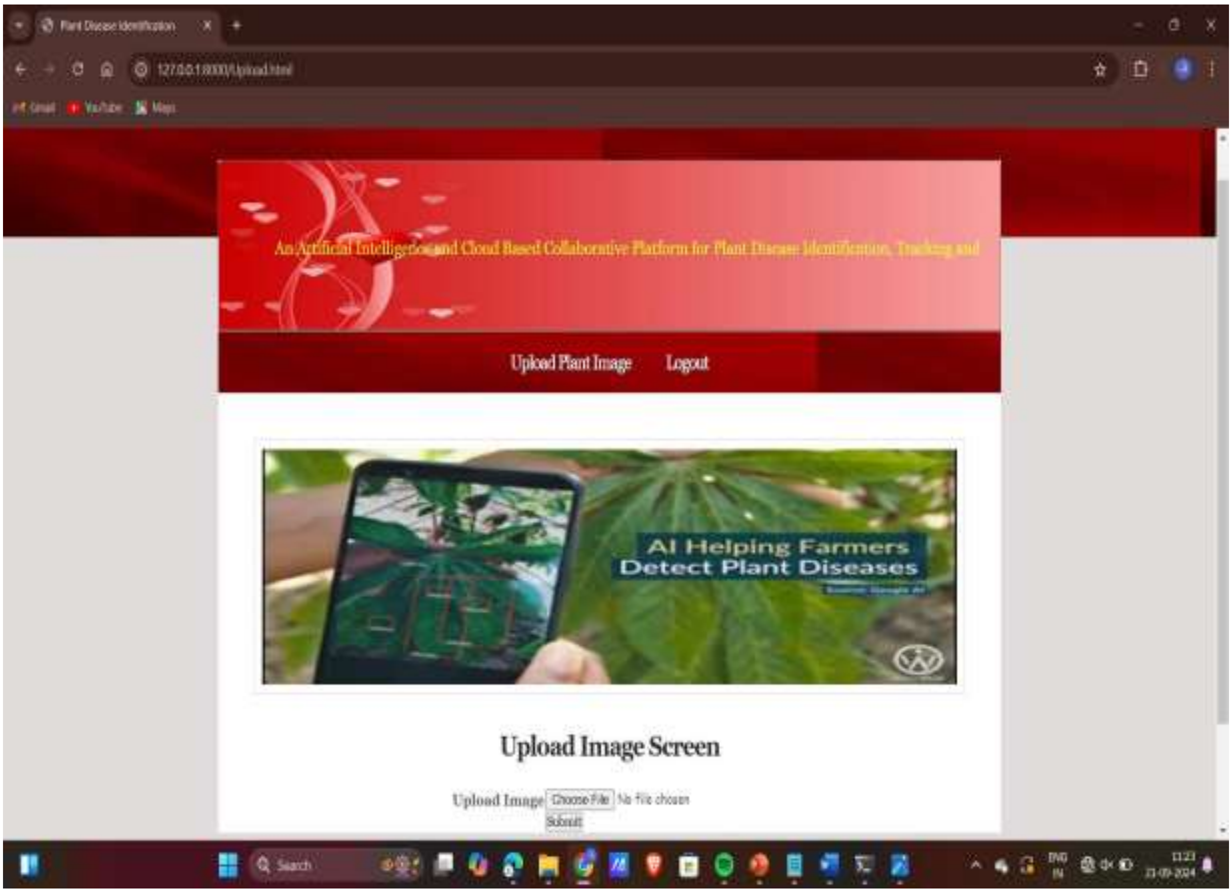
Screenshot 5.2: Screenshot of User Signup Screen for Plant Leaf Disease Prediction Using DeepLearning

5.3 Login.html



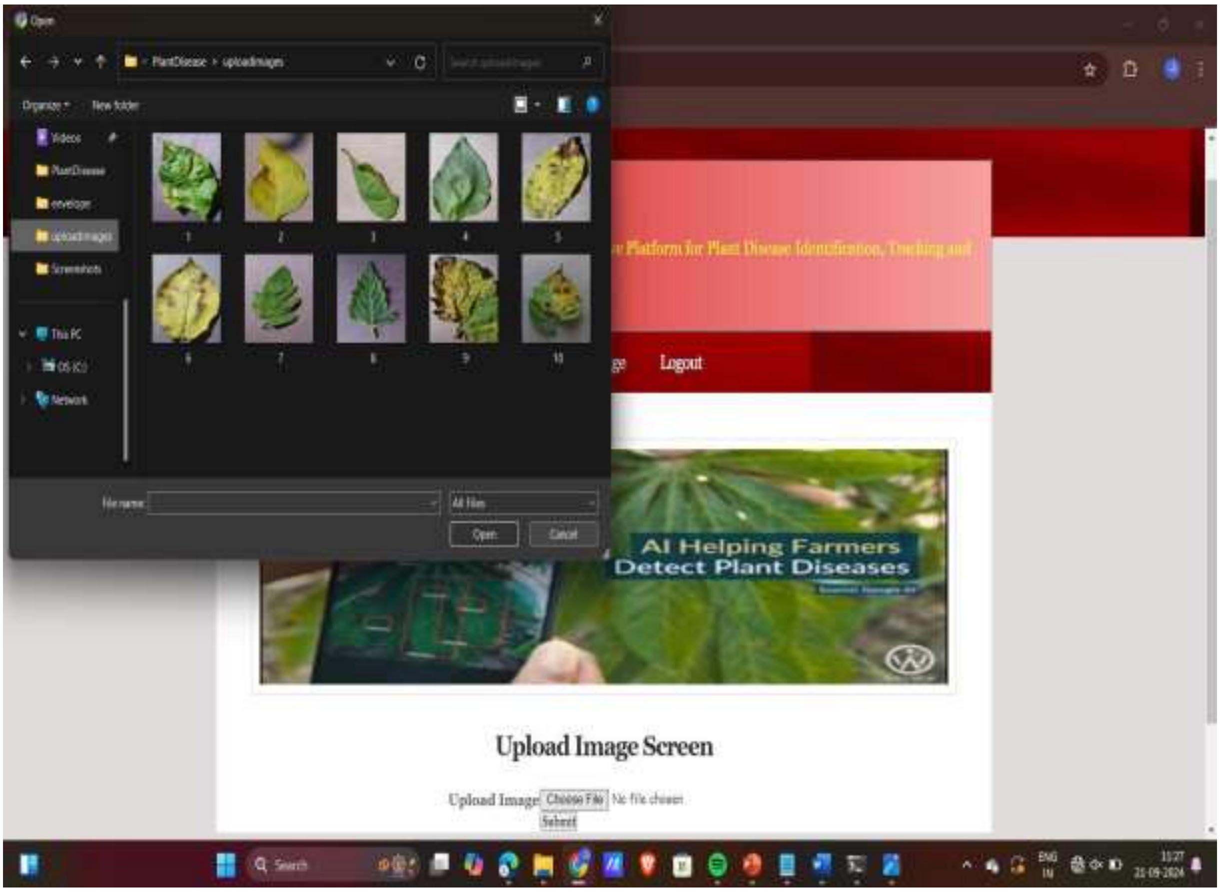
Screenshot 5.3: Screenshot of User Login Screen for Plant Leaf Disease Prediction Using Deep Learning

5.4 Upload.html



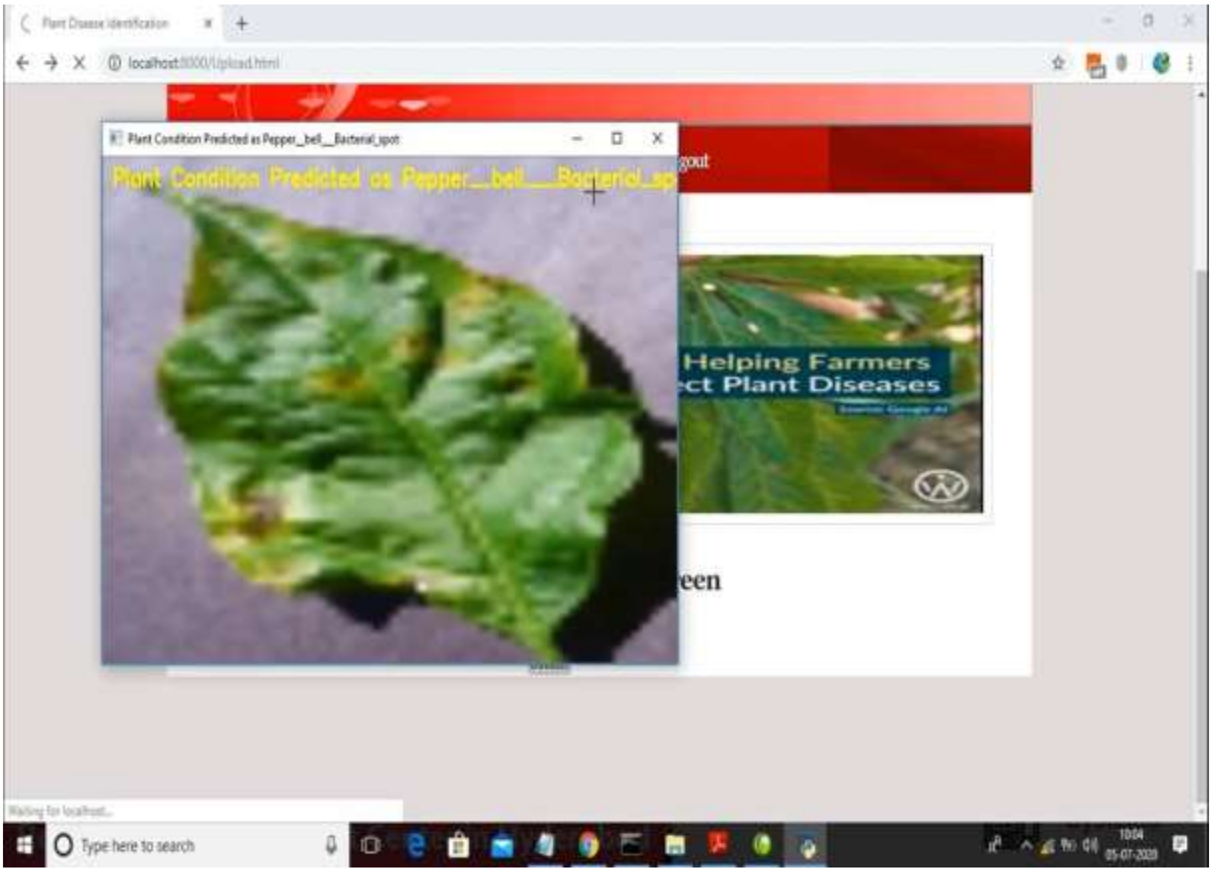
Screenshot 5.4: Uploading Data for Plant Leaf Disease Prediction Using Deep Learning

5.5 Uploading Image Data



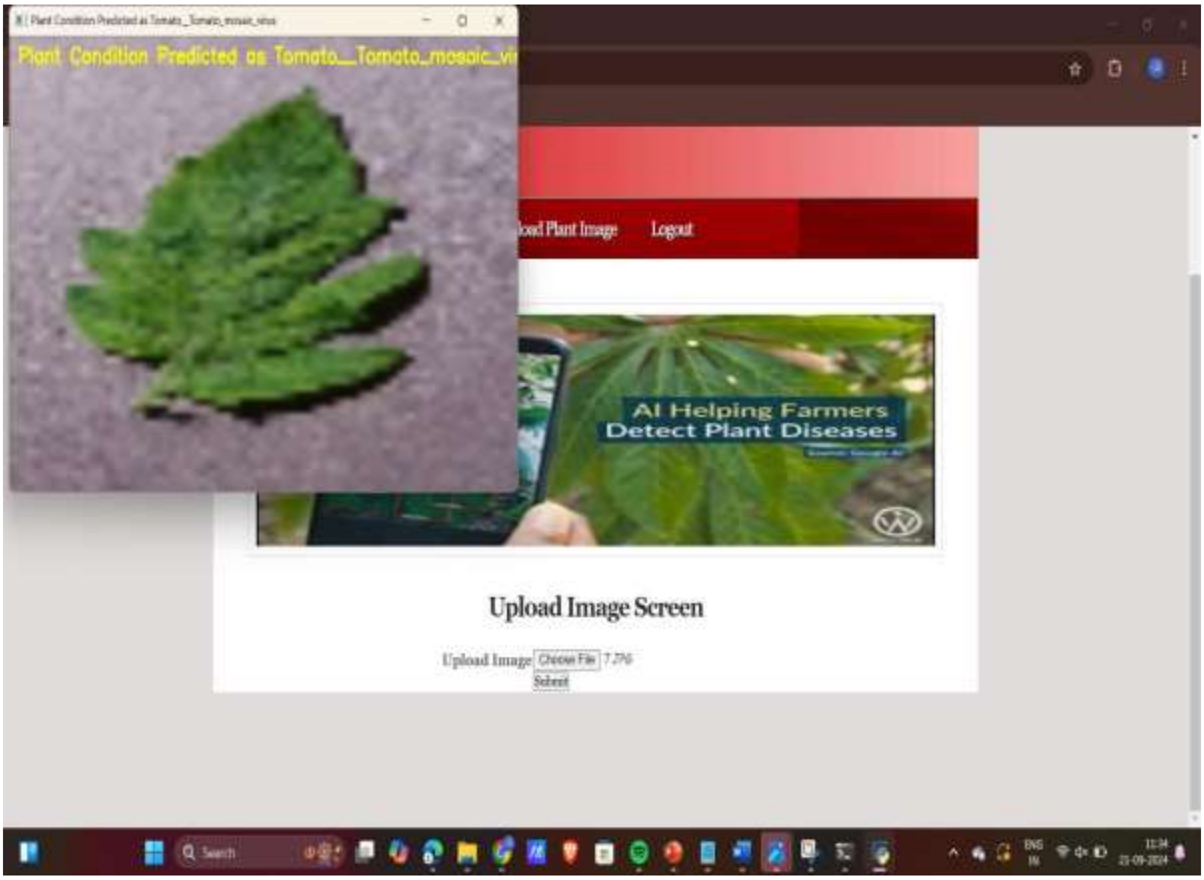
Screenshot 5.5: Uploading Image for Plant Leaf Disease Prediction Using Deep Learning

5.6 Displaying the Result for Pepper Bell Leaf



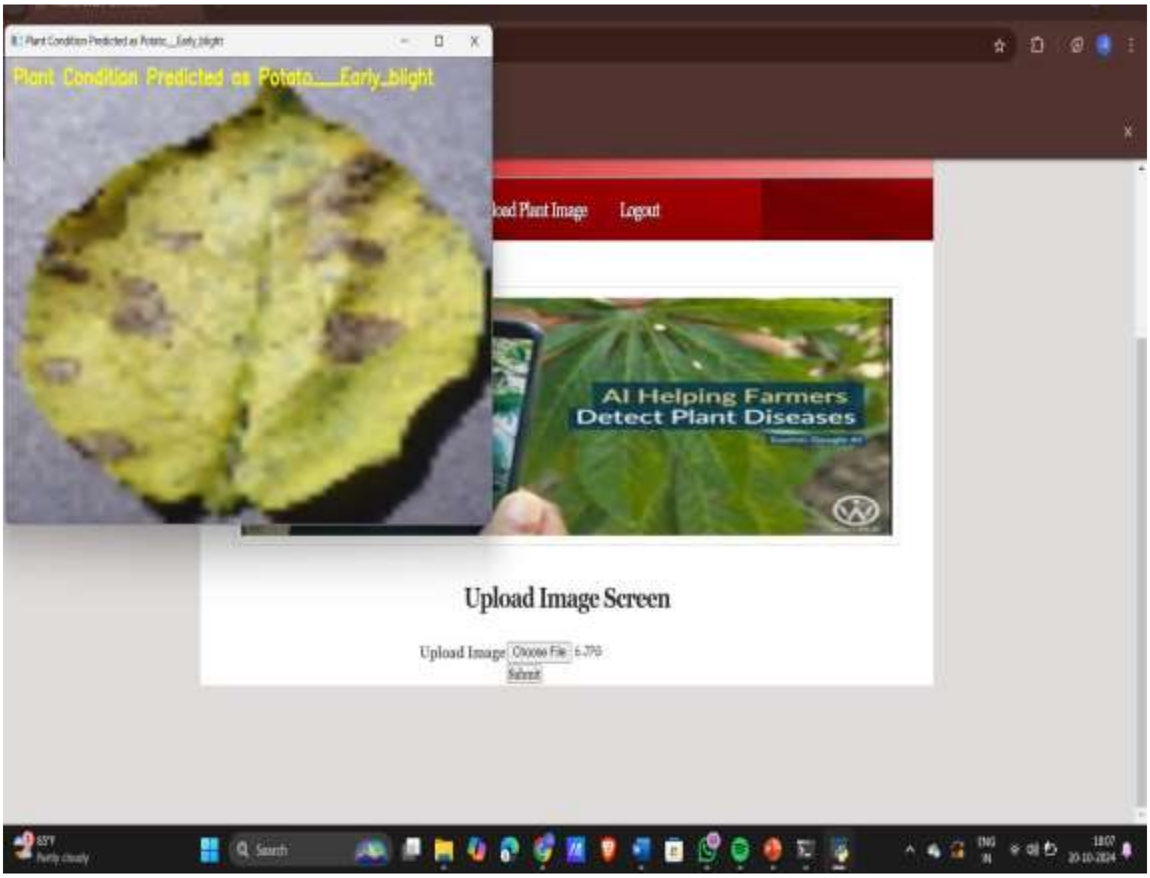
Screenshot 5.6 Displaying the Result for Plant Leaf Disease Prediction Using Deep Learning

5.7 Displaying the Result for Tomato Leaf



Screenshot 5.7 Displaying the Result for Plant Leaf Disease Prediction Using DeepLearning

5.8 Displaying the Result for Potato Leaf



Screenshot 5.8 Displaying the Result for Plant Leaf Disease Prediction Using Deep Learning

6. TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches andinternal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing theproblems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as the specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It testsa configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.3 TEST CASES

6.3.1 UPLOADING DATA

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads Plant leaf image data	Use it for classification	The user uploads the leaf image data.	Uploaded successfully
2	User uploads Plant leaf image data	Use it for classification	The user uploads the test leaf image data.	Uploaded successfully
3	User uploads Plant leaf image data	Use it for Classification	The user uploads the trained leaf image data.	Uploaded successfully

6.3.2 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Classification test 1	To check if the classifier performs its task	Plant leaf image isgiven	Gives the plant leafname
2	Classification test 2	To check if the classifier performs its task	Test data is given	Gives the plant leafdisease data
3	Classification test 3	To check if the classifier performs its task	Run CNN Algorit hm	Gives the predictedleaf result

7. CONCLUSION AND FUTURE SCOPE

7. CONCLUSION AND FUTURE SCOPE

7.1 PROJECT CONCLUSION

In summary, A major issue in the agricultural arena is the need for early, accurate crop disease prognoses and information on disease outbreaks. In order to help farmers make quick judgements on what disorder management measures to use, this research suggests an automated, low-value, and person-friendly quit-to-stop solution to this problem. This concept introduces a social collaborative platform to gradually increase accuracy, uses geocoded pictures for sickness density maps, applies deep Convolutional Neural Networks (CNNs) to disease classification, and imparts a professional interface for analytics, among other ways it improves upon present work. "Inception” is a high-quality deep CNN model that enables the cloud-based, user-facing mobile app to provide real-time illness classification. It is also possible to generate infection density maps using publicly accessible geolocation data within user-brought images stored in the cloud, in conjunction withexisting data on collective illness classifications. In conclusion, our experimental results show that the proposal is practically deployable in a number of ways: it can detect early symptoms ofdiseases, differentiate between illnesses within the same family, perform better with high fidelity real-life training records, become more accurate as the training dataset grows, and run well on a cloud-based entirely infrastructure that could handle a huge variety of disease classes.

7.2 FUTURE SCOPE

In order to strengthen the connection to the pollution, future research should expand the model to include more variables. We may supplement the picture collection with farmer inputs about soil, and publicly available environmental factors like temperature, humidity, and rainfall to improve our model's accuracy and enable disease predictions. We also want to decrease the frequency of expert engagement and broaden the spectrum of agricultural disorders that may be managed, excluding newly discovered diseases. More accurate class predictions with less human intervention will result from the ability to automatically normalise user-uploaded photos into the Training Database.

8. BIBLIOGRAPHY

8. BIBILOGRAPHY

8.1 REFERENCES

[1] L. Saxena and L. Armstrong, "A survey of image processing techniques for agriculture," in Proceedings of Asian Federation for Information Technology in Agriculture, 2014, pp. 401-413.

[2] E. L. Stewart and B. A. McDonald, "Measuring quantitative virulence in the wheat pathogen Zymoseptoria tritici using high-throughput automated image analysis," in Phytopathology 1049, 2014,
pp.985-992.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Image net classification with deep convolutional Neural networks," in Advances in Neural Information Processing Systems, 2012.

[4] TensorFlow. [Online]. Available: https://www.tensorflow.org/

[5] D. P. Hughes and and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," in CoRR abs/1511.08060, 2015.

[6] S. Raza, G. Prince, J. P. Clarkson and N. M. Rajpoot, "Automatic detection of diseased tomato plants using thermal and stereo visible light images," in PLOS ONE, 2015.

[7] D. L. Hernández-Rabadán, F. Ramos-Quintana and J. Guerrero Juk, "Integrating soms and a Bayesian classifier for segmenting diseased plants in uncontrolled environments," 2014, in the Scientific World Journal, 2014.

[8] S. Sankaran, A. Mishra, J. M. Maja and R. Ehsani, "Visible-near infrared spectroscopy for detection of huanglongbing in citrus orchards," in Computers and Electronics in. Agriculture 77, 2011, pp. 127-134.

8.2 WEBSITES

https://github.com/sindhuvurugonda568/Plant-leaf-Disease-Prediction.git