PROJECT REPORT
ON

BOTANIC IMAGE ANALYZER

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

STUDENT NAME	EXAM NO:
Kshitij Tripathi	**B191094267**
Saurabh Prakash	**B191094322**
Shivansh Shukla	**B191094334**
Aniket Udhane	**B191094357**

DEPARTMENT OF COMPUTER ENGINEERING

DR. D.Y.PATIL INSTITUTE OF ENGINEERING, MANAGEMENT & RESEARCH

AKURDI, PUNE 411044

SAVITRIBAI PHULE PUNE UNIVERSITY

2023 -2024

CERTIFICATE

This is to certify that the project report entitles

"BOTANIC IMAGE ANALYZER"

Submitted by

STUDENT NAME	EXAM NO
Kshitij Tripathi	**B191094267**
Saurabh Prakash	**B191094322**
Shivansh Shukla	**B191094334**
Aniket Udhane	**B191094357**

are bonafide student of this institute and the work has been carried out by him/her under the supervision of **Mrs. Maheshwari Chittampalli** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Mrs. Maheshwari Chittampalli
Guide
Department of Computer Engineering

Mrs. P. P. Shevatekar
Head
Department of Computer Engineering

Dr. Anupama V. Patil
Principal,
Dr. D.Y. Patil Institute of Engineering, Management & Research Akurdi Pune – 41

External Examinar Signature:
Place : Pune
Date :

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the preliminary project report on 'BOTANIC IMAGE ANALYZER'.

We would like to take this opportunity to thank our internal guide Mrs. Maheshwari Chittampalli for giving us all the help and guidance we needed. We are really greatful to them for their kind support. Their valuable suggestions were very helpful.

We are also grateful to Mrs. P. P. Shevatekar, Head of Computer Engineering Department, Dr. D. Y. Patil Institute of Engineering, Management & Research for her indispensable support, suggestions. Our special thanks to Prof. Amol Dhakne for providing various resources such as laboratory with all needed software platforms, continuous guidance, for Our Project.

Finally, we would like to thank our friends who have directly or indirectly helped us in our project work.

Kshitij Tripathi B191094267
Saurabh Prakash B191094322
Shivansh Shukla B191094334
Aniket Udhane B191094357
(BE Computer Engineering)

ABSTRACT

Plants have played a vital role in human civilization for centuries, offering sustenance, medicine, and ecological benefits. Accurate identification of plant species is essential for understanding their properties, characteristics, and potential applications, information about plants and diseases. Traditional methods of plant identification often involve labor-intensive processes such as manual observation of botanical characteristics, which are prone to human error and subjectivity. Additionally, not all individuals possess the expertise required to accurately identify plant species. These limitations highlight the need for automated and efficient approaches to plant identification.

This project is all about Botanic Image Analyzer that leverages image processing, deep learning, and machine learning techniques to automatically identify and classify plant species from images, providing detailed information about these plants and facilitating disease recognition. It explores the development, advantages, and potential applications of the leaf image analysis .The analysis highlights its impact on plant identification, education, agriculture, and conservation, providing insights into its implications for plant sciences.

In conclusion, the botanic image analysis project strives to create a user-friendly web application. To address challenges of Leaf image identification and plant disease detection by providing a comprehensive solution.

TABLE OF CONTENTS

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
CNN	Convolutional Neural Network
ReLU	Rectified Linear Unit
NN	Neural Network
IMG	Image
MPL	Max Pooling Layer
CV	Computer Vision

LIST OF FIGURES

1. INTRODUCTION

1.1 OVERVIEW

Plants have played a vital role in human civilization for centuries, offering sustenance, medicine, and ecological benefits. Accurate identification of plant species is essential for understanding their properties, characteristics, and potential applications. Traditional plant identification methods often involve labor-intensive processes such as manual observation of botanical characteristics, which are prone to human error and subjectivity. Additionally, not all individuals possess the expertise required to accurately identify plant species. These limitations highlight the need for automated and efficient approaches to plant identification.

The Botanic Image Analyzer emerges as a groundbreaking solution, automating plant species identification and offering valuable information about plants and its diseases.

Botanic image analyzer is a plant leaf identification and disease detection working application. In this, we use images of leaves of different plants, and using image processing to process those images into data chunks, further, after getting the desired input for the model we try to get the properties of the plant leaf such as species, its condition (diseased or not). We have a deep learning model trained on the training datasets collected from various sources. At last we have a database containing all the relevant information about the plant leaves available and after putting the leaf data into model it extracts the related information from the database using backend technology such as Node.js or Mongo db and to interconnect the frontend and database also. Thus all the results are presented to user.

1.2 MOTIVATION

Creating a Botanic Image Analyzer can contribute to the preservation of plant biodiversity. By automating the process of identifying and cataloging plant species, you can aid in conservation efforts, especially for endangered or rare plants.

1.3 Problem Statement:

"To Make a Web Application that helps to identify plant leaf and detect disease in the leaf based on the leaf image uploaded by the user as an input to the app."

1.4 Objectives:

1. Automated Plant Identification: Develop a web application that utilizes computer vision and machine learning to automatically identify and classify plant species based on user-submitted images.

2. User-Friendly Interface: Create an intuitive and user-friendly web interface that allows users, including researchers, nature enthusiasts, and students, to easily upload images for plant identification without requiring specialized botanical knowledge.

3. Accurate Species Recognition: Aim for a high level of accuracy in plant species identification to support research, education, and conservation efforts effectively.

4. Educational Tool: Provide a platform that can serve as an educational resource, helping users learn about different plant species and their characteristics.

5. Contribute to Conservation: Enable users to contribute to biodiversity conservation by cataloging and monitoring plant species, particularly rare or endangered ones.

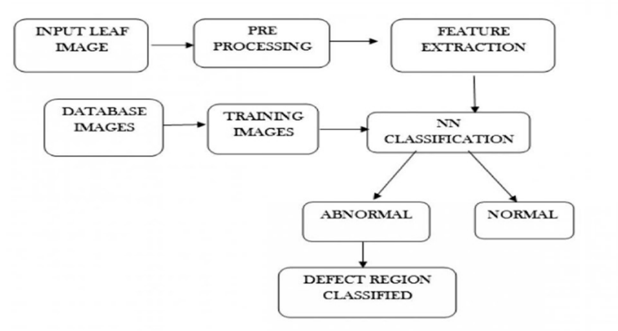
1.5 PROJECT SCOPE AND LIMITATIONS

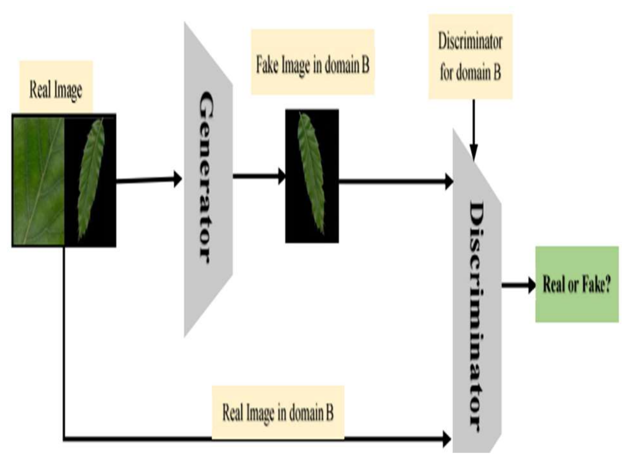
The successful completion of this project will have a significant positive impact on the agricultural sector, providing farmers and agricultural professionals with a valuable tool for early detection and treatment of plant diseases as it can accurately identify plant species and diseases from leaf images. This can lead to improved crop yields, reduced losses due to disease outbreaks, enhanced agricultural productivity, Enabling conservation efforts by identifying rare and endangered species and Enhancing crop health management and early disease detection.

Limitations

- The accuracy of the Botanic Image Analyzer may still result in false positives (incorrectly identifying a species) or false negatives (failing to identify a species). Continuous validation and improvements are required to reduce error rates.
- If the training data is biased towards certain plant species or regions, the Botanic Image Analyzer may exhibit bias in its identification results, potentially favouring common species over rarer ones.
- The ability of the analyzer to detect diseases in plants is limited by the extent to which the diseases manifest in the images.
- May only identify and classify plant species it has been trained on. It may not be able to generalize to entirely new, untrained species, which can limit its usefulness in diverse ecosystems.
- The performance heavily relies on the availability of high-quality image datasets. A limited or biased dataset may lead to issues with model accuracy and generalization.

2. LITERATURE SURVEY

Sr. No	Paper Title	Journal Name	Authors & Publication Date	Methodology
1	Plant Leaf Disease Classification and Detection System Using Machine Learning	ICCPET 2020	G. Geetha, S.Samundeswari, G.Saranya ,K.Meenakshi and M. Nithya 17/12/2020	Look at the raw data available to us and study it in-order to identify suitable attributes for the prediction of our selected label.  KNN algorithm we first start with the loading of the data and then initializing K to the selected number of neighbours, then for each example in the data we find the distance between the query example and the given current example from the given data. Then the data collected is sorted orderly from smallest to largest according to the distances, later we pick the first k entries from the sorted collection and take the labels of the selected k entries
2	A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning	IEEE ACCESS	Emmanual Moupojou Appolinair Tagne 29/03/2023	To identify diseases on raw field images we used Convolutional neural network. As expected, when the training and test sets are the same, the noise backgrounds and the multiplicity of leaves on the raw images reduce the models validation accuracies. These accuracies are further reduced when the models are trained on Plant Village and tested on Plant Doc or Field Plant. CNNs are made up of layers of neurons, each of which learns to detect different features of an image. For example, the first layer of a CNN might learn to detect edges, the second layer might learn to detect patterns of edges, and the third layer might learn to detect specific objects, such as plant. The results show that the existing models are not sufficiently accurate for plant disease detection and classification of images collected directly from the field, although the classification task results for Field Plant are better than those for Plant Doc.

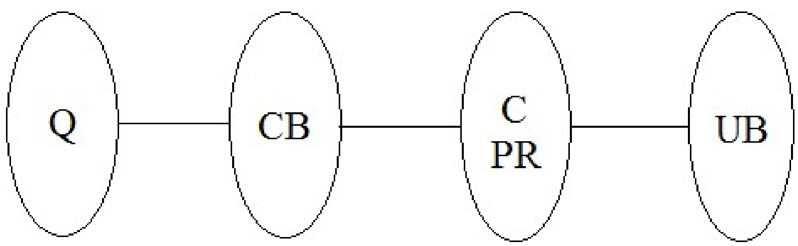
3	Detection of Apple Plant Diseases Using Leaf Images Through Convolutional NeuralNetwork	IEEE ACCESS	Arfat Ahmad Khan 28/12/2023	Authors proposed a general framework that CNN for heavy training to make high-Disease prone apple plant. The convolutional layer is responsible to perform convolution a filter (kernel) on an input image. The convolutional layer produces feature maps by finding the local conjunction that appears in the previous layers. CNNs work by applying a series of filters to an image. Each filter is a small matrix of weights that is applied to a small region of the image. The output of the filter is a new pixel value, which is calculated by multiplying the filter weights with the pixel values in the input region. The filters are applied to the image in a sliding window fashion, meaning that the filter is applied to each region of the image, one by one. The output of the filters is then fed into the next layer of neurons, which learns to detect more complex features.
4	A Deep Learning-Based Recognition Technique for Plant Leaf Classification	IEEE ACCESS	Kewen Xia 30/11/2021	A deep Convolutional Neural Network (CNN) can extract higher-level features progressively from the input images given to it by the multiple layers used in a model. The utilization of a conditional Generative Adversarial Network to tackle the problem of a lack of sufficient training data or uneven class balance that could be found within datasets in performing Deep Learning tasks. This serves to augment leaf image datasets, which have not been large enough, as this field still lacks a large number of datasets for adequate training of Deep Neural Net-works for better generalization 

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 ASSUMPTIONS AND DEPENDENCIES

1. User must have the knowledge of web based application.
2. User must have Internet Facility/ Wifi Connection.
3. User must have all required software to run the application.

3.1.1 MATHEMATICAL MODELING



Where,
Q = read the dataset
CB = preprocess
C= apply deep learning algorithm

PR= Preprocess request evaluation

UB = predict outcome

Failures:

1. Huge database can lead to more time consumption to get the information.
2. Hardware failure.
3. Software failure.

Success:

- Search the required information from available in Datasets.
- User gets result very fast according to their needs.

3.2 FUNCTIONAL REQUIRMENTS

1. Image Capture and Real-Time Processing: Capture and process video streams from multiple cameras in real-time.
2. User Interface: Provide a user-friendly web-based interface for accessing Images from the user
3. Integration: Integrate with external systems and devices, such as cloud-based storage and access control system

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 USER INTERFACES

The requirements section of hardware includes minimum of 1TB hard disk and 8 GB RAM with clock speed 2.5 GHz

3.3.2 HARDWARE INTERFACES

As this is an online application for Image Analysis we are not enabling or installing any hardware components for user interface.
It’s not an embedded system

- Processor – Intel Core i3
- Speed - 2.5 Ghz and Above
- Camera
- RAM - 4 GB (min)
- Hard Disk - 1 GB

3.3.3 SOFTWARE INTERFACES

This is the software configuration in which the project was shaped. The programming languageused, tools used are described here.

- Operating System : Windows
- Front-End : Streamlit
- Back-End :Python
- Tool : Jupyter Notebook or Google Colab
- Database : .json , MY SQL, Mongo DB

3.3.4 COMMUNICATION INTERFACES

- User can access the web application from remote location.
- Standard internet connection is required.
- TCP/UDP connection will be required

3.4 NON-FUNCTIONAL REQUIREMENTS

1. PERFORMANCE REQUIREMENTS

- High Speed:

  System should process requested task in parallel for various action to give quick response. Thensystem must wait for process completion.

- Accuracy:

  System should correctly execute process, display the result accurately. System output should be in user required format.

2. SAFETY REQUIREMENTS

The data safety must be ensured by arranging for a secure and reliable transmission media. The source and destination information must be entered correctly to avoid any misuse or malfunctioning.

3. SOFTWARE QUALITY ASSURANCE

- Availability [related to Reliability]
- Modifiability [includes portability, reusability, scalability]
- Performance
- Access-ability
- Usability[includes self-adaptability and user adaptability]

3.5 SYSTEM REQUIREMENTS

3.5.1 DATABASE REQUIREMENTS

MySQL : MySQL is an open-source relational database management system (RDBMS). It was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

Mongo DB: MongoDB is a popular open-source, NoSQL database management system designed for handling unstructured or semi-structured data. It falls into the category of document-oriented databases, which means it stores data in a flexible, JSON-like format called BSON (Binary JSON). MongoDB is often used in applications where data is not well-suited for traditional relational databases.

3.5.2 SOFTWARE REQUIREMENTS

Operating system	:	Windows 7 and above.
Coding Language	:	Python
IDE	:	Jupyter Notebook, Google Colab, VScode

3.5.3 HARDWARE REQUIREMENTS

System	:	Intel I3 Processor and above.
Hard Disk	:	200 GB.
Ram	:	4 GB.

3.6 ANALYSIS MODELS : SDLC MODEL TO BE APPLIED

We are going to use Agile methodology for this project. It's centered around adaptive planning, self-organization and short-delivery time. It's flexible, fast and aims for continuous improvement in quality using tools like Scrum and extreme programming(xp).

In this we are going to use scrum and extreme programming.

Scrum – is hands-on system consisting of simple interlocking steps and components.

xp – use with scrum and centered around frequent releases and short development cycles.

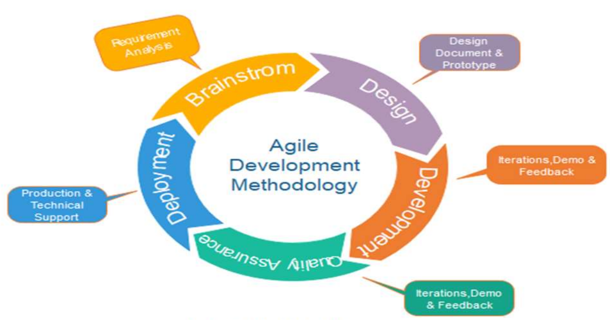


Fig. Agile Model

3.7 SYSTEM IMPLEMENTION PLAN

1. Requirement gathering and analysis

- Image Dataset: Gather a large dataset of leaf images. This dataset should include images of healthy leaves and leaves affected by various diseases or conditions. High-quality, labeled images are essential for training and testing models.
- Metadata: Record metadata about each image, such as the plant species and the specific disease or condition (if found).
- Machine Learning and Computer Vision Models: Choose the appropriate machine learning or deep learning models for image classification and object detection. Convolutional Neural Networks (CNNs) are commonly used for image-related tasks.

2. System Design:

System Design is the core concept behind the design of any distributed systems. System Design is defined as a process of creating an architecture for different components, interfaces, and modules of the system and providing corresponding data helpful in implementing such elements in systems.

3. Implementation:

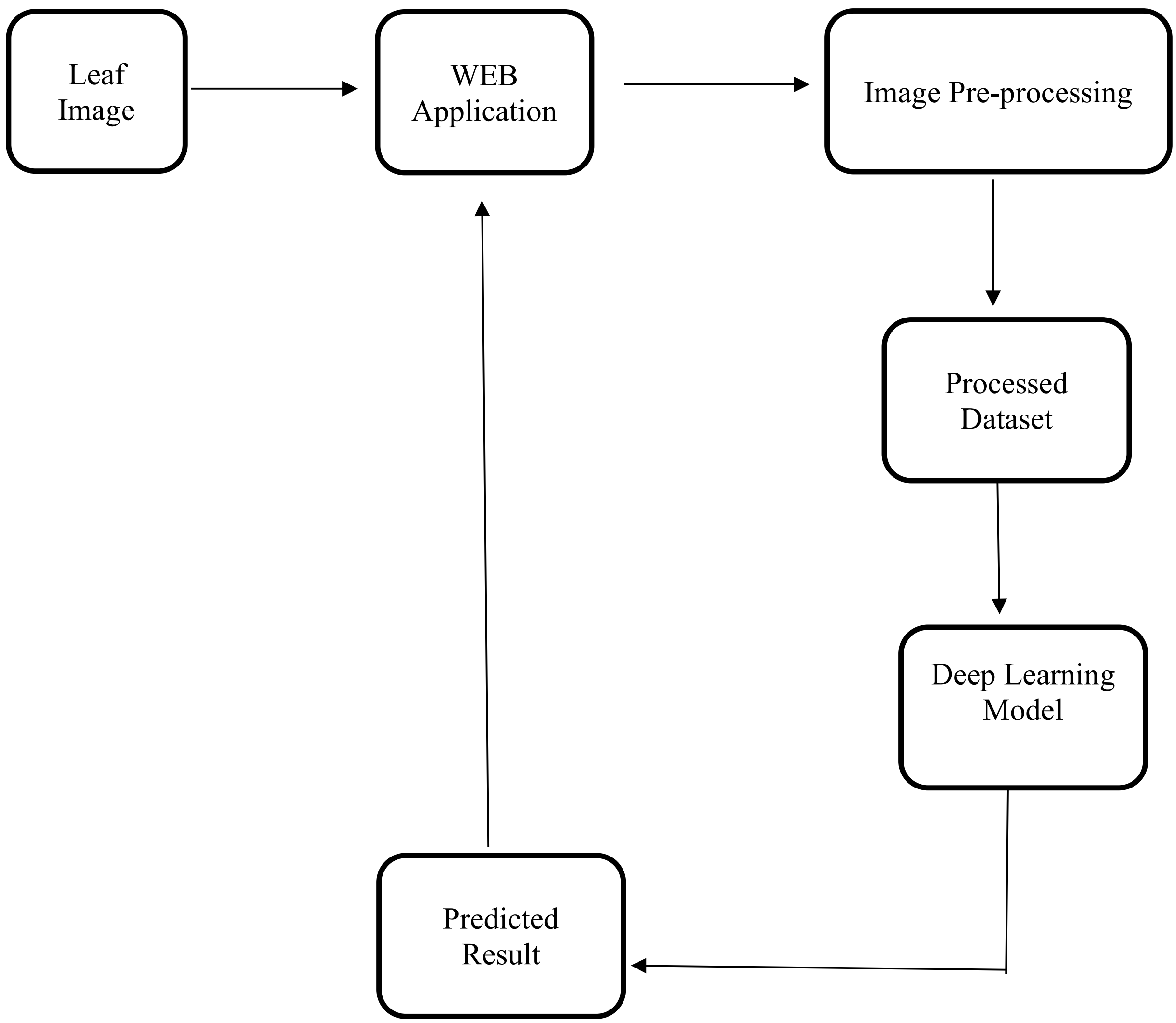
In implementation phase of our project we have implemented various module, model required for successfully getting expected outcome at the different module levels.

- Initiation: In this phase we identify the problem and gather all the resources(datasets) related to this project, we form a project for the problem, we identify the team and then we try to develop the project.
- Planning: This phase contains all the parts of project form scope to budget and beyond.
- Execution: In this phase we work on the plan and make the model as planned.
- Testing/Monitoring: During execution phase, after completion and after deployment we test the model as it fulfills our requirements or not and monitoring is done to keep the project development on right track.
- Deployment: The last phase of implementation after successful development to deploy the project on real world.

4. Maintenance: regular updates on the data and increased model prediction accuracy.

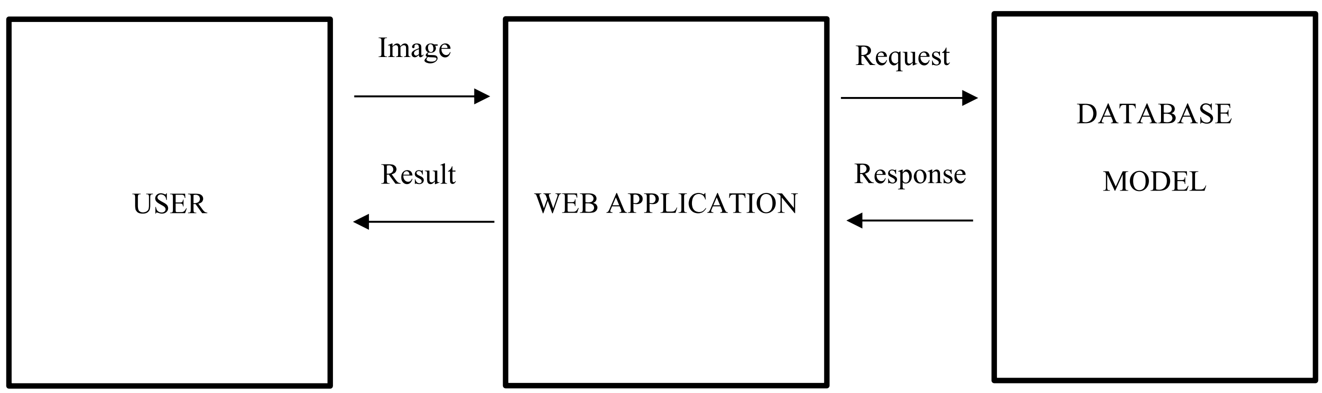
4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

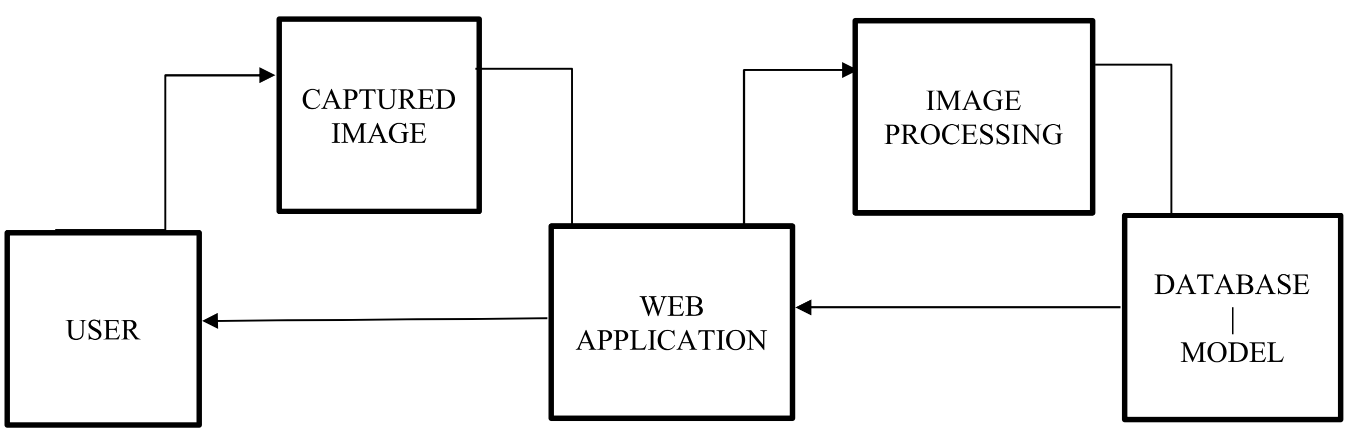


4.1 DATA FLOW DIAGRAMS

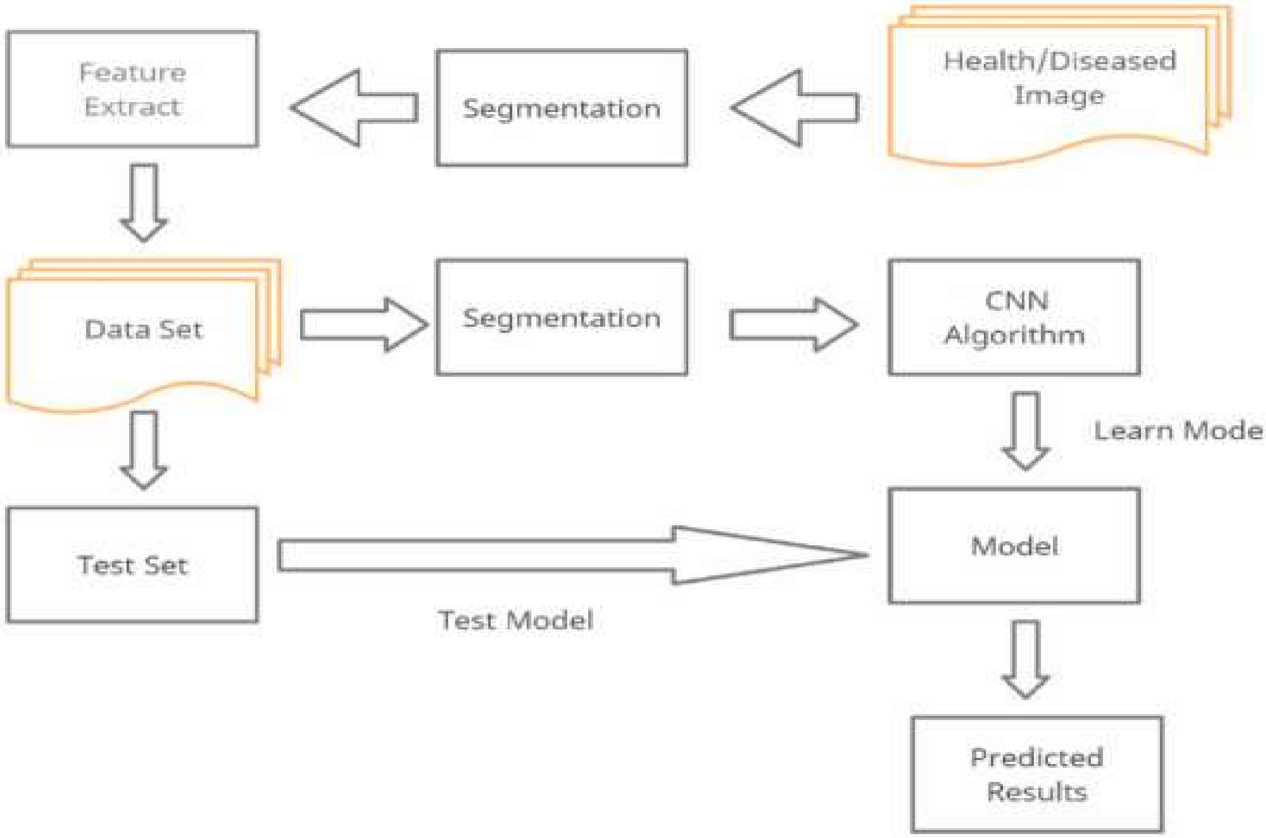
DFD level 0



DFD level 1



DFD level 2



Feature Extract
Segmentation
Health/Diseased Image
Data Set
Segmentation
CNN Algorithm
Learn Mode
Test Set
Model
Test Model
Predicted Results

4.2 UML DIAGRAM

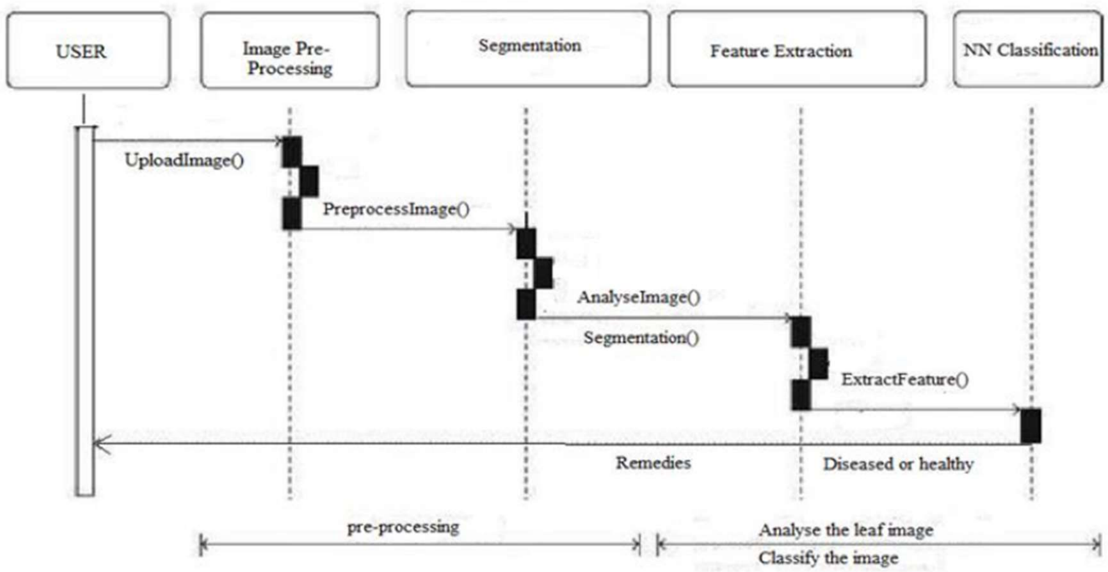


Fig1 . Sequence Diagram

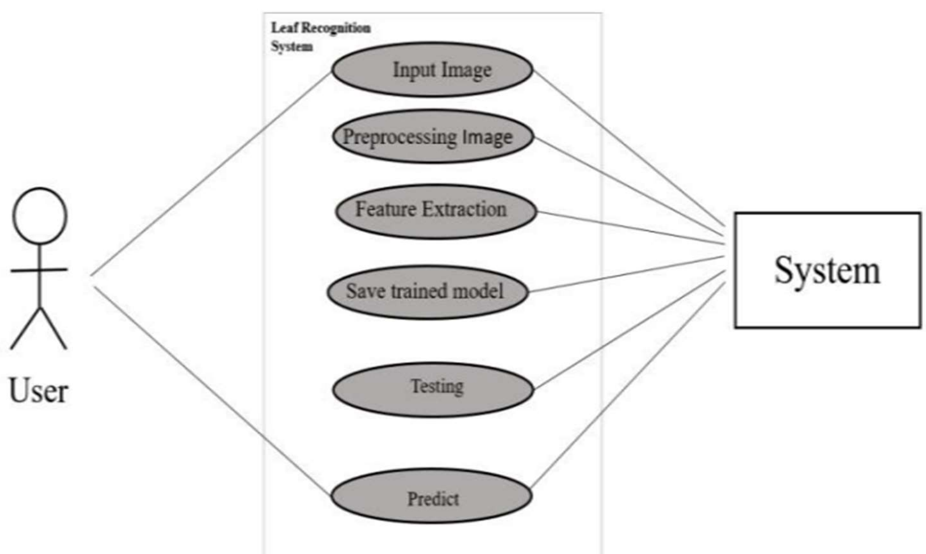


Fig 2. Use Case Diagram

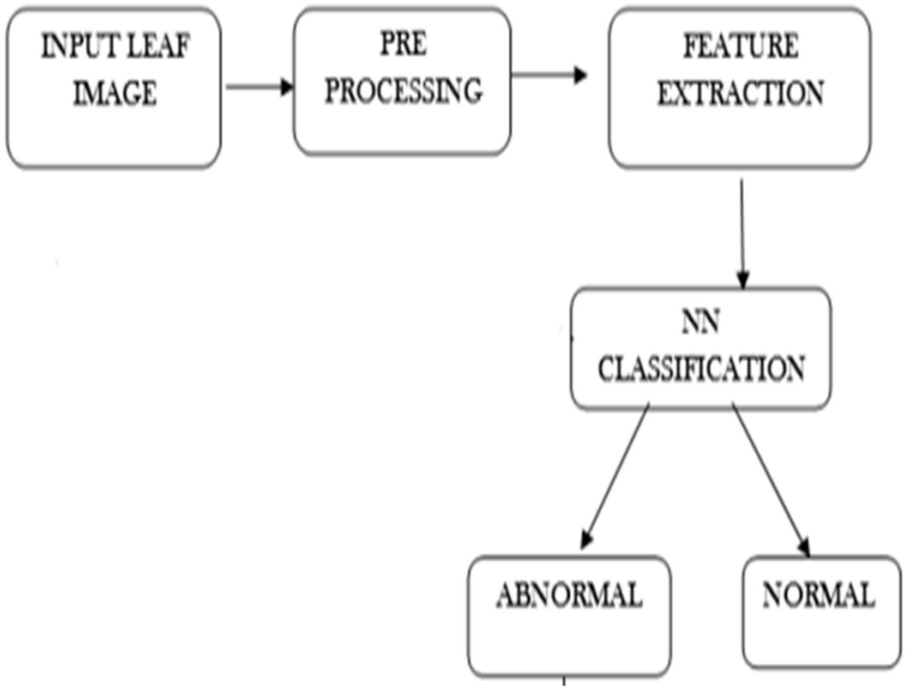
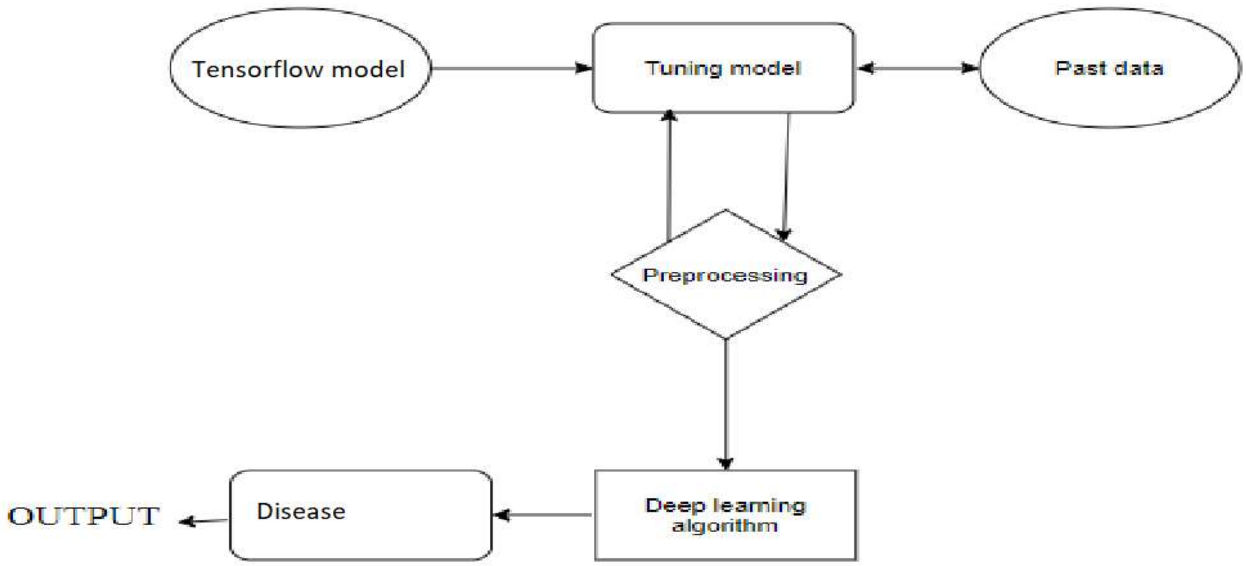


Fig 3. Activity Diagram

4.3 ENTITY RELATIONSHIP DIAGRAM



5. OTHER SPECIFICATION

5.1 Advantages

- **Increased Accuracy and Efficiency:** By integrating image processing, deep learning, and machine learning, the analyser provides highly accurate plant species identification, reducing the margin of error associated with human-based identification methods.

- **Time and Cost Savings:** Automation in plant identification significantly reduces the time and labour costs associated with traditional methods.

- **Educational and Agricultural Benefits:** The application caters to a wide range of users, from students and researchers to farmers, thereby promoting education and contributing to agricultural sustainability.

- **Longitudinal Plant Tracking:** The analyser allows for the tracking of plant growth and changes over time, enabling in-depth research and conservation efforts.

5.2 Applications

- **Automated Plant Species Identification:** The Botanic Image Analyzer employs image processing techniques to extract key features from plant leaves, which are used to train a deep learning model. This model can identify and classify plant species with remarkable accuracy, reducing the potential for human error.

- **Plant Information Retrieval:** In addition to leaf identification, the analyser provides comprehensive information about the recognized plant species. This includes details on medicinal properties, ecological habitats, and the detection of any diseases affecting the plant, making it a valuable resource for education and research.

- **Educational Benefits:** The Botanic Image Analyzer serves as an invaluable resource for learners, aiding in plant identification and education.

- **Agricultural Applications:** Farmers can utilize the analyser to identify crop diseases. Early detection of diseases enables timely action, leading to increased crop yield and reduced losses, thus promoting agricultural sustainability.

- **Conservation and Research:** This contributes to biodiversity studies and conservation efforts, helping to monitor and protect plant species.

5.3 Limitations

- **Concurrency:** Cannot take multiple images as input at a time.
- **Digital Image:** Doesn't work on images taken from google.
- **Artificial Intelligence:** AI can be Integrated for reinforcement learning.
- **Dependency on Image Quality:** The accuracy of disease detection can be affected by the quality of images uploaded. Low-quality or blurry images might lead to inaccurate results.

- **Updates and Maintenance:** The application needs regular updates to include new disease patterns, improve accuracy, and fix bugs. Without proper maintenance, the system can become outdated and less reliable over time.

- **Accessibility:** The application might not be accessible to users with disabilities if it lacks proper accessibility features.

- **Dependency on Internet:** The web application requires an internet connection to function, making it inaccessible in areas with poor connectivity.

6. PROJECT IMPLEMENTATION

6.1 Overview of Project Modules

- **Image Upload Module:**

  Image Upload Interface: Allows users to upload images of plant leaves for disease detection.

  Image Pre-processing: Processes the uploaded images to enhance quality and prepare them for analysis.

- **Disease Detection Module:**

  Image Analysis: Utilizes machine learning or deep learning algorithms to analyze the uploaded images and identify potential diseases.

  Disease Classification: Classifies the detected diseases based on the symptoms observed in the images.

- **Result Presentation Module:**

  Disease Identification: Displays the identified diseases along with relevant information, such as causes, symptoms, and treatments.

6.1.1 Model Details

- **Dataset and Pre-Processing**

  The Plant Village project provides a public dataset of plant leaf images. The dataset has 3171 RGB images of leaves that belong to four or five classes. The classes are based on the diseases that affect the leaves, such as black rot, scab, and cedar rust(Table.1). The other class is for healthy leaves that have no disease. The leaf images of size 256 × 256 for each class were taken with a simple background in a lab setting at different stages of plant growth. Fig.1 shows some examples of images from each class. The classes are named after the diseases or the healthy condition of the leaves. The Plant Village dataset is widely used in research. It is important to have diverse images of leaves in the dataset so that the models can learn different features during training. This helps to make the deep CNN model more generalizable. Augmentation is a technique to create artificial variations of the images. This work uses some transformations like shift, shearing, scaling, zoom, and flipping to change the images. These transformations make small changes in the images which help to add diversity to the training set.

This helps to prevent overfitting and makes the model more robust and adaptable which helps model to achieve tolerance and generalization.Fig.2 shows how the data augmentation techniques work. Fig. 3 shows all the different species of plant in the dataset.

- **Convolutional Layer**

The convolutional layer applies a filter (kernel) to an input image to detect features. The convolutional layer creates feature maps that show where and how much a feature is present in the previous layers by finding the local conjunctions. Basically, the convolutional layer has two parts: a linear convolution process and a non-linear activation function.

- **Pooling Layer**

The pooling layer is responsible for reducing the resolution of the feature maps produced by the convolutional layers. It shrinks the size of activation maps that have many parameters. This way, it lowers the computational cost, prevents overfitting and speeds up the training process. The main pooling operations are max, min, average. However, max pooling is the most popular and selects the highest value from each input patch. The pooling operation only modifies the dimensions n_H and n_W ,n_C remains affected.

- **Fully Connected Layer**

The convolutional neural network (CNN) has fully connected (FC) or dense layers that are similar to the layers in regular neural networks and usually connected at the end of a CNN to build output layers with a specific number of outputs for the better result. The Fully-Connected layers work on 1-D data. Flatten layer converts the 2-D output of previous layers into a 1-D format. The FC layers perform two kinds of functions: linear and non-linear transformations.

6.2 Tools and Technologies Used

The development of a leaf disease detection web application involves various tools and technologies to build different modules and ensure the application's functionality, scalability, and usability. Here are some commonly used tools and technologies for such projects:

- **Front-End**
  Streamlit : Framework for web Interface Development

- **Back-End**
  Python: Backend programming languages used to develop server-side logic.

- **Tool used**
  Jupyter Notebook

- **Libraries**
  Pandas: A powerful library for data manipulation and analysis, offering data structures like DataFrames and Series, along with tools for reading and writing data from various file formats.
  NumPy: The fundamental package for scientific computing in Python, providing support for large arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently.
  TensorFlow: An open-source machine learning framework developed by Google Brain for building and training deep learning models. It provides a comprehensive ecosystem of tools, libraries, and community resources for machine learning development.
  Keras: A high-level neural networks API built on top of TensorFlow, designed for fast experimentation and prototyping of deep learning models. It offers a user-friendly interface to build and train neural networks with minimal code.
  OpenCV: Open Source Computer Vision Library is a comprehensive library for computer vision and image processing tasks. It provides a wide range of functionalities for image and video analysis, including object detection, face recognition, and image enhancement.
  PyTorch: An open-source machine learning framework developed by Facebook's AI Research lab. It offers dynamic computational graphs and provides a flexible platform for building and training deep learning models, with strong GPU acceleration support.
  PIL (Pillow): The Python Imaging Library (PIL) or its fork Pillow is a library for opening, manipulating, and saving many different image file formats in Python. It provides simple and powerful image processing capabilities, including image resizing, filtering, and color adjustments.

6.3 Algorithm Details

In this model we have used a deep CNN model which is best suited for image based classification. It is a class of deep learning neural networks. For our model we included four-Convolutional layers containing 4 Max-pooling layers. Each Convolutional layer applies filters to inputs thus extracting spatial features in the dataset. After every Convolutional layer, we applied ReLU activation function to make the model non-linear, which enables the model to capture and learn complex patterns. Max-pooling layers reduces the spatial dimensionality of the input feature map while retaining all the important information within some local neighbourhood.

Flatten layers reduce the result to 1-Dimensional output to be further fed to fully connected layers. Fully connected layers learn the global patterns and relationships in the 1-dimensional result passed onto by a flatten layer. Fully connected layers are often called Dense layers, they perform global reasoning on the features extracted by convolutional and pooling layers.

This allows the network to capture complex relationships between features from different spatial locations. Dense layers enable complex decision making based on the extracted features .The details about the model has been shown in fig.5 and Table 1.

Once the machine learning model has been trained it is beneficial to save the model with all its parameters. Once it has been saved we can load the pre-trained model as and when needed and make classification without having the need to train the model from scratch a ga in. This saves both time and system resources which leads to faster classifications. Saving everything by creating a single archive file that contains everything using the Tensor-Flow, Save Model format (H5 format) and Creating a JSON file that only contains the architecture / configuration.

7. SOFTWARE TESTING

7.1 Type Of Testing

In deep learning models for plant disease detection, various types of testing are crucial to ensure models accuracy, robustness and generalizability. Here are the primary types of testing can be performed:

1. **Training-Validation-Testing split**

- Training set: Use to train the model
- Validation set: used to tune hyperparameters and prevent overfitting
- Test set: Used to evaluate the model's performance on unseen data

2. **Cross-Validation**

- k-Fold Cross-Validation: The data is divided into k subsets, and the model is trained and validated k times, each time using a different subset as the validation set and the remaining k-1 subsets as the training set. This helps in assessing the model's stability and robustness.

3. **Stratified Sampling**

- Ensures that each set (training, validation and testing) maintains the same proportion of classes as in the original dataset, which is important for imbalanced datasets

4. **Performance Metrics Evaluation**

- Accuracy: Measures the percentage of correctly classified instances
- Precision, Recall, and F1 Score: Especially important in cases of class imbalance
- Confusion Matrix: Provides a detailed breakdown of true positives, true negatives, false positives, and false negatives

5. **Error Analysis**
   - Misclassification Analysis: In-depth analysis of instances where the model predicted incorrectly to understand potential weaknesses.
   - Visual Inspection: Reviewing misclassified images to identify common patterns or errors.

6. **Robustness Testing**
   - Noise Addition: Testing the model's performance on images with added noise to simulate real-world conditions.
   - Adversarial Testing: Introducing subtle perturbations to images to see if the model can still correctly classify them.

7. **Generalization Testing**
   - Cross-Dataset Evaluation: Evaluating the model on a different dataset that was not used during training to assess its ability to generalize to new data
   - Geographical and Environmental Variations: Testing the model on images from different geographical locations or environmental conditions to ensure it generalizes well across diverse scenarios.

8. **Real-World Testing**
   - Field Trials: Deploying the model in real-world agricultural settings to test its performance in detecting plant diseases under practical conditions.
   - User Feedback: Collecting feedback from farmers or agronomists to refine and improve the model based on real-world usage.

9. **Incremental Learning and Continual Testing**
   - Online Learning: Continuously updating the model with new data as it becomes available.
   - Lifelong Learning: Ensuring the model retains previously learned information while adapting to new diseases or conditions

10. Scalability Testing

- Performance on Large Datasets: Assessing the model's efficiency and speed when processing large amounts of data
- Deployment Scalability: Ensuring the model can be deployed on various hardware configurations, from high-performance servers to mobile devices

11. Explainability and Interpretability Testing

- Saliency Maps and Attention Mechanisms: Using techniques to understand which parts of the image the model focuses on when making decisions.
- Model Transparency: Ensuring that the model's decision-making process can be understood and trusted by end-users, such as farmers and agricultural experts.

These testing methodologies help ensure that a deep learning model for plant disease detection is accurate, reliable, and applicable in real-world agricultural environments.

7.2 Test Cases and Test Results

7.2.1 Test Cases

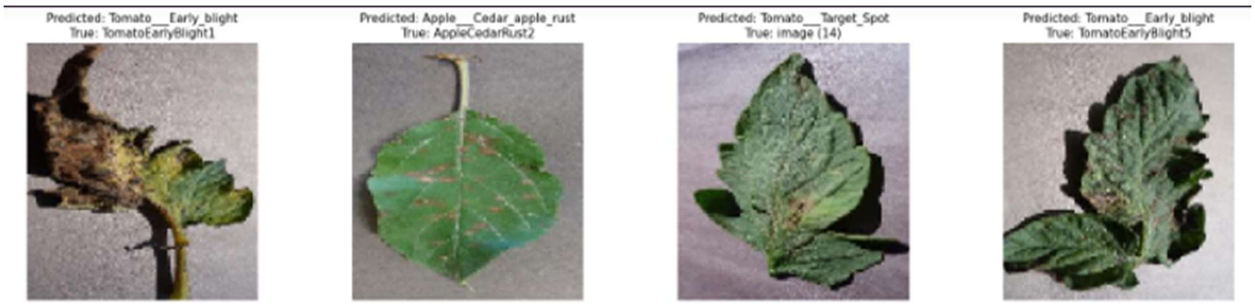
- Verify image upload functionality with valid image formats (e.g., JPEG, PNG).
- Verify error message for invalid image formats.
- Verify disease detection accuracy with known disease images.
- Verify multiple disease detection in a single image.
- Verify disease detection accuracy with known disease images.
- Verify confidence score or probability calculation.
- Verify disease information displayed for detected diseases.
- Verify responsive design across different devices and screen sizes.

7.2.2 Test Results

- Image uploaded successfully.
- Error message displayed for invalid image format
- Disease detected accurately.
- Multiple diseases detected in a single image.
- Disease information displayed correctly.
- Confidence score displayed correctly
- Disease information displayed correctly.
- Responsive design working across devices.

```python
print(
    f"Train Accuracy : {train_acc}\nTest Accuracy : {test_acc}\nValidation Accuracy : {validation_acc}"
)
```

```
Train Accuracy : 96.7
Test Accuracy : 98.9
Validation Accuracy : 98.7
```



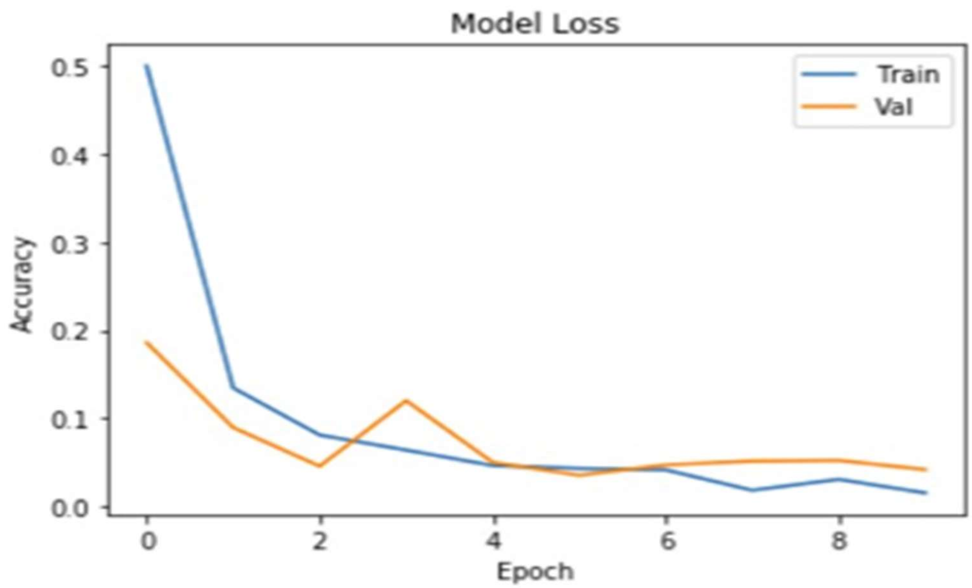
In [45]:

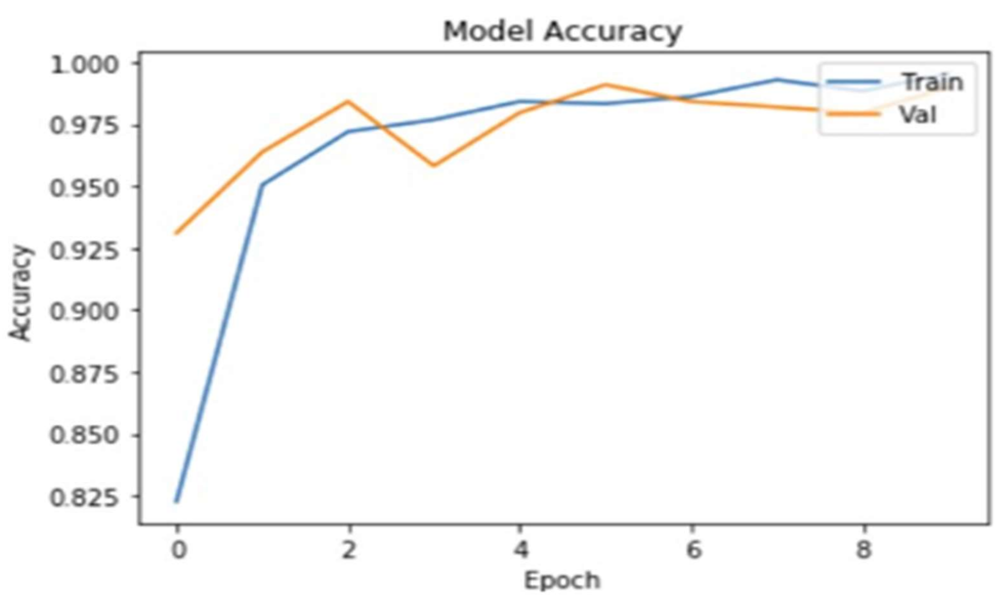
```python
# getting all predictions (actual label vs predicted)
for i, (img, label) in enumerate(test):
    print('Label:', test_images[i], ', Predicted:', predict_image(img, model))
```

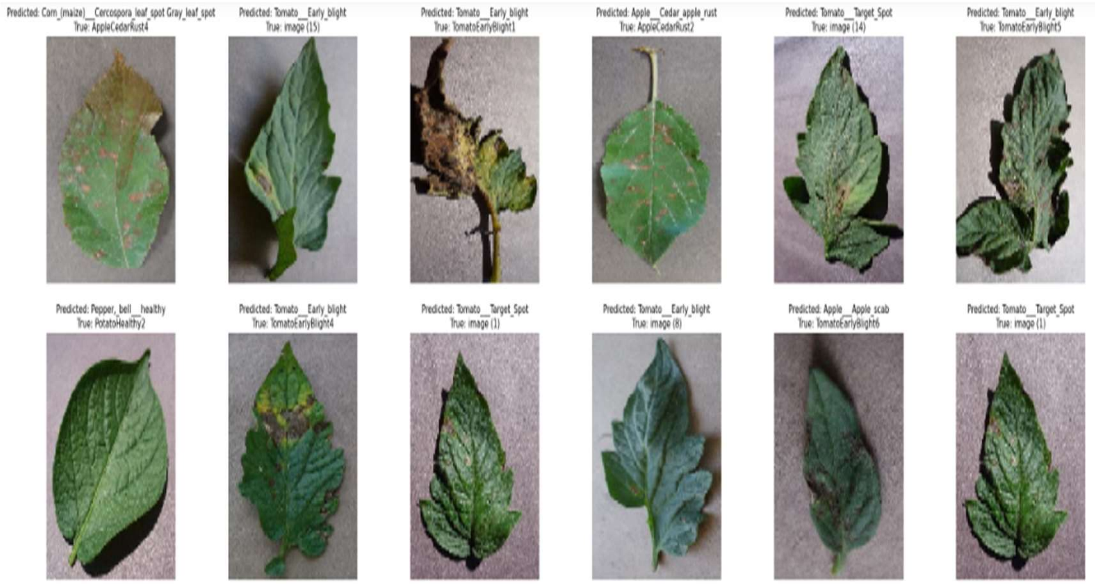
```
Label: AppleCedarRust1.JPG , Predicted: Apple___Cedar_apple_rust
Label: AppleCedarRust2.JPG , Predicted: Apple___Cedar_apple_rust
Label: AppleCedarRust3.JPG , Predicted: Apple___Cedar_apple_rust
Label: AppleCedarRust4.JPG , Predicted: Apple___Cedar_apple_rust
Label: AppleScab1.JPG , Predicted: Apple___Apple_scab
Label: AppleScab2.JPG , Predicted: Apple___Apple_scab
Label: AppleScab3.JPG , Predicted: Apple___Apple_scab
Label: CornCommonRust1.JPG , Predicted: Corn_(maize)___Common_rust_
Label: CornCommonRust2.JPG , Predicted: Corn_(maize)___Common_rust_
Label: CornCommonRust3.JPG , Predicted: Corn_(maize)___Common_rust_
Label: PotatoEarlyBlight1.JPG , Predicted: Potato___Early_blight
Label: PotatoEarlyBlight2.JPG , Predicted: Potato___Early_blight
```

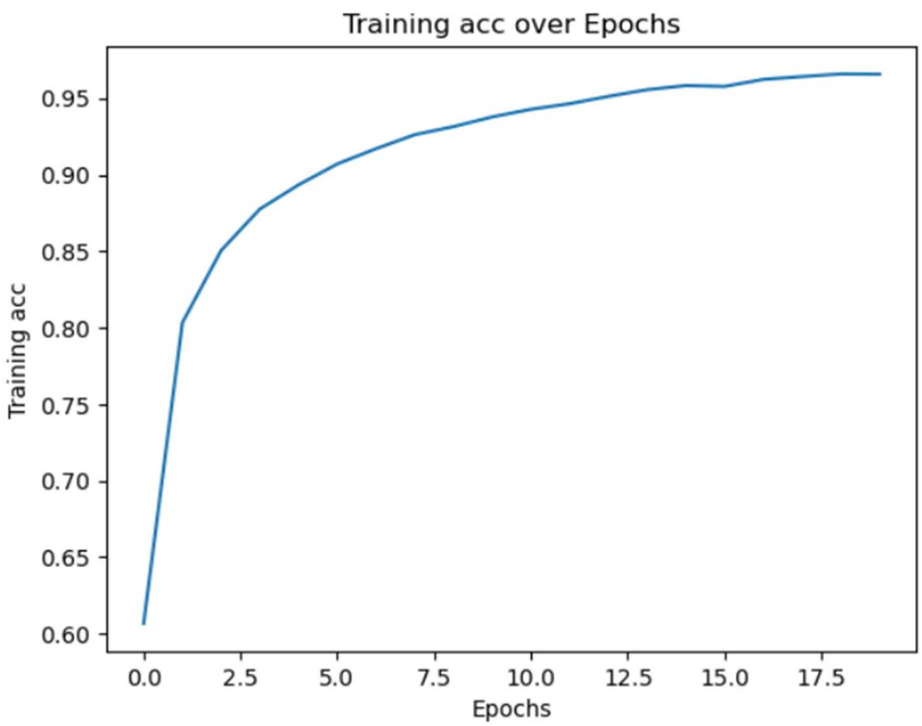
8. RESULTS

8.1 Outcomes

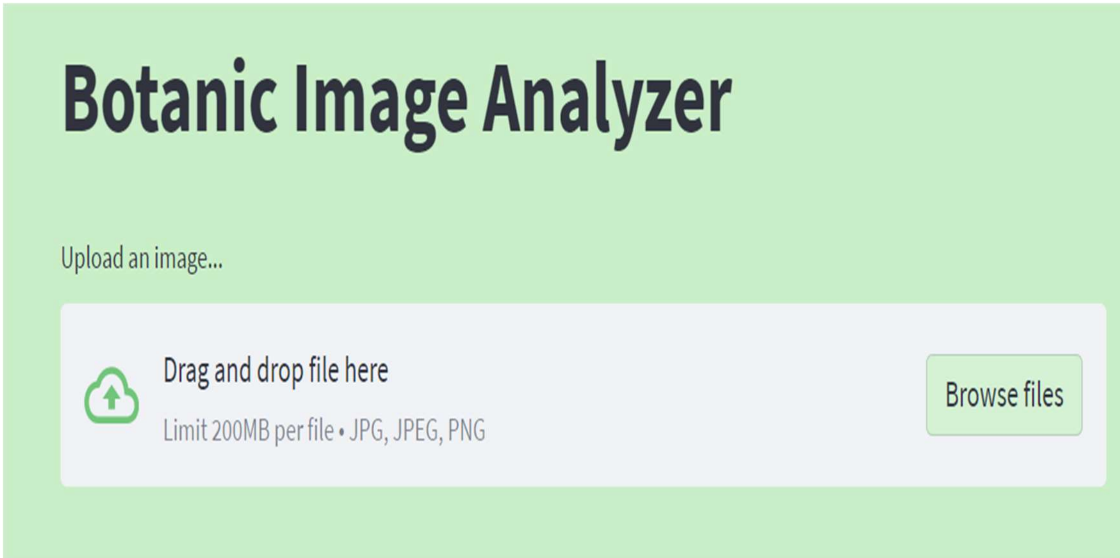


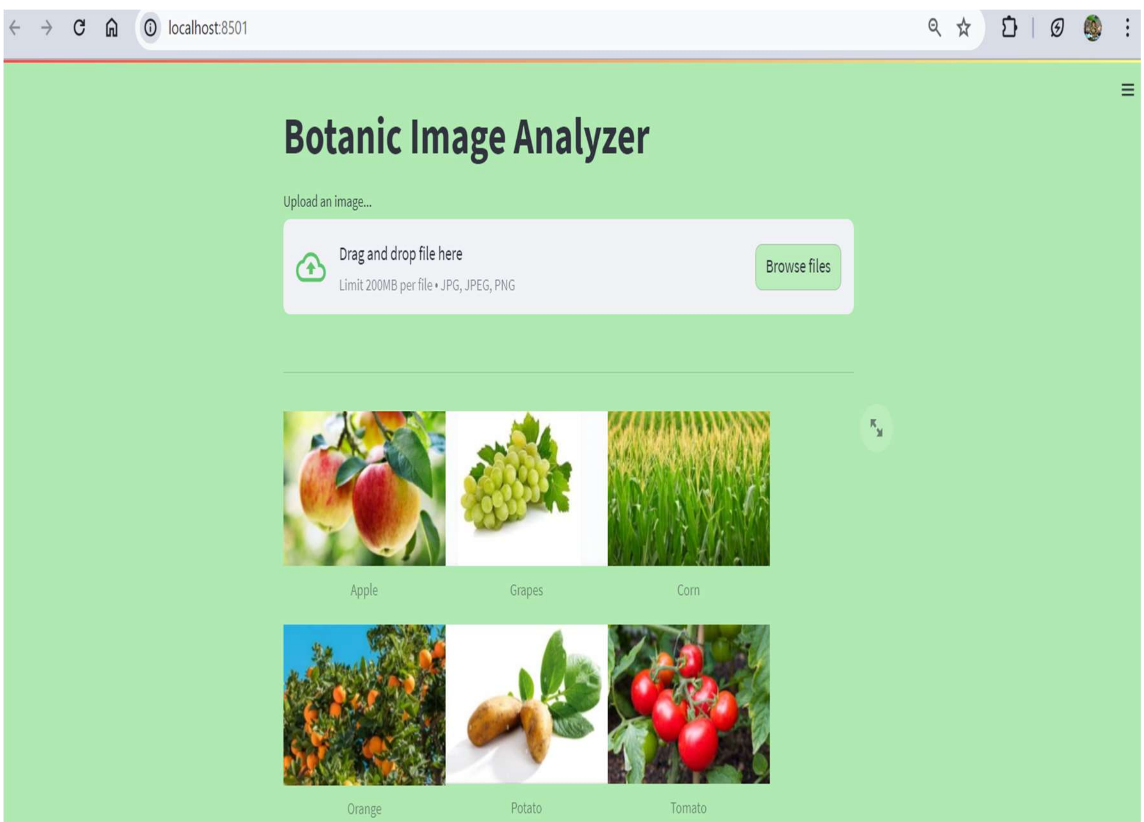


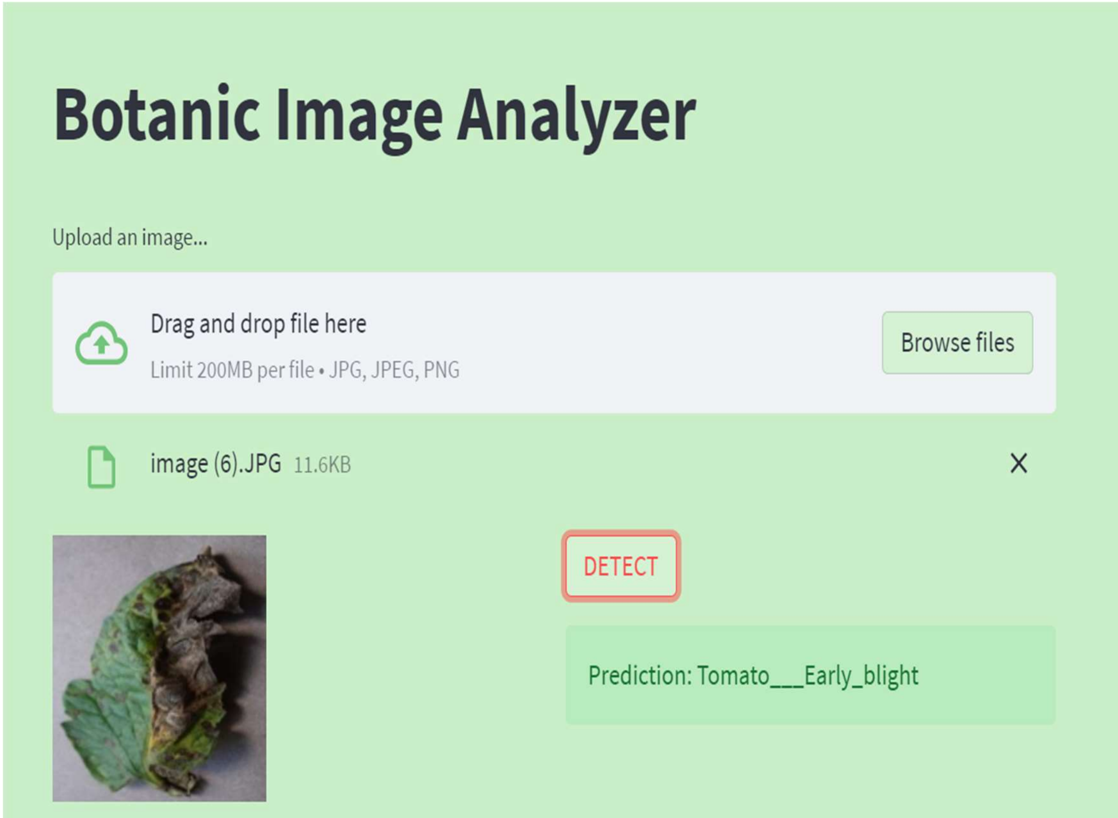

Predicted: Corn_(maize)__Cercospora_leaf_spot Gray_leaf_spot
True: AppleCedarRust4
Predicted: Tomato__Early_blight
True: image (15)
Predicted: Tomato__Early_blight
True: TomatoEarlyBlight1
Predicted: Apple__Cedar_apple_rust
True: AppleCedarRust2
Predicted: Tomato__Target_Spot
True: image (14)
Predicted: Tomato__Early_blight
True: TomatoEarlyBlight5
Predicted: Pepper,_bell__healthy
True: PotatoHealthy2
Predicted: Tomato__Early_blight
True: TomatoEarlyBlight4
Predicted: Tomato__Target_Spot
True: image (1)
Predicted: Tomato__Early_blight
True: image (8)
Predicted: Apple__Apple_scab
True: TomatoEarlyBlight6
Predicted: Tomato__Target_Spot
True: image (1)


Training acc over Epochs
Training acc
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epochs

8.2 Screenshots






Botanic Image Analyzer
Upload an image...
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
image (6).JPG 11.6KB
DETECT
Prediction: Tomato___Early_blight

9. CONCLUSION AND FUTURE WORK

Conclusion:

This project includes various web technologies and works as we wanted. It helps researchers, botanists, etc. in understanding plant properties by only leaf images. Also it helps in classifying the disease a plant is suffering from (if any). Developing this is a challenging task as a working model has never been developed and the most important part of this model is its deep learning model which predicts plant properties with great accuracy. Concluding this we put all the effort we could for this project.

Future scope:

The future scope for leaf image identification and disease detection projects is promising, and there are several areas where these projects can continue to evolve and make significant contributions:

- Expansion to New Plant Species
- Improved Accuracy and Generalization
- Robustness to Environmental Variations
- Climate Resilience
- Integration of Artificial Intelligence

The future of leaf image identification and disease detection projects holds the potential to significantly impact agriculture, conservation, and plant science by improving crop yields, promoting sustainable practices, and preserving biodiversity. These projects can play a vital role in addressing global challenges related to food security, environmental sustainability, and climate change.

Future research and development of the Botanic Image Analyzer may include improving its real-time capabilities, expanding the dataset to include more plant species, and enhancing the application's usability and functionality.

APPENDIX A

References:

- Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2016). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 16(11), 2022. https://www.mdpi.com/1424-8220/17/9/2022

- Jansen, M., Wentzell, P. D., & Fjeld, T. (2019). Machine learning for high-throughput stress phenotyping in plants. Trends in Plant Science, 24(2), 143-159. https://www.cell.com/trends/plant-science/fulltext/S1360-1385(15)00263-0

- Lee, S. H., Chan, C. S., Wilkin, P., Remagnino, P., & Nelson, J. D. (2015). Deep-plant: Plant identification with convolutional neural networks. In Proceedings of the British Machine Vision Conference (BMVC) (pp. 119.1-119.12). https://ieeexplore.ieee.org/abstract/document/7350839

- Minervini, M., Scharr, H., & Tsaftaris, S. A. (2015). Image analysis: The new bottleneck in plant phenotyping [Review]. IEEE Signal Processing Magazine, 32(4), 126-131.https://ieeexplore.ieee.org/abstract/document/7123050

- Pound, M. P., Atkinson, J. A., Townsend, A. J., Wilson, M. H., Griffiths, M., Jackson, A. S., ... & Murchie, E. H. (2017). Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. GigaScience, 6(10), 1-10. https://academic.oup.com/gigascience/article/6/10/gix083/4091592

- Singh, A., Ganapathysubramanian, B., Singh, A. K., & Sarkar, S. (2016). Machine learning for high-throughput stress phenotyping in plants. Trends in Plant Science, 21(2), 110-124. https://www.cell.com/trends/plant-science/fulltext/S1360-1385(15)00263-0

Books:

- Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
- Pattern Recognition and Machine Learning" by Christopher M. Bishop
- Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- Computer Vision: Algorithms and Applications" by Richard Szeliski

Websites:

- PlantVillage: https://www.kaggle.com/datasets/emmarex/plantdisease
- Kaggle: https://www.kaggle.com/c/leaf-classification
- GitHub: https://github.com/topics/plant-disease-detection

Additional resources:

- OpenCV: https://opencv.org/
- TensorFlow: https://www.tensorflow.org/
- PyTorch: https://pytorch.org/

APPENDIX B

PLAGIARISM REPORT

Plagiarism Detector.net

Nov 06, 2023

Plagiarism Scan Report

0% Plagiarized

100% Unique

Characters:4225	Words:567
Sentences:23	Speak Time: 5 Min

Excluded URL	None

Content Checked for Plagiarism

This project includes various web technologies and works as we wanted. It helps researchers, botanists, etc. in understanding plant properties by only leaf images. Also it helps in classifying the disease a plant is suffering from (if

Plagiarism Detector.net

May 29, 2024

Plagiarism Scan Report

0% Plagiarized

100% Unique

Characters:1843	Words:285
Sentences:15	Speak Time: 3 Min

Excluded URL	None

Content Checked for Plagiarism

In this model we have used a deep CNN model which is best suited for image based classification. It is a class of deep learning neural networks. For our model we included four-Convolutional layers containing 4 Max-pooling layers. Each Convolutional layer applies filters to inputs thus extracting spatial features in the dataset. After every Convolutional layer, we applied ReLU activation function to make the model non-linear, which enables the model to capture and learn complex patterns. Max-pooling layers reduces the spatial dimensionality of the input feature map while retaining all the important information within some local neighbourhood. Flatten layers reduce the result to 1-Dimensional output to be further fed to fully connected layers. Fully connected layers learn the global patterns and relationships in the 1-dimensional result passed onto by a flatten layer. Fully connected layers are

COPYRIGHT DETAILS:

1/11/24, 2:19 PM Form IV (Preview) :: Copyright Office

Print

Form XIV   Statement of Particulars   Statement of Further Particulars

FORM XIV
APPLICATION FOR REGISTERATION OF COPYRIGHT
[SEE RULE 70]

Diary Number: 1285/2024-CO/L

To

The Registrar of Copyrights,
Copyright Office,
Department of Industrial Policy & Promotion,
Ministry of Commerce and Industry,
Boudhik Sampada Bhawan,
Plot No. 32, Sector 14, Dwarka,
New Delhi-110075
Email Address: copyright@nic.in
Telephone No.: (Office) 011-28032496, 08929474194

Sir,

In Accordance with Section 45 of the Copright Act, 1957 (14 of 1957), I hereby apply for registration of Copyright and request that enteries may be made in the Register of Copyrights as in the enclosed Statement of Particulars.

1. I also send herewith duly completed the Statement of further Particulars relating to the work. (for Literary/Dramatic, Musical, Atristic works only) Literary/ Dramatic works

2. In accordance with rule 16 of the Copyright Rules, 1958, I have sent by prepaid registered post copies of this letter and of the Statement of Particulars and Statement of Further Particulars to other parties concerned as shown below:

Name of Party	Address of Party	Date of Dispatch
MAHESHWARI MOHAN CHITTAMPALLI	70/2, GANESH NAGAR, FAMOUS CHOWK, NEAR SONALI PHOTO STUDIO, NAVI SANGHVI, PUNE-411027	
DR. NALINI JAGTAP	DAPODI, PUNE-411012	
SAURABH PRAKASH	BHOJPUR, BIHAR-802163	
SHIVANSH SHUKLA	GORAKHPUR, UTTARPRADESH-273001	
KSHITIJ TRIPATHI	LUKHNOW, UTTARPRADESH-226001	
ANIKET UDHANE	DHAMANGAON, AMRAVATI-444709	

[See columns 7,11,12, and 13 of the Statement of Particulars and party referred in col.2 (e) of the Statement of Further Particulars.]

3. The prescribed fee has been paid, as per details below: 500/-

https://copyright.gov.in/FormIVPreview/frmFormIVPreview.aspx 1/2

Payment ID	Payment Date	Amount	Bank Name	Payment Mode
335938	11/01/2024	500		

4. Communications on this subject may be addressed to:

MAHESHWARI MOHAN
CHITTAMPALLI
70/2, GANESH NAGAR,
FAMOUS CHOWK, NEAR
SONALI PHOTO STUDIO,
NAVI SANGHVI, PUNE-
411027-411027
9860863164

5. I hereby declare that to the best of my knowledge and belief, no person, other than to whom a notice has been sent as per paragraph 2 above any claim or interest or dispute to my copyright of this work or its use by me.

6. I hereby verify that the particulars given in this Form and the Statement of Particulars and Statment of Further Particulars are true to the best of my knowledge, belief and information and nothing has been concealed there from.

List of Enclosures:

1. 2 Copies of Work

2. DD/IPO of Rs.500 Per Work

3. Authorization from author/publisher

4. If the application is being filed through attorney , a specific Power of Attorney in original duly signed by the applicant and accepted by the attorney

Place:

Date: 11/01/2024

For : MRS. MAHESHWARI MOHAN CHITTAMPALLI

Proprietor

Copyright Office

Copyright Office
Government of India

Acknowledgement Slip
(Date: 11/01/2024)

Diary Number: **1285/2024-CO/L**	Form Received: **Online**
Copyright Reg. of: **Literary/ Dramatic**	Titled: **BOTANIC IMAGE ANALYZER**

Communication Address

Name	Address	Phone Number
MAHESHWARI MOHAN CHITTAMPALLI	70/2, GANESH NAGAR, FAMOUS CHOWK, NEAR SONALI PHOTO STUDIO, NAVI SANGHVI, PUNE-411027-411027	9860863164

Financial Details

Payment ID	Amount	Bank Name	Payment Mode	Payment Date
335938	500			11/01/2024

* For future communication please mention this DIARY No.

INSTRUCTIONS

For the purpose of processing the application, following documents are mandatory to send by post along with the acknowledgement slip(Office Copy).

1.	2 Copies of work
2.	DD/IPO of Rs.(as applicable) per work favouring Registrar Copyright Office payable at New Delhi (Not applicable for online payment)
3.	Authorization from author/publisher
4.	If the work is being used on goods or capable of being used on the goods
5.	If the application is being field through attorney, a specific power of attorney in original duly signed by the applicant and accepted by the attorney
6.	Search Certificate from Trade Mark Office(TM-60) (*Only in case of Artistic work*).
7.	Applicant must take a print out of the application, sign it and send along with the other documents.

Kindly send the above documents within 30 Days from the date of online submission on the following address given by herewith: .

Office of the Registrar of Copyrights
Copyright Office, Department for Promotion of Industry & Internal Trade
Ministry of Commerce and Industry
Boudhik Sampada Bhawan,
Plot No. 32, Sector 14, Dwarka,
New Delhi-110078
Email Address: copyright@nic.in
Telephone No.: 011-28032496

APPLICANT'S COPY
THANK YOU