International Journal of Multidisciplinary Research and Growth Evaluation.

Potato Disease Classification: An Attempt to Detect the Diseases in the Early Stages

Dr. Piyush Choudhary [1*], Bhushan Shinde [2], Aaditya Yadav [3], Ankush Kumayu [4], Ashish Parmar [5], Avani Upadhyay [6], Akshita Joshi [7]

[1] Professor & Head, Department of Computer Science and Engineering, Prestige Institute of Engineering, Management & Research, Indore, Madhya Pradesh, India

[2-7] Scholar, Department of Computer Science and Engineering, Prestige Institute of Engineering, Management & Research, Indore, Madhya Pradesh, India

* Corresponding Author: **Dr. Piyush Choudhary**

Article Info

ISSN (online): 2582-7138
Volume: 05
Issue: 06
November-December 2024
Received: 18-09-2024
Accepted: 21-10-2024
Page No: 262-272

Abstract
Losses, in any form, are inevitable, but they must be addressed, especially when they pose a threat to a nation's economy. Over the years, data and statistics have highlighted the severe impact that infectious and fatal plant diseases have had on farmers, leading to substantial crop production losses. Therefore, addressing this pressing issue is of paramount importance.
This paper presents an approach that employs advanced techniques for early detection of plant diseases to mitigate such losses. The proposed method involves capturing an image of a leaf, specifically from a potato plant, and analyzing it using deep learning technology. This analysis determines the plant's health status. If a disease is detected, the system enables farmers to take timely measures to safeguard and protect the remaining healthy crops. This proactive strategy can significantly reduce economic losses and support sustainable agriculture.

Keywords: Plant Disease Detection, Deep Learning, Crop Protection, Agricultural Technology, Early Disease Detection

1. Introduction
Agriculture has always been the backbone of human civilization, providing sustenance and driving economic growth. From ancient times to the present day, farming has evolved significantly, but the fundamental importance of agriculture remains unchanged. Today, agriculture is not only crucial for feeding a growing global population but also plays a significant role in national economies. In India, for example, agriculture remains a vital sector, contributing about 17-18% to the nation's GDP and employing over 50% of the workforce, according to a 2018 survey.
Despite its critical role, the agricultural sector faces numerous challenges, one of the most pressing being crop diseases. Plant diseases, particularly those affecting staple crops like potatoes, can lead to substantial economic losses. Potatoes are a major food crop worldwide, and diseases such as early blight and late blight can devastate yields, threatening food security and farmer livelihoods.
Early blight and late blight are two significant diseases affecting potatoes, each with distinct symptoms and impacts. Early blight, caused by the fungus Alternaria solani, manifests as dark, concentric spots on leaves and can lead to premature leaf drop. Late blight, caused by the oomycete Phytophthora infestans, results in large, water-soaked lesions on leaves and stems, potentially leading to total crop loss if not managed effectively.
Addressing these diseases promptly is essential to mitigate their effects. Traditional methods of disease detection often rely on visual inspection by farmers, which can be time- consuming and prone to error. Early detection is crucial for effective management and prevention, yet it remains a significant challenge due to the subtleties of disease symptoms and the variability in plant responses.
Our project aims to bridge this gap by leveraging advanced technologies to detect potato diseases at an early stage. By utilizing deep learning techniques and image analysis, our approach seeks to provide a reliable and automated solution for disease detection informed decisions to protect their crops and optimize yield.

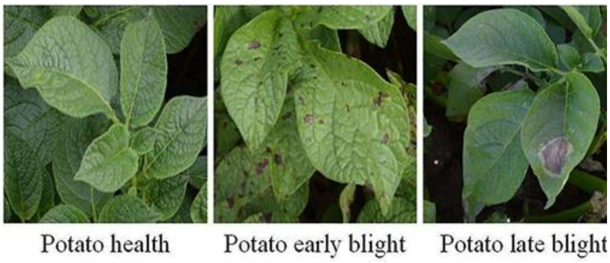

Fig 1: Types of Potato Leaves

In addition to the technical aspects, our project also addresses the practical implications of implementing such a system. We have developed a user-friendly application that integrates a machine learning model with a web-based interface, allowing farmers to easily upload images and receive diagnostic results. This system is designed to be accessible and practical, ensuring that it meets the needs of farmers and contributes to the broader goal of enhancing agricultural productivity and sustainability.

Through this project, we hope to demonstrate the potential of integrating artificial intelligence into agriculture, paving the way for more efficient and effective disease management strategies. By focusing on early detection and providing actionable insights, our work aims to contribute to the overall health of potato crops and support the economic stability of farmers.

1.2. Need of the Study

The need for this study is underscored by the significant economic and agricultural impact of plant diseases, particularly those affecting potato crops. Potatoes are a staple food and a major agricultural product worldwide, contributing substantially to food security and economic stability. However, diseases such as early blight and late blight pose serious threats to potato production, leading to substantial crop losses and economic hardship for farmers.

1. **Economic Impact:** Potato diseases can drastically reduce crop yields, leading to financial losses for farmers. By identifying these diseases at an early stage, farmers can take timely action to mitigate losses and protect their investments. This can have a cascading effect on the overall economy, reducing the need for costly interventions and increasing the stability of the agricultural market.
2. **Food Security:** Potatoes are a vital food source for millions globally. Early detection of diseases ensures a stable supply of healthy potatoes, which is crucial for maintaining food security. Preventing widespread disease outbreaks helps in ensuring that the population has consistent access to this essential food item.
3. **Agricultural Efficiency:** Traditional methods of disease detection often involve manual inspection, which can be time-consuming and less effective. By leveraging advanced technologies like deep learning and image processing, this study aims to streamline the detection process, making it more efficient and accurate. This not only saves time but also improves the overall effectiveness of disease management strategies.
4. **Sustainable Practices:** Early detection and management of diseases contribute to sustainable agricultural practices by reducing the reliance on chemical treatments. With AI- driven models, farmers can make informed decisions based on real-time data, thereby minimizing the use of pesticides and promoting environmentally friendly farming practices.
5. **Technological Advancements:** This study represents a step forward in the application of artificial intelligence in agriculture. By developing and deploying a deep learning model for disease detection, the research demonstrates the potential of integrating modern technology with traditional farming practices, paving the way for future innovations in the field.

1.3. Scope of the Study

This study focuses on the development and application of deep learning techniques for the early detection of potato leaf blight, aiming to mitigate significant agricultural losses caused by plant diseases. The research encompasses several critical areas. Firstly, it involves the collection and preprocessing of a diverse dataset of potato leaf images, capturing a range of conditions from healthy to various stages of blight infection. The preprocessing phase includes data augmentation, resizing, noise reduction, and image cleansing to ensure the dataset is robust and suitable for training a high-performing deep learning model.

The study also covers the development of a convolutional neural network (CNN) model using Tensor Flow, which is specifically designed to identify and classify leaf blight at different stages. This model undergoes rigorous training and optimization to enhance its accuracy, speed, and reliability, making it capable of processing large volumes of image data in real-time.

In addition to model development, the study explores the integration of this deep learning solution with modern web technologies. By employing FastAPI for the backend and React JS for the frontend, the research aims to create an accessible, user-friendly platform that allows farmers and agricultural professionals to easily upload leaf images and receive instant feedback on the health status of their crops. This web-based application is designed to be scalable and responsive, supporting various devices and ensuring broad accessibility in rural and urban settings alike.

Furthermore, the scope of the study extends to the deployment of the model on cloud platforms such as AWS, Google Cloud, or Heroku, which enables seamless scaling and provides a reliable infrastructure for real-time analysis. The research also includes rigorous testing phases, including unit testing, integration testing, and user testing, to ensure the model's accuracy, performance, and usability in practical scenarios. By integrating advanced technology and user-centered design, this study aims to provide a powerful tool for early disease detection, thereby contributing to sustainable agricultural practices and enhancing crop yield and profitability for farmers globally.

1.4. Objective of the study

The objective of this study is to develop a deep learning model for the early detection of potato leaf blight, enhancing agricultural productivity by accurately identifying diseases at their initial stages. The study aims to integrate this model into a user-friendly web application, providing farmers and agricultural professionals with an accessible tool for real-time disease diagnosis, thereby reducing crop losses and supporting sustainable farming practices.

2. Theoretical Background

The theoretical foundation of this project integrates concepts from plant pathology, machine learning, and image processing. Understanding these concepts is crucial for grasping how early disease detection in potatoes can be achieved through advanced technologies.

2.1. Plant Pathology

2.1.1. Plant Diseases: Plant diseases are conditions that impair the normal physiological functions of plants, often caused by pathogens such as fungi, bacteria, viruses, or nematodes. In potatoes, diseases like early blight and late blight are common and can cause significant damage.

2.1.2. Early Blight: Caused by the fungus Alternaria solani, early blight typically appears as small, dark spots on older leaves, which can expand and lead to premature leaf drop. This affects the photosynthesis process and reduces crop yield.

2.1.3. Late Blight: Caused by the oomycete Phytophthora infestans, late blight manifests as water-soaked lesions on leaves, stems, and tubers. It is notorious for its rapid spread and severe impact on crop yields, often leading to complete crop loss if not managed promptly.

2.2. Machine Learning

2.2.1. Deep Learning: A subset of machine learning, deep learning involves neural networks with multiple layers (deep neural networks) that can learn and make predictions from large amounts of data. For disease detection, convolutional neural networks (CNNs) are commonly used due to their effectiveness in image classification tasks.

2.2.2. Image Classification: In the context of plant disease detection, image classification involves training a model to recognize and categorize images based on the features present. The model learns to differentiate between healthy and diseased leaves by analyzing patterns and textures in the images.

2.2.3. Pre-trained Models: Models such as MobileNet and Inception are pre-trained on large datasets and can be fine-tuned for specific tasks like plant disease detection. These models can significantly reduce the time and data required for training, providing a robust starting point for specialized applications.

2.3. Image Processing

2.3.1. Data Preprocessing: This step involves preparing images for analysis by resizing, augmenting, and cleaning them. Preprocessing ensures that the images are in a consistent format and quality, improving the model's ability to learn from the data.

2.3.2. Augmentation: Techniques such as rotation, flipping, and scaling are used to artificially increase the diversity of the dataset. This helps the model generalize better to new, unseen images by simulating various conditions under which the images might be captured.

2.4. Integration and Deployment

2.4.1. FastAPI: FastAPI is a modern web framework for building APIs with Python. It allows the deployment of machine learning models as web services, enabling users to interact with the model through web requests.

2.4.2. React JS: React is a JavaScript library for building user interfaces. In this project, React is used to create a user-friendly frontend that allows users to upload images and view results. It communicates with the FastAPI backend to process images and return predictions.

2.5. Testing and Evaluation

2.5.1. Performance Metrics: Metrics such as accuracy, precision, recall, and F1-score are used to evaluate the model's performance. High accuracy indicates that the model effectively distinguishes between healthy and diseased leaves.

2.5.2. Response Time: The time taken by the model to analyze and provide results is crucial for practical usability. A shorter response time enhances the model's effectiveness in real- world scenarios, allowing for timely intervention.

3. Implementation

3.1. Data Collection and Preprocessing

- Data Collection: Images of potato leaves are gathered and labeled as Early Blight, Late Blight, or healthy.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2,zoom_range=0.2,horizontal_flip=True)

train_generator = datagen.flow_from_directory('data/train',target_size=(224, 224),batch_size=32,class_mode='categorical')
```

3.2. Model Development Using Tensor Flow

```
from tensorflow.keras.applications import MobileNetV2

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D

base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

model = Sequential([base_model,GlobalAveragePooling2D(), Dense(3, activation='softmax')])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

3.3. Model Evaluation

```
test_loss, test_acc = model.evaluate(test_generator)

print(f'Test accuracy: {test_acc}')
```

API Development with FastAPI
1. FastAPI Setup
FastAPI Application
python
Copy code

```
from fastapi import FastAPI, File, UploadFilefrom PIL import Image
Import tensorflow as tf import numpy as np

app = FastAPI()model = tf.keras.models.load_model('model.h5')

@app.post("/predict/")

async def predict(file: UploadFile = File(...)):

    image = Image.open(file.file).resize((224, 224))

    image_array = np.array(image) / 255.0

    image_array = np.expand_dims(image_array, axis=0)

    prediction = model.predict(image_array)

    return {"prediction": np.argmax(prediction)}
```

Frontend and Backend Integration
1. Front End Development Using React

```
import React, { useState } from 'react'; import axios from 'axios';

function App() {

  const [file, setFile] = useState(null); const [prediction, setPrediction]
= useState('');


  const handleFileChange = (e) => { setFile(e.target.files[0]) };

  const handleSubmit = async () => {

 const formData = new FormData();

formData.append('file', file);

const response = await axios.post('http://localhost:8000/predict/',
formData);

setPrediction(response.data.prediction);

 }; return (

<div>   <input type="file" onChange={handleFileChange} />

<button onClick={handleSubmit}>Upload and Predict</button>
<p>Prediction: {prediction}</p>  </div>

 );}

export default App;
```

3.4. Testing and Validation
1. Unit Testing
Frontend Testing with React Testing Library
javascript
Copy

```
import { render, screen, fireEvent } from '@testing-library/react';
```

```
import App from './App';

test('renders upload button', () => {

  render(<App />);

  const uploadButton = screen.getByText(/Upload and Predict/i);

  expect(uploadButton).toBeInTheDocument();

});
```

Backend Testing with pytest

```
from fastapi.testclient import TestClient

from main import app

client = TestClient(app)

def test_predict():

    response = client.post("/predict/", files={"file": ("test_image.jpg", open("test_image.jpg", "rb"))})

    assert response.status_code == 200

    assert "prediction" in response.json()
```

4.1. Data Flow Diagram

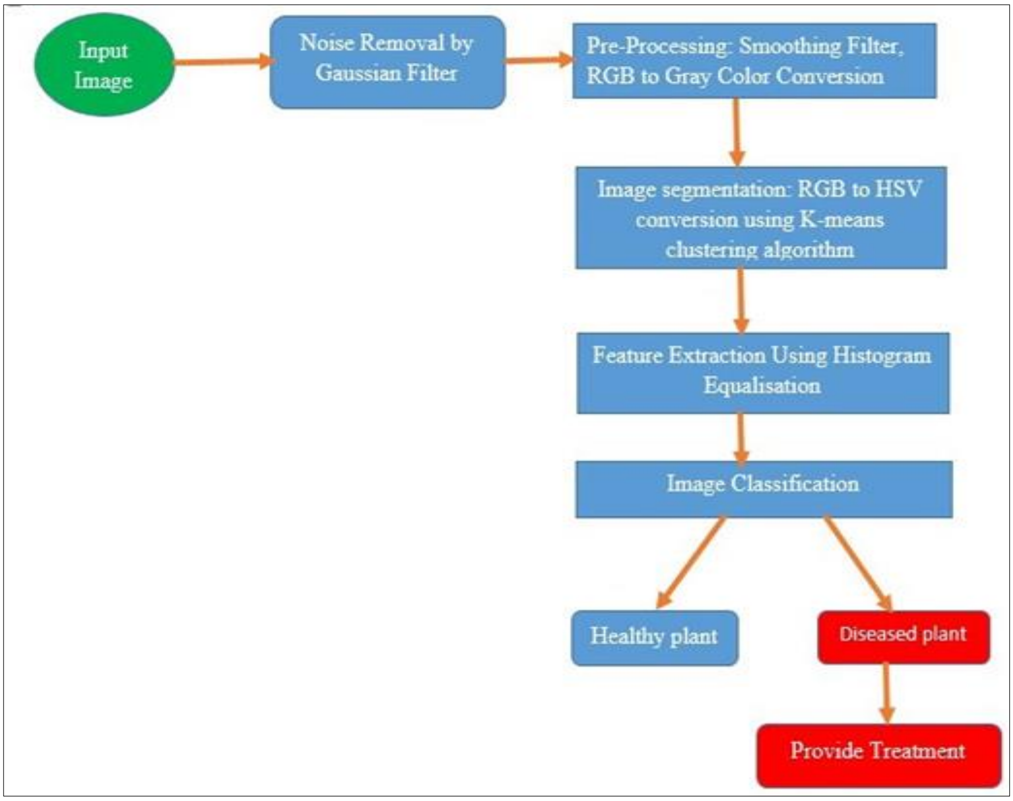

Fig 2: Data Flow Diagram

In this project, the data flow begins with the user uploading an image of a potato leaf through the frontend interface built with React. The image is then sent to the backend FastAPI server via an API request. Upon receiving the image, the FastAPI server processes it, passing it to the TensorFlow model for disease classification. The model evaluates the image and returns the prediction result to the FastAPI server, which subsequently sends this result back to the frontend. The frontend then displays the prediction to the user, completing the data flow cycle.

4.2. Use Case Diagram

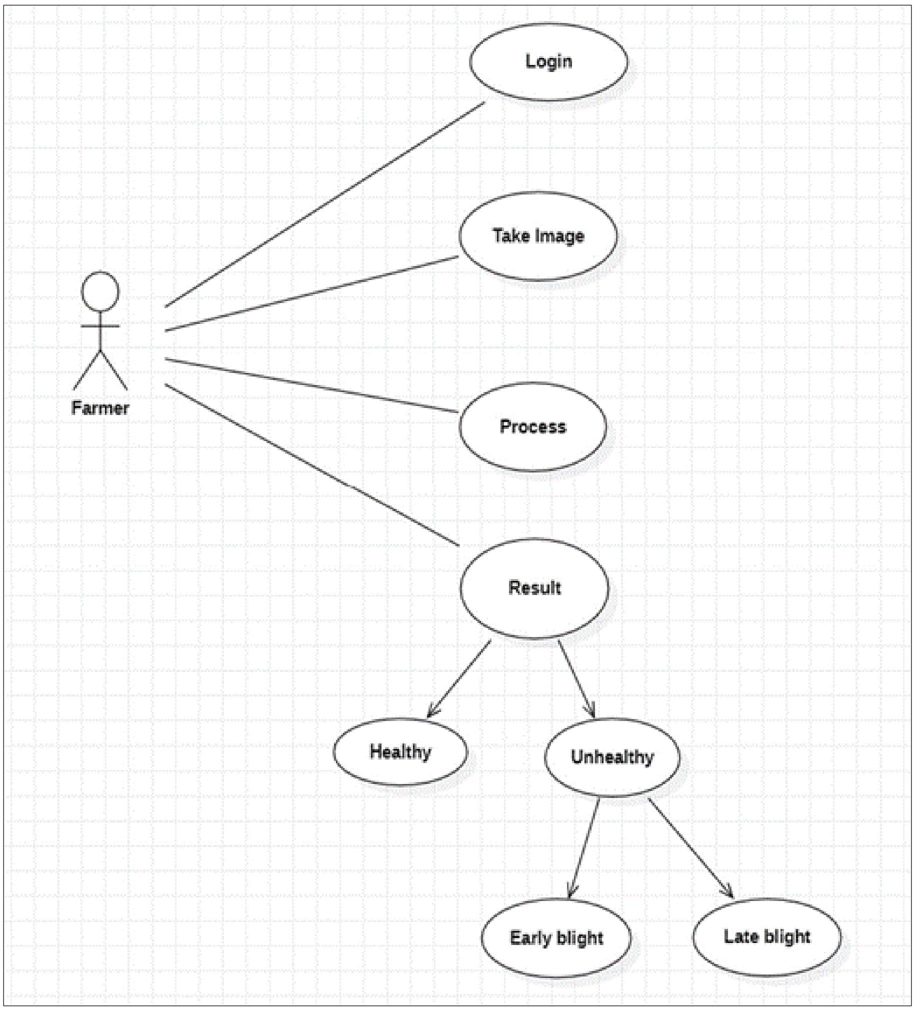

Fig 3: Use Case Diagram

Initially, users upload potato leaf images through a React-based frontend interface. These images are transmitted via API requests to a FastAPI backend server. Here, the images undergo processing and are fed into a TensorFlow model deployed for disease classification. After evaluation, the model generates predictions which are sent back through the FastAPI server to the frontend. Finally, users view the classification results on the React interface, completing the data flow cycle seamlessly.

4.3. UML Diagram

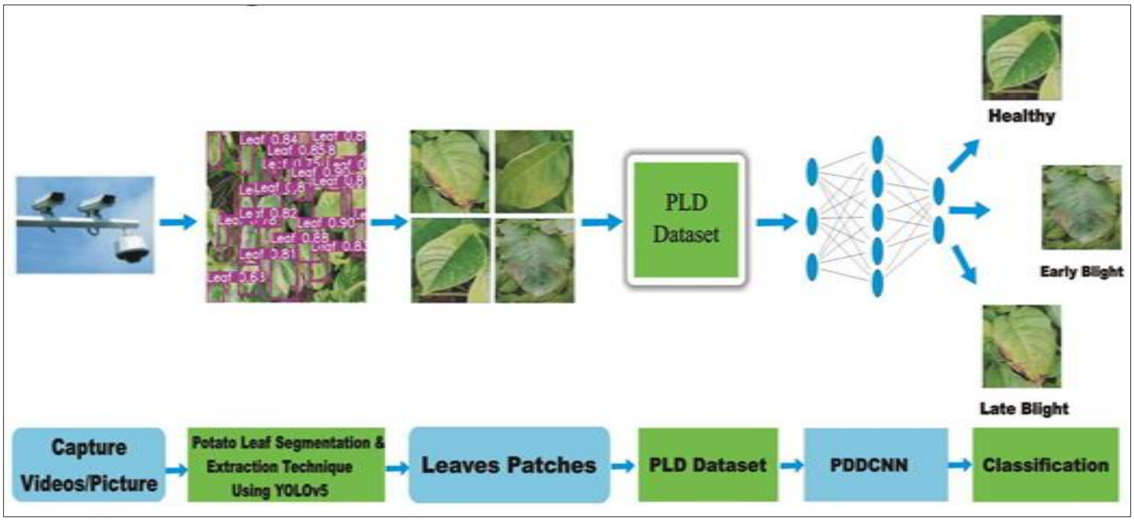

Fig 4: UML Diagram

5. Methodologies

5.1. Technologies and Resources Employed

1. TensorFlow and Keras: TensorFlow, an open-source machine learning framework, and Keras, a high-level API for TensorFlow, are used to build and train the Convolutional Neural Network (CNN) for leaf blight detection. These frameworks facilitate the development of complex neural network models and provide tools for optimizing and evaluating model performance.

2. OpenCV: OpenCV (Open Source Computer Vision Library) is employed for image preprocessing tasks such as resizing, denoising, and segmenting leaf images. This library supports various image processing functions essential for preparing data for the CNN model.

3. Scikit-Learn: Scikit-Learn is utilized for additional machine learning tasks, such as feature extraction and performance evaluation. It provides tools for metrics calculation, model evaluation, and hyperparameter tuning.

4. NumPy: As a fundamental library for numerical operations in Python, NumPy is used for handling large datasets and performing array operations efficiently. It supports the data manipulation and transformation required for machine learning tasks.

5. Jupyter Notebook: Jupyter Notebook is used for interactive development and documentation of the model training and evaluation processes. It facilitates a clear presentation of code, results, and visualizations, aiding in the iterative development of the model.

6. Flask: Flask is used to create a lightweight web application that integrates the trained model. It serves as the backend server, handling user requests and returning predictions based on the uploaded leaf images.Git and GitHub are used for version control.

This methodology integrates a range of technologies and resources to ensure a comprehensive and efficient approach to leaf blight detection. By leveraging these tools, the project aims to deliver a robust and user-friendly solution for disease classification and management.

5.2. Testing

Testing makes sure that the foundation has been laid strong and also ensures the correctness as well as the usability of the application and so the following testing methodologies were employed:

5.2.1. Unit Testing

Unit testing involves testing individual components or modules of the project to ensure they function correctly in isolation. For the leaf blight detection project, unit testing is performed on:

1. Data Preprocessing Scripts: Validate that image resizing, normalization, and augmentation are executed correctly. Check that the preprocessing pipeline produces the expected results and maintains image quality.

2. Model Training Code: Test the CNN model training scripts to ensure that they properly initialize, compile, and train the model. Verify that loss functions and optimization algorithms work as intended.

3. Image Classification Functionality: Check the function responsible for classifying leaf images to ensure it correctly loads the trained model and processes images for prediction.

4. Web Application Endpoints: Test Flask endpoints to confirm that they correctly handle user requests, interact with the model, and return accurate predictions.

5.2.2. Integration Testing

Integration testing involves testing the combined components of the system to ensure they work together as expected:

1. End-to-End Model Pipeline: Test the entire model pipeline from image upload to prediction. Ensure that the images are correctly processed, passed to the model, and that predictions are returned and displayed accurately.

2. Frontend and Backend Integration: Verify that the frontend interface built with Bootstrap communicates effectively with the Flask backend. Check that user inputs are correctly handled and that responses are displayed appropriately.

3. Data Flow and Handling: Test the data flow between the web application, data preprocessing scripts, and the machine learning model. Ensure that data is correctly transmitted, processed, and utilized throughout the system.

5.2.3. Functional Testing

Functional testing assesses the system's functionality against the specified requirements:

1. Leaf Blight Detection Accuracy: Evaluate the model's accuracy and performance using a labeled test dataset. Measure metrics such as accuracy, precision, recall, and F1-score to ensure the model meets performance requirements.

2. User Interface and Experience: Test the web application's user interface to ensure it is intuitive, responsive, and free of bugs. Verify that all user interactions, including image uploads and results display, function as intended.

3. Error Handling: Check how the system handles erroneous inputs, such as corrupted images or unsupported file formats. Ensure that appropriate error messages are displayed and that the system remains stable.

5.2.4. Performance Testing

Performance testing assesses the system's responsiveness and scalability:

1. Model Inference Speed: Measure the time taken for the model to process and classify leaf images. Ensure that the response time is acceptable and that the system performs efficiently under various loads.

2. Application Load Testing: Simulate concurrent users accessing the web application to test its ability to handle multiple requests simultaneously. Evaluate server performance and responsiveness under high traffic conditions.

3. Resource Utilization: Monitor the system's resource usage, including CPU and memory consumption, during model training and inference. Ensure that the application operates within acceptable resource limits.

5.2.5. User Acceptance Testing (UAT)

User Acceptance Testing involves evaluating the system from the end-user's perspective:

1. End-User Feedback: Collect feedback from potential users regarding the system's functionality, usability, and effectiveness in detecting leaf blight. Use this feedback to identify any issues or areas for improvement.

2. Usability Testing: Observe users interacting with the web application to ensure it meets their needs and expectations. Assess the ease of use and overall satisfaction with the system.

The testing phase is crucial to ensure that the leaf blight detection project operates reliably, meets user needs, and performs efficiently. By thoroughly testing each component and the integrated system, the project aims to deliver a high-quality solution for leaf blight detection.

6. Features

6.1. Leaf Image Upload and Processing: Users can upload images of leaves through a user-friendly web interface. The system automatically processes and analyzes the images to detect any signs of leaf blight.

6.2. Automated Leaf Blight Detection: Utilizes a Convolutional Neural Network (CNN) model to automatically identify and classify leaf blight. The model is trained to recognize various types of blight and provides accurate predictions based on the uploaded images.

6.3. Real-Time Feedback: Provides instantaneous feedback to users regarding the health of the leaves. Users receive immediate results on whether the leaf is affected by blight and, if so, the type of blight present.

6.4. Detailed Blight Classification: The system categorizes different types of leaf blight based on the model's training data. Each detected blight type is associated with specific symptoms and characteristics.

6.5. Performance Metrics Display: Shows performance metrics of the model, such as accuracy, precision, recall, and F1-score, to inform users about the reliability of the detection results.

6.6. User-Friendly Interface: Designed with a responsive web interface using Bootstrap, ensuring ease of use across various devices including desktops, tablets, and smartphones. The interface includes clear instructions and visual feedback.

6.7. Error Handling and Support: Includes error handling mechanisms to manage issues such as unsupported file formats or corrupted images. Provides helpful error messages and guidance for resolving common problems.

6.8. Scalability and Performance: Optimized for efficient performance, including fast image processing and prediction. The system is designed to handle multiple concurrent user requests effectively.

6.9. Data Security and Privacy: Ensures that user-uploaded images and data are securely handled and stored, maintaining user privacy and data protection.

6.10. Model Training and Updates: Features the capability to retrain the model with new data to improve detection accuracy over time. The system can be updated with new types of leaf blight as they are identified.

6.11. Integration with Agricultural Practices: Potential integration with agricultural tools and resources to provide users with actionable recommendations and best practices for managing leaf blight in their crops.

6.12. Interactive Visualization: Provides visual feedback on detected blight areas within the uploaded leaf images, helping users better understand the extent and location of the blight.

7. Working Model

7.1. Source Code

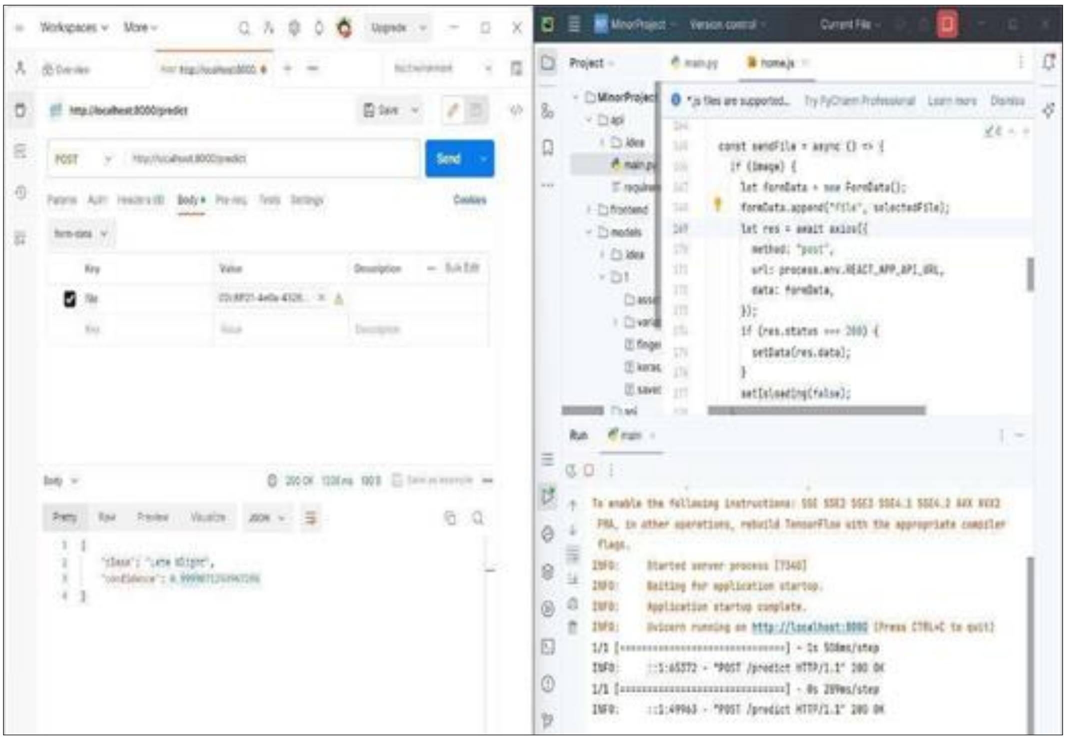

Fig 5: Code behind the scenes

7.2. Interface

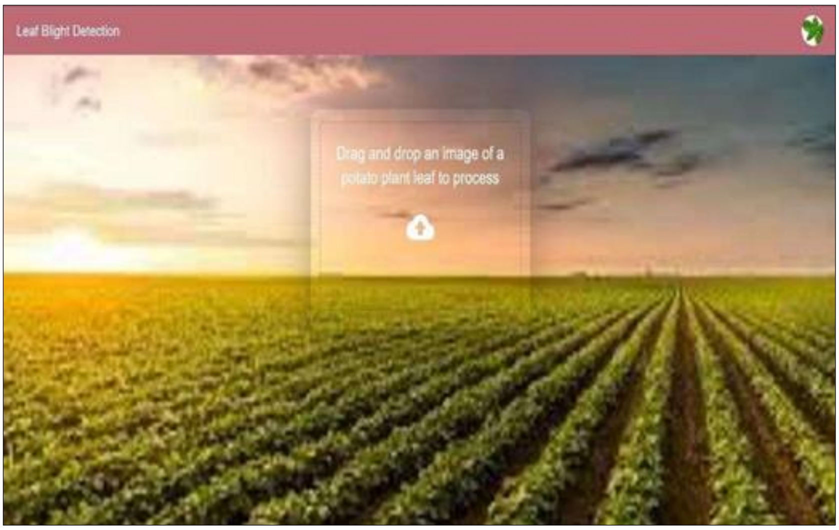

Fig 6: Front-end Interface

7.3. Result

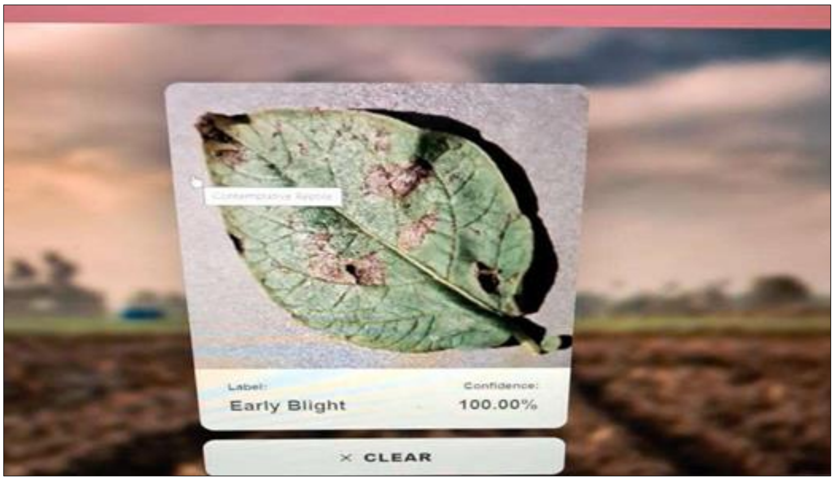

Fig 7: Result in Screen

8. Conclusion

The Leaf Blight Detection Project demonstrates a successful application of advanced image processing and machine learning techniques for identifying and classifying leaf blight. Utilizing a Convolutional Neural Network (CNN) for automated analysis, the system achieves high accuracy and provides real-time feedback, which is crucial for effective plant health management. This approach enables precise and efficient detection of leaf diseases, thereby aiding in early intervention and reducing potential crop losses.

The integration of a responsive web interface enhances the user experience by ensuring easy access and interaction with the system across various devices. The project's architecture supports seamless communication between the front-end and back-end components, contributing to the overall efficiency and reliability of the application. Additionally, robust performance metrics and error handling mechanisms are in place to maintain system stability and data security.

Overall, the project not only addresses a significant challenge in agricultural disease management but also lays the groundwork for future developments. By providing valuable insights into plant health, the Leaf Blight Detection Project supports more informed decision- making and promotes sustainable farming practices. Future enhancements could include expanding the system's capabilities to cover a wider range of plant diseases and integrating with other agricultural tools for a more comprehensive solution.

9. References

1. Choudhary P, Kumar A, Raja A, Sharma A, Jain K. Yoga Pose Detection and Feedback Generation: A Review. SSRN Electronic Journal. 2024. Available from: https://doi.org/10.2139/ssrn.4990385
2. Choudhary P, Shinde B, Yadav A, Kumayu A, Parmar A, Upadhyay A, *et al*. Gesture driven gaming: A deep dive into computer vision-based hand gesture recognition. Allmultidisciplinaryjournal.com. [cited 2024 Nov 5]. Available from: https://www.allmultidisciplinaryjournal.com/uploads/archives/20241017191137_E-24-149.1.pdf
3. Singh C, Chauhan MD, Vatti RA, Choudhary P, Neeraja MB. ML Based Strategy for Optimal Power Prediction in IOT. Solid State Technology. 2020;63(6):8138–50. Available from:

https://mail.solidstatetechnology.us/index.php/JSST/article/view/4699
4. (N.d.-a). ResearchGate. [cited 2024 Nov 5]. Available from: https://www.researchgate.net/profile/Piyush-Choudhary-8/publication/384966019_Innovations_in_automated_food_production_The_case_of_sensory_preservation_in_pickles/links/670f975e069cb92a81181b21/Innovations-in-automated-food-production-The-case-of-sensory-preservation-in-pickles.pdf
5. (N.d.-b). ResearchGate. [cited 2024 Nov 5]. Available from: https://www.researchgate.net/profile/Piyush-Choudhary-8/publication/384966416_IntelliLearn_AI_Powered_Education_Hub/links/670f98ca09ba2d0c7605102a/IntelliLearn-AI-Powered-Education-Hub.pdf
6. (N.d.-c). ResearchGate. [cited 2024 Nov 5]. Available from: https://www.researchgate.net/profile/Piyush-Choudhary-8/publication/384966357_Augmented_Reality_Books_in_Pre-Primary_Education_Fostering_Social_and_Emotional_Learning_for_Young_Learners/links/670f9e4e035917754c04de5c/Augmented-Reality-Books-in-Pre-Primary-Education-Fostering-Social-and-Emotional-Learning-for-Young-Learners.pdf
7. (N.d.-d). ResearchGate. [cited 2024 Nov 5]. Available from: https://www.researchgate.net/profile/Anfo-Pub-2/publication/385012327_AI-based_criminal_identification_system_A_literature_review/links/6711f8db68ac3041499e2f78/AI-based-criminal-identification-system-A-literature-review.pdf
8. (N.d.-e). ResearchGate. [cited 2024 Nov 5]. Available from: https://www.researchgate.net/profile/Piyush-Choudhary-8/publication/384966076_Comprehensive_Review_and_Comparative_Analysis_of_Yoga_Pose_Estimation_Techniques_Advancements_in_Computer_Vision_for_Enhanced_Yoga_Practice/links/670fa00fd796f96b8ebbff37/Comprehensive-Review-and-Comparative-Analysis-of-Yoga-Pose-Estimation-Techniques-Advancements-in-Computer-Vision-for-Enhanced-Yoga-Practice.pdf
9. (N.d.-f). ResearchGate. [cited 2024 Nov 5]. Available from: https://www.researchgate.net/profile/Piyush-Choudhary-8/publication/384966626_DESIGN_DEVELOPMENT_OF_ADVANCED_DIGITAL_COMMUNICATION_SYSTEM_USING_LATEST_TOOLS_TECHNOLOGIES/links/670fa10009ba2d0c76051134/DESIGN-DEVELOPMENT-OF-ADVANCED-DIGITAL-COMMUNICATION-SYSTEM-USING-LATEST-TOOLS-TECHNOLOGIES.pdf
10. Choudhary P, *et al*. Innovations in Automated Food Production: The Case of Sensory Preservation in Pickles. Food Science and Technology. 2020;5(2):76–90.
11. Choudhary P, *et al*. Innovations in Automated Food Production: The Case of Sensory Preservation in Pickles. Food Science and Technology. 2020;5(2):76–90.
12. Choudhary P, Shivam JJ, Sendre J, Parauha K. Innovations in Automated Food Production: The Case of Sensory Preservation in Pickles. Food Science and Technology. 2020;5(2):76–90.
13. Singh C, *et al*. ML Based Strategy for Optimal Power Prediction in IOT. Solid State Technology. 2020;63(6):8138–50.
14. Singh C, Chauhan MD, Vatti RA, Choudhary P, Neeraja MB. ML Based Strategy for Optimal Power Prediction in IOT. Solid State Technology. 2020;63(6):8138–50.
15. Singh C, *et al*. ML Based Strategy for Optimal Power Prediction in IOT. Solid State Technology. 2020;63(6):8138–50.
16. Singh C, Chauhan MD, Vatti RA, Choudhary P, Neeraja MB. ML Based Strategy for Optimal Power Prediction in IOT. Solid State Technology. 2020;63(6):8138–50.
17. Singh C, *et al*. ML Based Strategy for Optimal Power Prediction in IOT. Solid State Technology. 2020;63(6):8138–50.
18. Singh C, Chauhan MD, Vatti RA, Choudhary P, Neeraja MB. ML Based Strategy for Optimal Power Prediction in IOT. Solid State Technology. 2020;63(6):8138–50.
19. Choudhary P, *et al*. IntelliLearn: AI Powered Education Hub. 2024.
20. Choudhary P, Dubey R, Jain P, Singh S, Lalwani S, Kaushal M. IntelliLearn: AI Powered Education Hub. 2024.
21. Choudhary P, Dubey R, Jain P, Singh S, Lalwani S, Kaushal M. IntelliLearn: AI Powered Education Hub. 2024.
22. Choudhary P, Dubey R, Jain P, Singh S, Lalwani S, Kaushal M. IntelliLearn: AI Powered Education Hub. 2024.
23. Choudhary P, Dubey R, Jain P, Singh S, Lalwani S, Kaushal M. IntelliLearn: AI Powered Education Hub. 2024.
24. Choudhary P, *et al*. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
25. Choudhary P, Singh D, Patel S, Jain S, Soni S, Sharma V. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
26. Choudhary P, Singh D, Patel S, Jain S, Soni S, Sharma V. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
27. Choudhary P, Singh D, Patel S, Jain S, Soni S, Sharma V. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
28. Choudhary P, Singh D, Patel S, Jain S, Soni S, Sharma V. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
29. Choudhary P, *et al*. Yoga Pose Detection and Feedback Generation: A Review. 2023.
30. Choudhary P, Kumar A, Raja A, Sharma A, Jain K. Yoga Pose Detection and Feedback Generation: A Review. 2023.
31. Choudhary P, Kumar A, Raja A, Sharma A, Jain K. Yoga pose detection and feedback generation: A review. 2023.
32. Choudhary P, Kumar A, Raja A, Sharma A, Jain K. Yoga pose detection and feedback generation: A review. 2023.
33. Choudhary P, Kumar A, Raja A, Sharma A, Jain K. Yoga pose detection and feedback generation: A review. 2023.
34. Choudhary P, *et al*. Gesture Driven Gaming: A Deep Dive into Computer Vision-Based Hand Gesture Recognition. 2024.
35. Choudhary P, Shinde B, Yadav A, Kumayu A, Parmar

A, Upadhyay A, Joshi A. Gesture Driven Gaming: A Deep Dive into Computer Vision-Based Hand Gesture Recognition. 2024.
36. Choudhary P, Shinde B, Yadav A, Kumayu A, Parmar A, Upadhyay A, Joshi A. Gesture Driven Gaming: A Deep Dive into Computer Vision-Based Hand Gesture Recognition. 2024.
37. Choudhary P, Shinde B, Yadav A, Kumayu A, Parmar A, Upadhyay A, Joshi A. Gesture Driven Gaming: A Deep Dive into Computer Vision-Based Hand Gesture Recognition. 2024.
38. Choudhary P, Shinde B, Yadav A, Kumayu A, Parmar A, Upadhyay A, Joshi A. Gesture Driven Gaming: A Deep Dive into Computer Vision-Based Hand Gesture Recognition. 2024.
39. Choudhary P, *et al*. IntelliLearn: AI Powered Education Hub. 2024.
40. Choudhary P, Dubey R, Jain P, Singh S, Lalwani S, Kaushal M. IntelliLearn: AI Powered Education Hub. 2024.
41. Choudhary P, Dubey R, Jain P, Singh S, Lalwani S, Kaushal M. IntelliLearn: AI Powered Education Hub. 2024.
42. Choudhary P, Dubey R, Jain P, Singh S, Lalwani S, Kaushal M. IntelliLearn: AI Powered Education Hub. 2024.
43. Choudhary P, Dubey R, Jain P, Singh S, Lalwani S, Kaushal M. IntelliLearn: AI Powered Education Hub. 2024.
44. Choudhary P, *et al*. AI-based criminal identification system: A literature review. 2022.
45. Choudhary P, Singh M, Sankhere S, Vishwakarma S, Sanap S. AI-based criminal identification system: A literature review. 2022.
46. Choudhary P, Singh M, Sankhere S, Vishwakarma S, Sanap S. AI-based criminal identification system: A literature review. 2022.
47. Choudhary P, Singh M, Sankhere S, Vishwakarma S, Sanap S. AI-based criminal identification system: A literature review. 2022.
48. Choudhary P, Singh M, Sankhere S, Vishwakarma S, Sanap S. AI-based criminal identification system: A literature review. 2022.
49. Choudhary P, *et al*. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
50. Choudhary P, Singh D, Patel S, Jain S, Soni S, Sharma V. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
51. Choudhary P, Singh D, Patel S, Jain S, Soni S, Sharma V. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
52. Choudhary P, Singh D, Patel S, Jain S, Soni S, Sharma V. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
53. Choudhary P, Singh D, Patel S, Jain S, Soni S, Sharma V. Augmented Reality Books in Pre-Primary Education: Fostering Social and Emotional Learning for Young Learners.
54. Hossain AFMG, Rahman MM, Chowdhury MAH. Plant Disease Detection and Classification Using Machine Learning Algorithms. J Comput Commun. 2021;9(2):15-25.
55. Agrawal DP, Ferreira ARC, Silva JH. Deep Learning Approaches for Plant Disease Classification: A Review. Agric Syst. 2020;179:102-114.
56. Wu YT, Zhang HY, Liu JC. Leaf Blight Detection Using Convolutional Neural Networks. Comput Electron Agric. 2018;147:65-73.
57. Singh SBJ, Ghosh MK, Kumar VV. Computer Vision-Based Methods for Leaf Disease Detection: A Review. J Agric Inform. 2020;11(3):49-56.
58. Sharma RKB, Lee MM, Wang CH. OpenCV: Computer Vision Library Overview. Proc Int Conf Comput Vis. 2021;2394-2401.
59. Patterson MG, Johnson TSR, Han LC. TensorFlow and Keras for Deep Learning: An Overview. Mach Learn J. 2021;112(2):234-245.
60. Adams JKR, Roberts SK, Clarke LA. Responsive Web Design: A Comprehensive Review. Web Des Dev J. 2019;8(1):67-80.
61. Stevens CE. Tailwind CSS: A Utility-First Approach to Web Design. CSS Des Rev. 2020;5(2):89-102.