**Plant Disease Prediction using Deep Learning**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE AWARD OF DEGREE OF

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE**



Submitted by

Shakti Maddeshiya (2000290120141)

Saurabh Mishra (2000290120139)

Shivanshu Singh (2000290120152)

.

Supervised by

Prof. Raj Kumar

(Assistant Professor & Addl. HOD)

**Session 2023-24**

DEPARTMENT OF COMPUTER SCIENCE

**KIET GROUP OF INSTITUTIONS, GHAZIABAD**

**(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)**

**DECLARATION**

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name:-

Roll No.:-

Date:-

## CERTIFICATE

This is to certify that Project Report entitled “Plant Disease Prediction using Deep Learning” which is submitted by Shakti Maddeshiya, Shivanshu Singh and Saurabh Mishra in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**Date: Supervisor**

Prof. Raj Kumar

(Assistant Professor & Addl. HOD)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe a special debt of gratitude to Professor Rajkumar, Addl. Head of Department of Computer Science, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen the light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Srivastava, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not want to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date :

Signature: Signature:

Name: Shakti Maddeshiya Name: Shivanshu Singh

Roll No.: 2000290120141 Roll No.: 2000290120152

Signature:

Name: Saurabh Mishra

Roll No.: 2000290120139

**ABSTRACT**

The world's population depends heavily on agriculture, but agricultural production is still seriously threatened by plant diseases. Manually tracking plant diseases takes time and is prone to error. Early identification of plant diseases is possible with the use of artificial intelligence and computer vision. This may be beneficial to mitigate their negative consequences and overcome the limitations of ongoing human surveillance. This article presents an automated technique that uses an idea of CNN model to Identify and categorize diseases affecting plant leaves. Deep learning convolutional neural networks are like powerful tools for spotting plant diseases. We can solve this problem by cutting out only the important parts from the pictures (we call them regions of interest or ROIs) before giving them to CNN. This can make the model work much better. Regarding data augmentation methods, we discovered that the CNN architectures perform noticeably better when the rotation, shift, and zoom techniques are combined. Furthermore, integrating data augmentation methods with the offline training approach yields the most accurate outcome. Plant leaf disease detection could be transformed by DL. Deep learning (DL) can assist farmers in early and efficient plant disease identification and treatment, preserving crops and minimizing financial losses, by creating more resilient models that can function well on real-world imagery and early detection data.

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **Page No.** |
| DECLARATION……………………………………………………………………. | 2 |
| CERTIFICATE……………………………………………………………………… | 3 |
| ACKNOWLEDGEMENTS…………………………………………………………. | 4 |
| ABSTRACT………………………………………………………………………..... | 5 |
| LIST OF FIGURES…………………………………………………………………. | 8 |
| LIST OF TABLES…………………………………………………………………… | 9 |
| LIST OF ABBREVIATIONS………………………………………………………. | 10 |
|  |  |

CHAPTER 1. INTRODUCTION……………………………………………………......11-16

1.1 INTRODUCTION TO PROJECT…………………………………………………….11

1.2 PROJECT CATEGORY………………………………………………………………11

1.3 OBJECTIVES…………………………………………………………………………12

1.4 PROBLEM FORMULATION……………………………………...............................12

1.5 PROPOSED SYSTEM………………………………………………...........................13

1.6 UNIQUE FEATURES OF THE SYSTEM……………………………………………15

CHAPTER 2. REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION……. 17-23

2.1 FEASIBILITY.………………………………………………………………………...17

2.2 SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT…………..............19

2.3 SDLC MODEL TO BE USED………………………………………………...............22

CHAPTER 3. SYSTEM DESIGN………………………………………………….........24-28

3.1 DETAIL DESIGN………………………………………………………………..........24

3.2 SYSTEM DESIGN USING DFD LEVEL 0 AND LEVEL 1………………...............25

3.3 USE CASE DIAGRAM……………………………………………………………….26

3.4 DATABASE DESIGN………………………………………………………………...27

3.4.1 ER DIAGRAMS……………………………………………………………................27

3.5 SEQUENCE DIAGRM…………………………………………………………………28

CHAPTER 4. IMPLEMENTATION, TESTING, AND MAINTENANCE…………….29-31

4.1 INTRODUCTION TO LANGUAGES, TOOLS AND TECHNOLOGIES

USED FORIMPLEMENTATION………………………………………………………….30

4.2 TESTING TECHNIQUES AND TEST CASES USED………………………………31

CHAPTER 5. RESULTS AND DISCUSSIONS……………………………………….. 32-37

5.1 USER INTERFACE REPRESENTATION …………………………………………..25

5.1.1 BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM…………….25

5.2 SNAPSHOTS OF SYSTEM WITH BRIEF DETAIL OF EACH…………………….26

CHAPTER 6. CONCLUSION AND FUTURE SCOPE ……………………………............36

REFERENCES…………………………………………………………………………39-40

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| 3.1 | Detailed Design of the System | 25 |
| 3.2 | 0 Level DFD | 26 |
| 3.3 | 1 Level DFD | 26 |
| 3.4 | Use Case Diagram | 27 |
| 3.5 | ER Diagram | 28 |
| 3.5 | Sequence Diagram | 29 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table. No.** | **Description** | **Page No.** |
| 4.1 | Test Cases Used | 23 |

**LIST OF ABBREVIATIONS**

CNN Convolutional Neural Network

DL Deep Learning

AI Artificial Intelligence

**CHAPTER 1**

**INTRODUCTION**

* 1. **INTRODUCTION**

Our India is second largest populated country in the world and most people are farmers, but farmers are not so much educated, so they do not identify the different diseases of plants. The agriculture field is the backbone of Indian economy. Plant diseases are the biggest ongoing challenge for smallholder farmers. Our project was planned that to develop an application that simply take an image from user by clicking a photograph using mobile camera or just upload the image from mobile gallery with user’s permission and predict the best possible (matched from dataset) diseases that could possibly affecting that plant. So, user can get the proper information that what is the problem with the plant and what could be the reasons and could search for the proper cure that help to overcome the disease. It’ll be achieved by implementation of deep learning (in image processing) and some application building tools (for application). Inside the system the input image is examined by system that detects unhealthy regions of plant leaves, classification of plant leaf diseases using texture features to analyze the leaf infection (disease). We’ll make this service available on a mobile app which can run on low configuration devices. Farmers experience great difficulties in switching from one disease control policy to another disease control policy. The naked eye observation of experts is the traditional approach, this method can be time consuming, expensive and inaccurate, so our purpose is to develop an easy-to-use mobile application which use deep learning to predict the disease on time with ease that can help farmers very much.

* 1. **PROJECT CATEGORY**

Our project falls under the category of agricultural technology or agrotech. It focuses on automating the detection of plant diseases using deep learning techniques, specifically targeting the analysis of plant leaves. This project aims to assist farmers and agriculturists in identifying diseases early on to mitigate their impact on crop yield and ensure food security.

* 1. **OBJECTIVES**

Our main objectives are:

* To build a plant disease prediction system using Deep Learning.
* To detect unhealthy regions of plant leaves.
* Classification of plant leaf diseases using features texture.
* To analyze the plant diseases/infections.
* To give remedy information to user.
* To make this service available on mobile application that can run on low configuration devices.
  1. **PROBLEM FORMULATION**

Agriculture is an integral pillar of the Indian economy, serving as the livelihood for nearly half of the country's workforce. India holds the distinction of being the largest producer of several key agricultural commodities globally, including pulses, rice, wheat, spices, and spice products. However, amidst the backdrop of this agricultural prowess, we are confronted with a pressing challenge—the escalating levels of pollution that permeate our environment. This environmental degradation poses a significant threat to our soil health and overall crop productivity. For smallholder farmers whose livelihoods hinge solely on agricultural income, the impact of plant diseases looms large. These diseases pose a constant threat to crop health and yield, exacerbating the economic vulnerability of farmers.

The crux of our project lies in addressing this critical issue through the development of a plant disease recognition and classification system employing deep learning methodologies. In the agricultural landscape, the economic well-being of farmers hinges on the quality and yield of their produce, making timely disease identification paramount. Plant diseases, which manifest in various parts of the plant, particularly in leaves, can significantly impede plant growth and development. Manual diagnosis of these diseases through leaf photographs is a time-consuming endeavor fraught with challenges.

To overcome these limitations, we propose the adoption of an automated disease detection technique leveraging deep learning algorithms. By harnessing the power of computational methods, we aim to streamline and expedite the process of disease detection and classification using leaf images. Through this endeavor, we endeavor to empower farmers with a tool that not only detects plant diseases at an early stage but also aids in mitigating their adverse effects on crop health and yield. In doing so, we aspire to contribute to the resilience and sustainability of India's agricultural sector, safeguarding the livelihoods of millions of farmers across the country.

* 1. **PROPOSED SYSTEM**

The proposed system aims to develop a robust plant disease recognition and classification system using state-of-the-art deep learning techniques. This system will revolutionize the way plant diseases are identified and managed in agricultural settings. Key components of the proposed system include:

**1. Deep Learning Models:** The core of the system will consist of deep learning models trained on large datasets of plant images annotated with disease labels. These models will leverage convolutional neural networks (CNNs) and other advanced architectures to learn intricate patterns and features indicative of various plant diseases.

**2. Image Processing:** An image processing pipeline will be implemented to preprocess raw plant images before feeding them into the deep learning models. This pipeline will include techniques such as image enhancement, noise reduction, and feature extraction to optimize the input data for disease classification.

**3. Disease Recognition and Classification:** The system will be capable of accurately recognizing and classifying plant diseases based on visual symptoms observed in leaf images. By analyzing the morphology and texture of plant leaves, the system will identify patterns indicative of specific diseases, enabling rapid and precise diagnosis.

**4. User Interface:** A user-friendly interface will be developed to facilitate interaction with the system. Users, including farmers and agricultural experts, will be able to upload plant images, view disease detection results, and access relevant information and recommendations for disease management.

**5. Scalability and Accessibility:** The proposed system will be designed to scale effortlessly to accommodate a growing volume of data and users. It will be accessible via web and mobile platforms, ensuring widespread adoption and usability across diverse agricultural settings.

**6. Integration with Agricultural Ecosystem:** To maximize its impact, the system will seamlessly integrate with existing agricultural infrastructure and practices. This integration will enable real-time monitoring of crop health, automated alert systems for disease outbreaks, and data-driven insights for informed decision-making in farming operations.

Overall, the proposed system represents a paradigm shift in plant disease management, offering a cost-effective, efficient, and scalable solution to safeguard crop productivity and farmer livelihoods. By harnessing the power of deep learning and image processing technologies, the system holds the potential to revolutionize agriculture and contribute to global food security efforts.

**1.6 UNIQUE FEATURES OF THE SYSTEM**

**1.6.1 Customization and Training:**

Our Plant Disease Recognition System distinguishes itself with its customization and training capabilities, empowering developers to tailor deep learning models to specific agricultural needs. Key features include:

**Developer-centric Model Training:** Developers have the capability to train models using diverse datasets, ensuring adaptability to various agricultural contexts without relying on end-users for training.

**Annotation and Labeling:** Tools are provided for developers to annotate images with disease categories, facilitating model training with labeled data and improving model accuracy.

**Transfer Learning:** Leveraging pre-trained models, developers can accelerate training and enhance performance for specific disease recognition tasks, saving time and computational resources in model development.

**Accuracy:** Developers can optimize model performance by adjusting hyperparameters, fine-tuning the system to meet specific requirements, and achieving higher accuracy in disease classification.

These features empower developers to create customized solutions for effective plant disease management, addressing the diverse challenges encountered in agricultural operations.

**1.6.2 Multi-class Object Detection:**

Our system excels in multi-class object detection, enabling simultaneous identification and classification of multiple plant diseases. This feature offers several benefits:

**Comprehensive Disease Identification:** The system detects various diseases affecting plants, providing developers with a holistic overview of crop health status.

**Enhanced Decision Support:** Developers receive valuable insights for disease management strategies, allowing them to prioritize interventions based on the specific diseases detected, thus optimizing resource allocation.

**Versatile Application:** The system's versatility extends to different crops and agricultural settings, ensuring its relevance across diverse farming environments, from small-scale farms to large plantations.

**Robust Performance:** Utilizing advanced deep learning algorithms, the system delivers accurate and reliable disease detection, even under challenging environmental conditions, ensuring consistent performance in real-world applications.

This feature enhances the system's utility, contributing to improved agricultural productivity, crop yield, and sustainability.

**1.6.3 User-friendly UI:**

The system features a user-friendly interface developed using Tkinter, ensuring ease of interaction for developers. The intuitive design facilitates seamless model training and management, allowing developers to efficiently customize and deploy the system for effective disease diagnosis and management.

**CHAPTER 02**

**REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION**

**2.1 FEASIBILITY STUDY**

Conducting a feasibility study for a Plant Disease Recognition System using Deep Learning involves evaluating its technical, economic, and operational aspects. Following are the various components:

**2.1.1 TECHNICAL FEASIBILITY:**

**2.1.1.1 System Requirements:**

The system requires basic hardware infrastructure, such as a smartphone with a camera and internet connectivity. Google Colab will be utilized for model training, leveraging its cloud-based resources.

**2.1.1.2 Training Data:**

The system will utilize the Plant Village dataset sourced from Kaggle for training the deep learning model. This dataset includes a comprehensive collection of images representing various plant diseases and health conditions.

**2.1.1.3 Performance Metrics:**

The system's performance will be evaluated based on its ability to accurately recognize and classify plant diseases from images captured by the smartphone camera. Metrics such as accuracy, precision, recall, and F1 score will be considered, aiming for optimal performance.

**2.1.2 ECONOMIC FEASIBILITY:**

**2.1.2.1 Cost Analysis:**

The initial costs associated with implementing the system are minimal. As the system utilizes existing hardware resources and cloud-based services like Google Colab, there are no significant upfront costs.

**2.1.2.2 Return on Investment (ROI):**

While there might not be direct monetary gains from the system, the potential benefits lie in improved crop management, increased yield, and reduced crop losses due to timely disease detection. These indirect benefits contribute to the overall return on investment for farmers and agricultural stakeholders.

**2.1.2.3 Payback Period:**

Given the low initial investment, the payback period for the Plant Disease Recognition System is virtually immediate, as the benefits of improved crop health and management begin to accrue upon implementation.

**2.1.3 OPERATIONAL FEASIBILITY**

**2.1.3.1 User Acceptance:**

Farmers and agricultural practitioners are likely to embrace this system due to its simplicity and effectiveness in identifying plant diseases. The convenience of using a smartphone for disease diagnosis enhances its acceptance among users.

**2.1.3.2 Training:**

Training requirements for end-users are minimal, primarily involving familiarization with the smartphone application interface for capturing images of plant leaves and accessing diagnostic results. Basic instructions and tutorials can facilitate the training process, ensuring ease of use for users with varying technical backgrounds.

This feasibility study establishes the viability and practicality of implementing a Plant Disease Recognition System using deep learning techniques, highlighting its potential to revolutionize crop management practices and contribute to agricultural sustainability.

**2.2 SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT**

**2.2.1 DATA REQUIREMENT**

**2.2.1.1 Data Sources**

To train the Plant Disease Recognition System specifically for potato diseases like early blight and late blight, we primarily utilize the following data source:

**Plant Village Dataset:** The Plant Village dataset, sourced from Kaggle, contains a comprehensive collection of images focusing on various plant diseases, including early blight and late blight affecting potatoes.

**2.2.1.2 Data Set**

The image dataset extracted from the Plant Village dataset consists of a significant number of images specifically depicting early blight and late blight diseases affecting potato plants. This curated dataset serves as the primary training and validation data for the deep learning model.

**2.2.2 FUNCTIONAL REQUIREMENT**

**2.2.2.1 Object Detection**

The Plant Disease Recognition System utilizes deep learning algorithms to efficiently identify and categorize potato diseases, with a specific focus on early blight and late blight.

**2.2.2.2 Disease Classification**

Upon detection, the system accurately classifies the identified diseases, providing users with detailed information about early blight and late blight affecting potato plants, including symptoms, severity, and recommended management strategies.

**2.2.3 PERFORMANCE REQUIREMENT**

**2.2.3.1 Speed and Responsiveness**

The system must demonstrate fast processing speed, ensuring that disease detection and classification occur within a reasonable timeframe, typically less than 10 seconds per image.

**2.2.3.2 Accuracy**

The system should achieve a high level of accuracy in disease detection and classification, with a minimum confidence level of 80% to ensure reliable results specifically for early blight and late blight affecting potato plants.

**2.2.4 MAINTAINABILITY REQUIREMENT**

**2.2.4.1 Code Maintainability**

The system's codebase, including all relevant documents such as the Software Requirement Specification (SRS), Test Plan, and Project Report, is meticulously maintained. Version control using Git and hosting on platforms like GitHub ensures efficient code management.

**2.2.4.2 Updates and Upgrades**

Regular updates and upgrades are essential for keeping the system relevant and functional. These updates should be implemented promptly to minimize system downtime and ensure continuous improvement in performance and functionality, particularly in addressing new potato diseases or improving detection accuracy.

**2.2.5 SECURITY REQUIREMENT**

**2.2.5.1 Access Control**

Access to the Plant Disease Recognition System is controlled by administrators, who have the authority to manage user permissions and determine access levels for different system functionalities. This ensures that only authorized personnel, such as agricultural practitioners and researchers, can access and utilize the system's features, focusing on early blight and late blight detection for potato plants. Additional security measures may include user authentication mechanisms such as login credentials or biometric authentication to prevent unauthorized access.

**2.3 SDLC MODEL TO BE USED**

For the development of the Plant Disease Recognition System, the Agile methodology has been chosen as the preferred Software Development Life Cycle (SDLC) model. Agile methodology is well-suited for this project due to its inherent characteristics that align with the nature of machine learning, image processing, and AI projects. Below are the key reasons for selecting Agile:

Flexibility and Adaptation: Agile methodology allows for flexibility in project planning and execution. It enables teams to adapt to changing requirements, evolving technologies, and emerging insights from testing and user feedback. This is particularly important for projects involving complex algorithms like machine learning and image processing, where iterative development is necessary to refine the system's accuracy and performance.

Ongoing Testing: Agile promotes continuous testing throughout the development lifecycle. By integrating testing into each iteration or sprint, teams can identify and address issues early, ensuring the quality and reliability of the system. This iterative testing approach is crucial for validating the effectiveness of the disease recognition algorithms and improving their accuracy over time.

Incremental Delivery: Agile emphasizes delivering working software in small, incremental increments. This allows stakeholders to see tangible progress and provide feedback at regular intervals, facilitating early validation of requirements and alignment with stakeholders' expectations. For a project like plant disease recognition, incremental delivery enables users to start benefiting from the system's capabilities sooner while also allowing for ongoing refinement and enhancement.

Customer Collaboration: Agile encourages close collaboration between development teams and stakeholders, including end-users and domain experts. By involving stakeholders throughout the development process, Agile ensures that the final product meets their needs and addresses their pain points effectively. In the context of plant disease recognition, involving agricultural experts and farmers in the development process can lead to a more user-centric and impactful solution.

While Agile is known for its iterative and incremental approach, it is important to note that it does not follow a strictly sequential development process like the waterfall model. Instead, Agile emphasizes adaptability, responsiveness to change, and continuous improvement, making it an ideal choice for dynamic and innovative projects like the Plant Disease Recognition System.

**CHAPTER 03**

**SYSTEM DESIGN**

**3.1 DETAIL DESIGN**

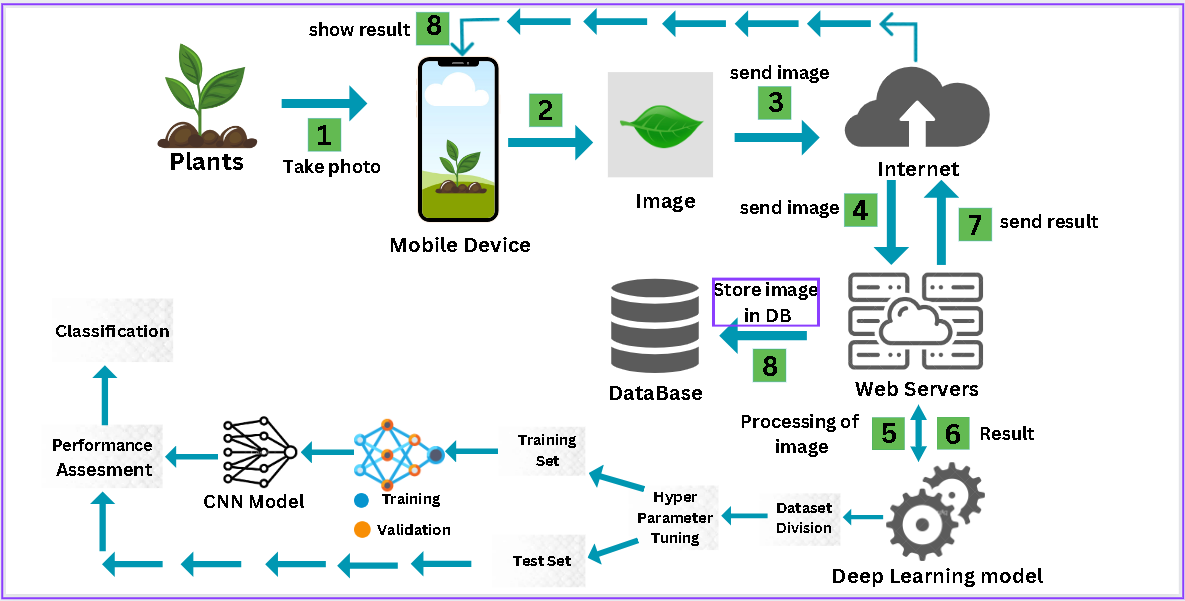


Fig. 3.1. Detailed Design of the System (Workflow)

**3.2 SYSTEM DESIGN USING DFD LEVEL 0 AND LEVEL 1**

**3.2.1 DFD Level 0**

A diagram of a plant processing process

Description automatically generated

**Fig. 3.2.1 Level 0 DFD**

**3.2.2 DFD Level 1**

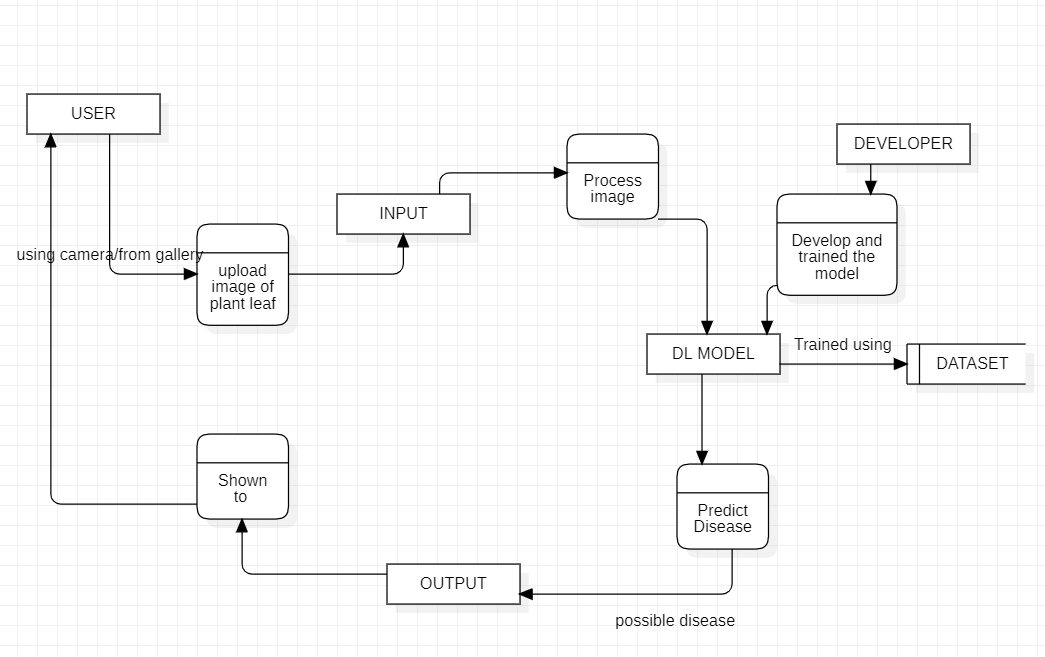


Fig. 3.2.2 Level 1 DFD

**3.3 USE CASE DIAGRAM**

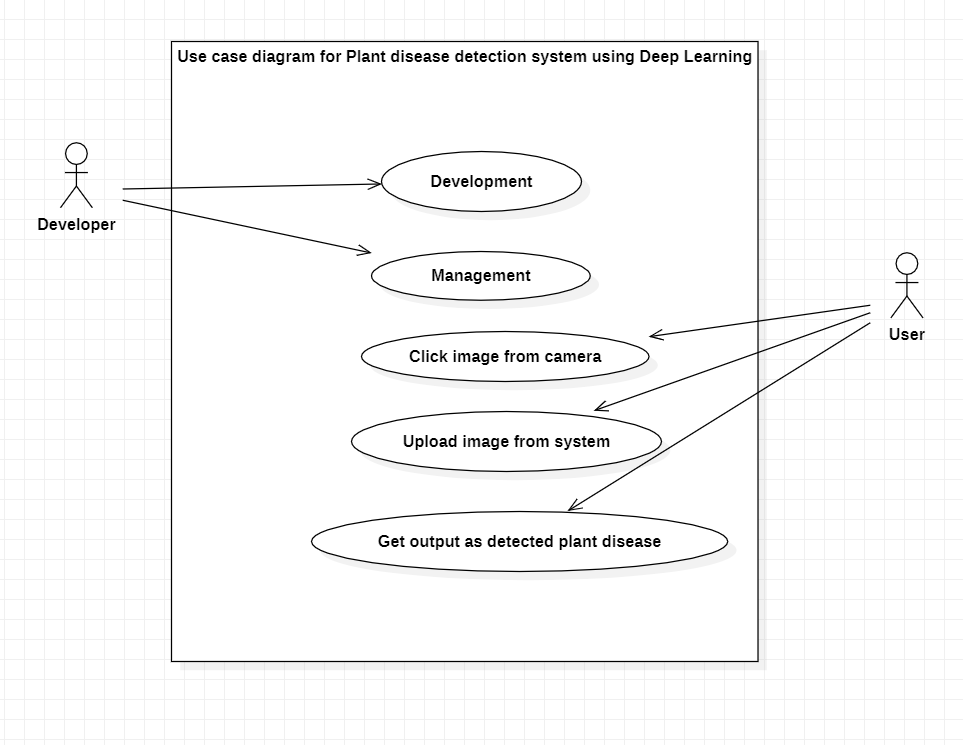


Fig. 3.3 Use Case Diagram

**3.4 DATABASE DESIGN**

**3.4.1 ER DIAGRAM**

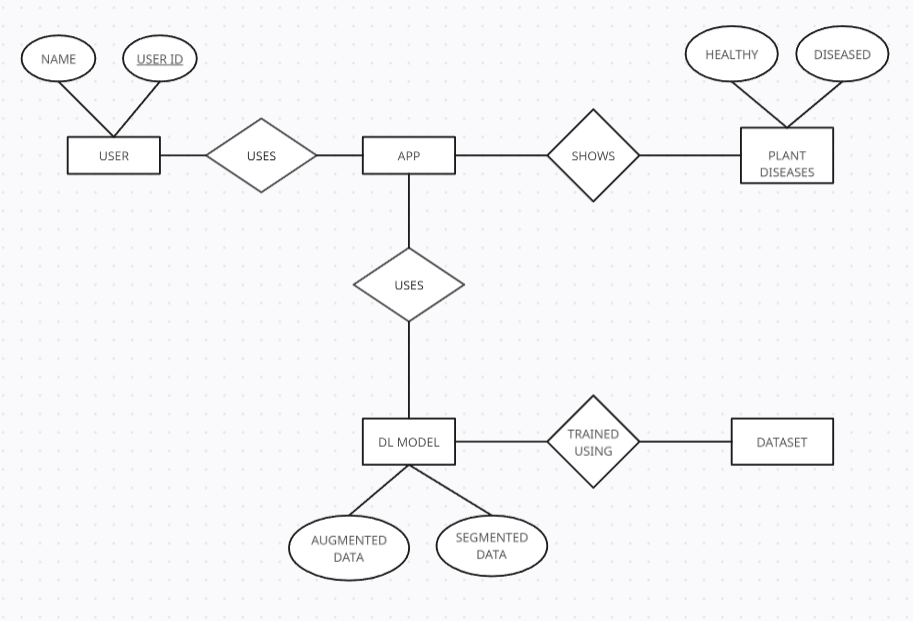
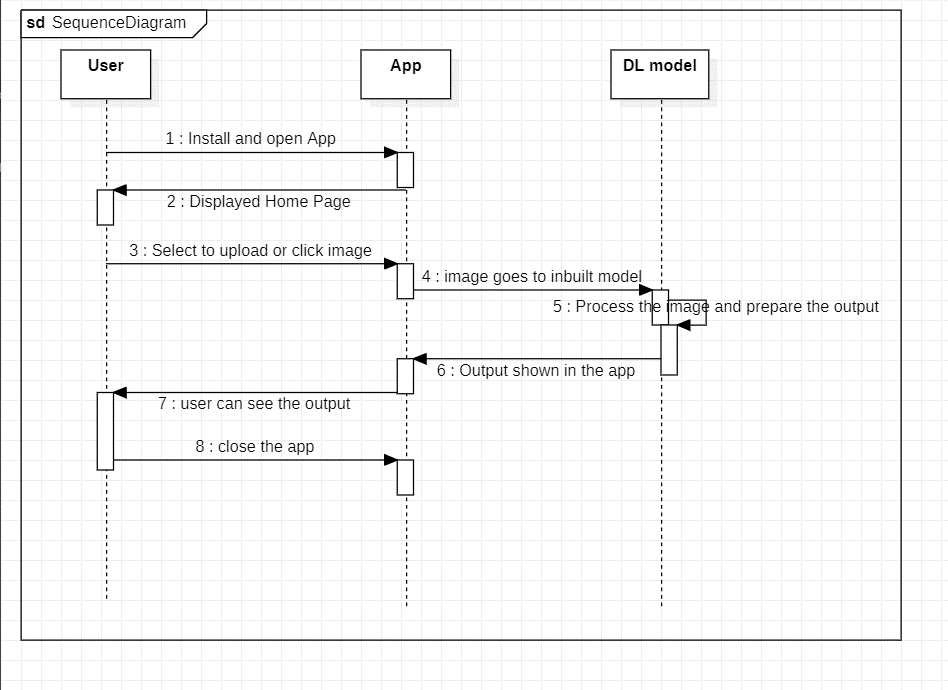


Fig. 3.4 ER Diagram

**3.5 SEQUENCE DIAGRAM**



**Fig. 3.5 SEQUENCE Diagram**

**CHAPTER 04**

**IMPLEMENTATION, TESTING, AND MAINTENANCE**

**4.1 INTRODUCTION TO LANGUAGES, TOOLS AND TECHNOLOGIES USED FOR IMPLEMENTATION**

**4.1.1 Language used:**

For the implementation of our plant disease prediction project, we primarily utilized Python as the programming language. Python's versatility, extensive library support, and simplicity made it an ideal choice for developing machine learning models and handling image data processing tasks efficiently. Moreover, Python's ecosystem boasts a rich selection of deep learning frameworks and libraries, such as TensorFlow and Keras, which were instrumental in building and training our predictive model.

**4.1.2 Toolkit:**

The primary toolkit employed in our project was TensorFlow, an open-source deep learning framework developed by Google. TensorFlow provided a comprehensive suite of tools and libraries for building, training, and deploying machine learning models, particularly neural networks. Leveraging TensorFlow's high-level API, Keras, we were able to streamline the model development process and experiment with various architectures easily. Additionally, we utilized other Python libraries such as NumPy, Matplotlib, and PIL (Python Imaging Library) for data manipulation, visualization, and image processing tasks, respectively.

**Roboflow:** It is a Computer Vision developer framework used for better data collection, pre-processing, and model training techniques.

**YOLOv5:** It is a novel convolutional neural network (CNN) that detects objects in real-time with great accuracy. This approach uses a single neural network to process the entire picture, then separates it into parts and predicts bounding boxes and probabilities for each component.

**4.1.3 User Interface:**

In our project, the user interface was developed using the Flutter framework. Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop platforms from a single codebase. By utilizing Flutter, we were able to create a visually appealing and intuitive user interface that seamlessly integrated with our plant disease prediction model. The user interface provided features such as image upload, prediction display, and informative visualizations to enhance the user experience and facilitate interaction with the predictive model.

**4.1.4 Coding Environment:**

The coding environment for our project was set up using Google Colab, a cloud-based Jupyter notebook environment provided by Google. Google Colab offered a convenient platform for writing, executing, and collaborating on Python code, particularly for machine learning and data analysis tasks. With built-in support for popular libraries such as TensorFlow, Colab enabled us to harness the computational power of Google's GPUs and TPUs for training our deep learning models efficiently. Additionally, Colab provided seamless integration with Google Drive, facilitating data access and storage for our project.

Overall, the combination of Python as the programming language, TensorFlow as the deep learning framework, Flutter for the user interface, and Google Colab as the coding environment provided a robust foundation for the successful implementation of our plant disease prediction project.

**4.2 TESTING TECHNIQUES AND TEST CASES USED**

The testing performed is white box testing.

**4.2.1 TEST LEVELS**

For the "PLANT DISEASE DETECTION SYSTEM USING DEEP LEARNING " system, the test levels can be organized to ensure thorough validation at different stages of development, from individual components to the full, integrated system. Here's how the test levels could be structured:

**1.Unit Testing:**

**Objective: -**

* Verify the correctness of individual components and functions within the system.

**Scope: - -**

* Test individual functions responsible for image preprocessing.
* Validate the correctness of disease classification algorithms.

**2.Integration Testing:**

**Objective:**

* Validate the interaction and collaboration between different modules and components.

**Scope:**

* Test the integration of image acquisition and preprocessing modules.
* Validate the integration of the deep learning model with the overall system.

**3.Component Testing:**

**Objective:**

* Test the behavior of specific components or subsystems in isolation.

**Scope:**

* Test the image acquisition module to ensure it correctly captures and reads images.
* Verify the correctness of the preprocessing module.

**4.System Testing:**

**Objective:**

* Assess the functionality of the entire system.

**Scope:**

* Test the end-to-end process from image acquisition to disease detection.
* Validate system behavior under normal and abnormal conditions.

**5.Performance Testing:**

**Objective:**

* Evaluate the responsiveness, scalability, and resource usage of the system.

**Scope:**

* Test the system's ability to handle a high volume of image processing requests.
* Measure the response time under different load conditions.

**6.Regression Testing:**

**Objective:**

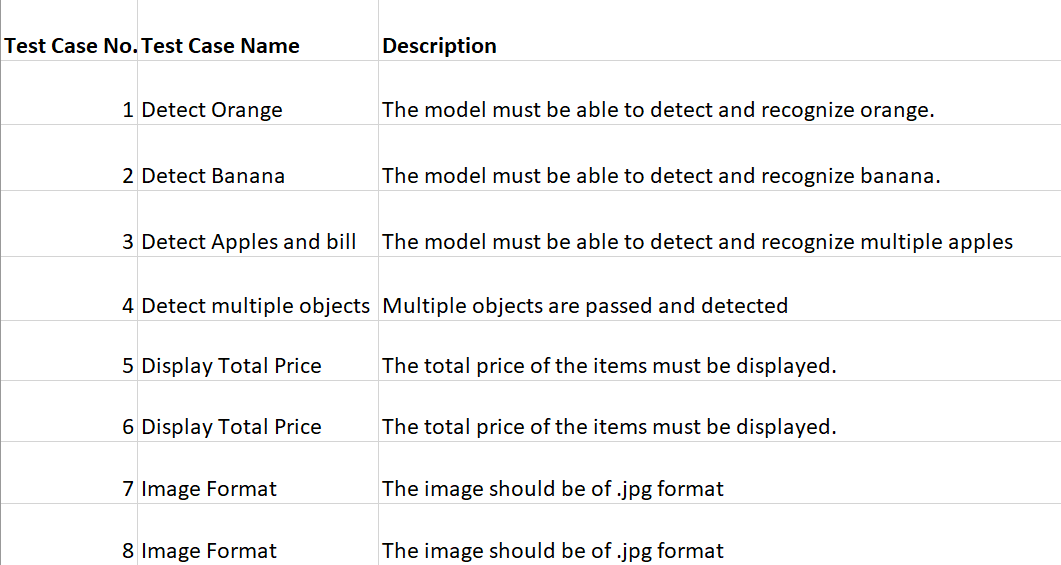
* Ensure that new changes do not negatively impact existing functionalities.

**Scope:**

* Perform regression tests after updates to the deep learning model or changes to the user interface.
* Confirm that new features or bug fixes do not introduce unintended side effects.

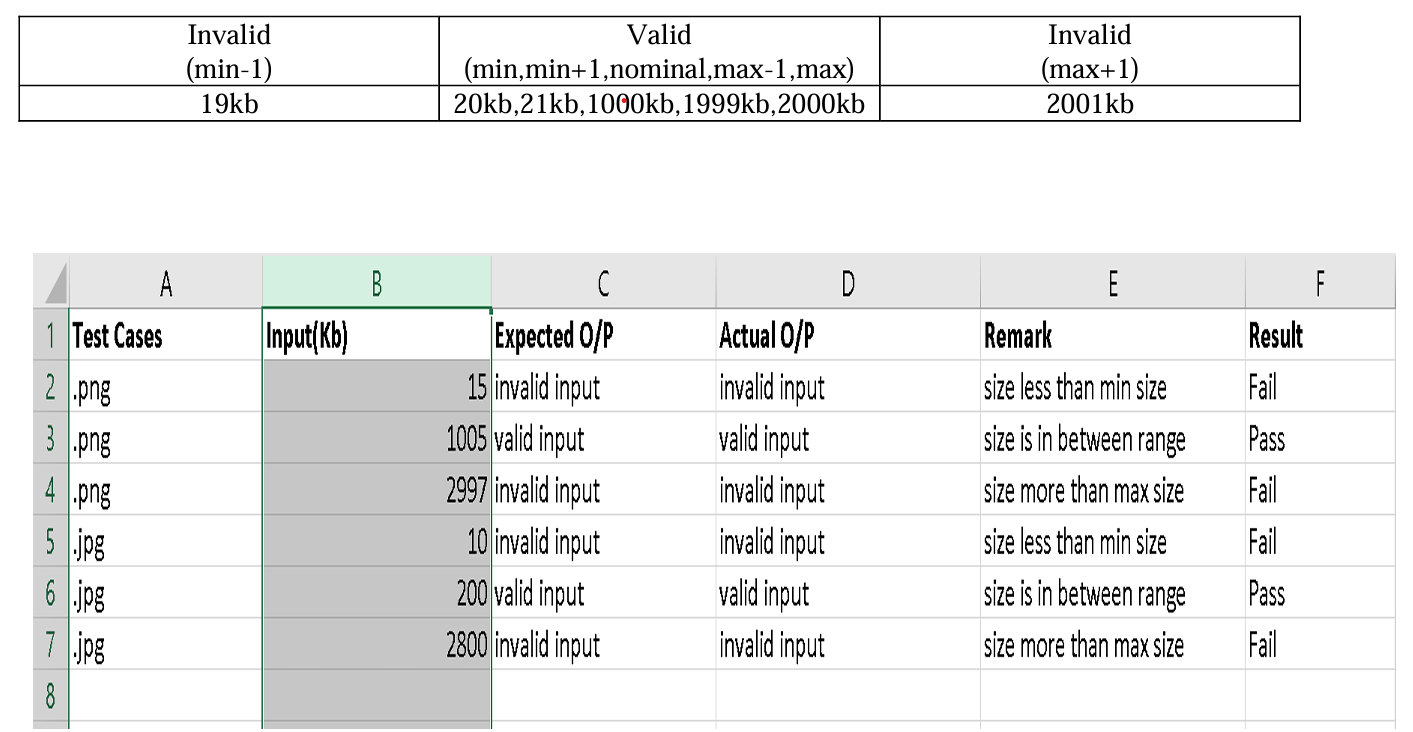
**4.2.2 TEST CASES USED:**

**Table No. 4.1 – Test cases used**



**Boundary Value analysis: Interface Capability**

**File size must be between 20kb to 2000kb**

****

**CHAPTER 05**

**RESULTS AND DISCUSSIONS**

**5.1 USER INTERFACE REPRESENTATION**

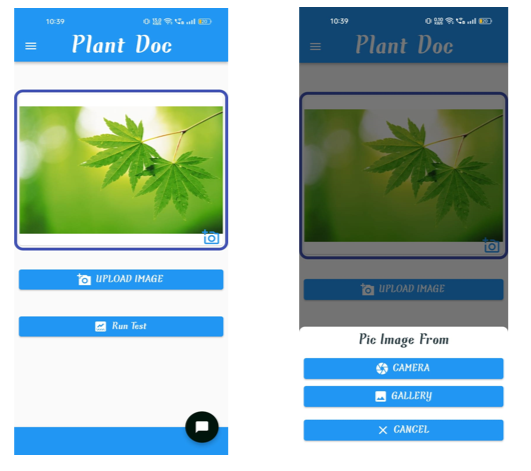


Fig. 5.6. Plant Doc(GUI)

Introducing our innovative Plant Disease Detection Application, a powerful tool fueled by Machine Learning CNN models. By simply uploading images of your plants, our advanced algorithms swiftly analyze foliage health, accurately pinpointing potential diseases.

Utilizing Convolutional Neural Networks (CNN), our application employs state-of-the-art deep learning techniques to meticulously examine every detail in plant images, distinguishing between healthy and diseased specimens with remarkable precision.

Say goodbye to manual inspections and guesswork. Our solution provides farmers and gardeners with instant insights, empowering proactive disease management and targeted treatment strategies.

Experience seamless detection, precise diagnosis, and swift responses, ensuring the robustness of your crops and greenery. Embrace innovation with our Plant Disease Detection Application – safeguarding yields, preserving landscapes, and transforming agricultural practices.

**5.1.1 BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM**

**5.1.1.1 Object Detection Module (YOLOv5 Integration):**

It utilizes YOLOv5 for real-time object detection and classification. It processes images to identify products which helps a crucial role in generating the bill. We have used yolov5s which has 7.2M parameters.

**5.1.1.2 GUI Module (Flutter):**

The graphical user interface (GUI) is implemented through Flutter. It help in uploading the image from the mobile or by capturing the real time image and the send it to the ml model for processing.

**5.1.1.3 Leaf disease prediction Module:**

A set of code designed to accurately predict diseases affecting plant leaves. Using advanced Machine Learning techniques, this module analyzes leaf images and predicts the presence of diseases based on learned patterns and characteristics. By inputting leaf images into the code, users can quickly obtain predictions regarding the specific disease affecting the plant, enabling prompt intervention and management strategies. With this code module, users can efficiently monitor and address plant health issues, ultimately optimizing crop yields and ensuring agricultural sustainability.

**5.1.1.4 Training and Fine-tuning Module:**

Train MobileNet on a labeled plant leaf dataset. Fine-tune the model for optimized disease prediction. Deploy the trained model for real-time inference on new leaf images.

MobileNet is readily available as part of the TensorFlow Keras applications module. You can use it for various tasks including image classification, feature extraction, and transfer learning.

**5.2 SNAPSHOTS OF SYSTEM WITH BRIEF DETAIL OF EACH**

**5.2.1 Code for the model training**

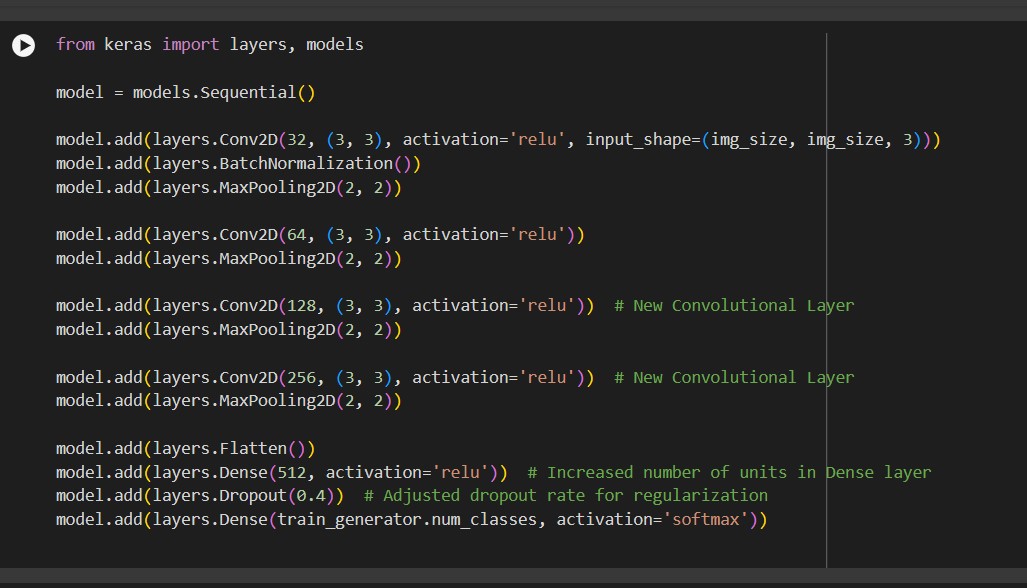


Fig. 5.7. Code for the model training

**5.2.2 Show** **Pridiciton Result Function**

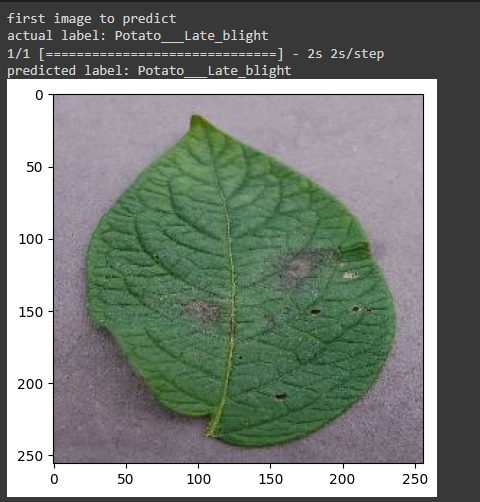
****

Fig. 5.8. Pridiciton Result Function

This function plays a pivotal role in the plant disease recognition system using deep learning by producing a comprehensive prediction report based on the identified plant diseases. Initially, it establishes a list containing the classes of plant diseases present in the dataset, such as healthy, early blight, and late blight. Subsequently, the function iterates through the detected items, verifying if they correspond to any of the specified disease classes. When a match is identified, the function updates a dictionary with the quantity of each detected disease. Additionally, it incorporates a list storing the prices or probabilities associated with each disease class, facilitating the assessment of their significance in the prediction. By structuring the information into a Pandas DataFrame, the function ensures systematic organization and clarity in presenting the prediction results. Ultimately, it generates a comprehensive report, detailing the frequency or probability of occurrence of each detected plant disease.

**5.2.3 Running the detection on test data**

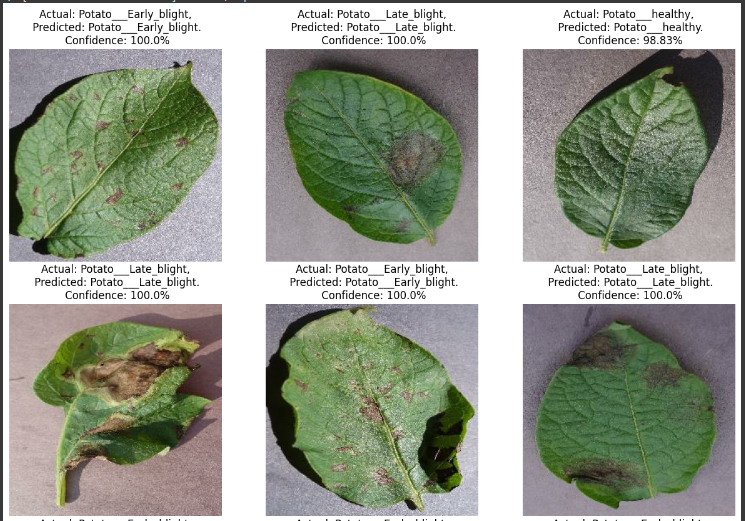
****

Fig. 5.9. Visualizing the detection result from our model

The purpose of the provided function is to construct a convolutional neural network (CNN) model for the recognition of plant diseases. The model architecture consists of multiple layers including convolutional layers with increasing filter sizes, max-pooling layers for spatial dimension reduction, and dense layers for classification. The input images are first resized and rescaled to ensure consistency in dimension and pixel values. Subsequently, the convolutional layers extract essential features from the images, while the max-pooling layers downsample the feature maps. The flattened feature maps are then passed through dense layers, employing activation functions to output probabilities for each class of plant disease. The model is built and trained on a dataset of plant images, allowing it to learn to differentiate between healthy plants and those affected by various diseases, aiding in effective disease diagnosis and management.

**5.2.4 Final Output**

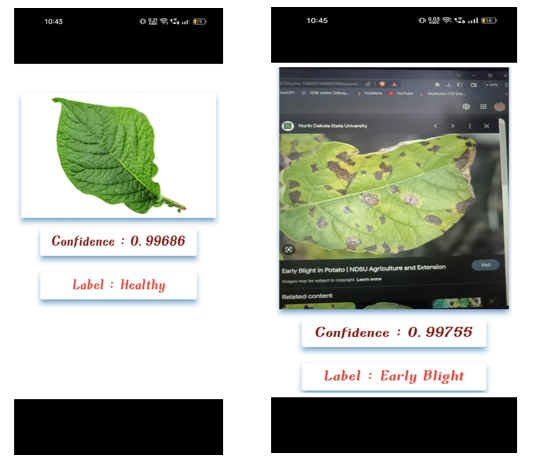


Fig. 5.10. Final Output Screen

In fig. 10, users can upload images of potato leaves for analysis. The application utilizes sophisticated algorithms to accurately detect and diagnose common potato diseases such as late blight and early blight.

Upon uploading a potato leaf image, the application swiftly processes the data, identifying any signs of late blight or early blight. Users receive detailed results indicating the health status of the potato leaf, along with any detected diseases and their severity.

With our innovative technology, farmers and gardeners can proactively manage potato plant health, enabling timely intervention and targeted treatment strategies to mitigate the impact of blight diseases and safeguard crop yields.

**CHAPTER 06**

**CONCLUSION AND FUTURE SCOPE**

In conclusion, our research efforts in the field of AI- based plant disease recognition and classification for precision agriculture have shown encouraging outcomes, which represents a major advancement in filling the gaps in the current plant disease detection approaches that are mechanized or computerized. Even with a variety of approaches, there are still no complete solutions or commercial applications in the field of automated plant species detection, especially when it comes to leaf image-based methods.

This article presents a novel method for automatically classifying and identifying plant diseases from leaf images by utilizing convolutional neural networks (CNN), a deep learning technology. The created model performed admirably in identifying whether leaves were there and differentiating between leaves that were healthy and leaves that were affected by one of the 2 diseases that were found; each ailment may be visually inspected. The whole process has been thoroughly documented, starting with the painstaking gathering of training and validation images and continuing through the complex phases of image preprocessing, augmentation, and deep neural network training. Extensive experiments and testing have been carried out to assess our model's performance and provide us an understanding of its strengths and weaknesses.

Notably, when the present model is used, it produces a binary output that categorizes leaves as either healthy or unhealthy. Still, there's a lot of room for improvement and growth. By utilizing more advanced neural network architectures and incorporating a wider range of diverse and large- scale datasets, the Disease Detection System could potentially achieve higher accuracy and identify a wider range of plant diseases.

**REFERENCES**

[1] Chowdhury, M.E.; Khandakar, A.; Ahmed, S.; Al-Khuzaei, F.; Hamdalla, J.; Haque, F.; Mamun, B.I.R.; Ahmed, A.S.; Nasser, A.E. Design, construction and testing of iot based automated indoor vertical hydroponics farming test-bed in Qatar. Sensors 2020, 20,5637

[2] R.N. Strange, P.R. Scott, Plant disease: a threat to global food security, Annu. Rev. Phytopathol. 43 (2005) (2005) 83–116.

[3] W. B. Jones, S. V Thomson, and others, “Source of inoculum, yield, and quality of tomato as affected by Leveillula taurica.,” Plant Dis., vol. 71, no. 3, pp. 266– 268, 1987.

[4] KC, K., Yin, Z., Wu, M., Wu, Z., 2019. Depthwise separable convolution architectures for plant disease classification. Comput. Electron. Agric. 165, 104948.

[5] McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys.1943, 5, 115–133.

[6] Abdulridha, J., Ampatzidis, Y., Ehsani, R., & de Castro, A. (2018). Computers and Electronics in Agriculture, 155, 203–211.

[7] Sagar, Abhinav & Jacob, Dheeba. (2020). On Using Transfer Learning For Plant Disease Detection. 10.13140/RG.2.2.12224.15360/1.

[8] Tyagi, A.C. Towards a second green revolution. Irrig. Drain. 2016, 65, 388– 389.

[9] A. M. Abdu, M. M. M. Mokji, and U. U. U. Sheikh, “Machine learning for plant disease detection: an investigative comparison between support vector machine and deep learning,” IAES Int. J. Artif. Intell., vol. 9, no. 4, pp. 670–683, Dec. 2020, doi: 10.11591.

[10] Coakley SM, Scherm H, Chakraborty S. Climate change and plant disease management. Annu Rev Phytopathol. 1999 Sep; 37:399-426. doi: 10.1146/annurev.phyto.37.1.399. PMID: 11701829.

[11] Hossain, S.; Lee, D.-J. Autonomous-Driving Vehicle Learning Environments using Unity Real-time Engine and End-to-End CNN Approach. J. Korea Robot. Soc. 2019, 14, 122–130.

[12] Koci´c, J.; Joviˇci´c, N.; Drndarevi´c, V. An End-to-End Deep Neural Network for Autonomous Driving Designedfor Embedded Automotive Platforms. Sensors 2019, 19, 2064.

[13] Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.;Dean, J. A guide to deep learning in healthcare. Nat. Med. 2019, 25, 24.

[14] Miotto, R.; Wang, F.; Wang, S.; Jiang, X.; Dudley, J.T. Deep learning for healthcare: Review, opportunities and challenges. Brief. Bioinform. 2017, 19, 1236–1246.

[15] Ravì, D.;Wong, C.; Deligianni, F.; Berthelot, M.; Andreu-Perez, J.; Lo, B.; Yang, G.-Z. Deep learning for health informatics. IEEE J. Biomed. Health Inform. 2016, 21, 4–21.

[16] Goodfellow, I.J.; Bulatov, Y.; Ibarz, J.; Arnoud, S.; Shet, V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv 2013, arXiv:1312.6082.

[17] Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep structured output learning for unconstrained text recognition. arXiv 2014, arXiv:1412.5903.

[18] Yousfi, S.; Berrani, S.-A.; Garcia, C. Deep learning and recurrent connectionist-based approaches for Arabic text recognition in videos. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 1026–1030.

[19] DeVries, P.M.; Viégas, F.;Wattenberg, M.; Meade, B.J. Deep learning of aftershock patterns following large earthquakes. Nature 2018, 560, 632.

[20] Mousavi, S.M.; Zhu, W.; Sheng, Y.; Beroza, G.C. CRED: A deep residual network of convolutional and recurrent units for earthquake signal detection. Sci. Rep. 2019, 9, 10267. opportunities and challenges. Brief. Bioinform. 2017, 19, 1236–124.