Project Report

Plant Disease Detection Using Convolutional Neural Networks(Multiclass Classification)

Name: Suhani Tomar

Enrollment_No: 22C21123

Course Name: Computer Vision

Course Code: C2610C2

Instructor Name: Dr. Pradeep Laxkar

Submission Date: 19 April, 2025

1.1 Introduction

Computer vision is a powerful field in artificial intelligence that enables machines to "see" and understand images and videos just like humans. It is widely used in many areas such as self-driving cars, facial recognition, medical imaging, and agriculture.

One of the major problems faced by farmers is the early identification of plant diseases. If diseases are not detected on time, they can destroy entire crops, leading to major losses. Manual checking of plant leaves is time-consuming and not always accurate, especially when done on a large scale.

The motivation behind choosing this project is to help farmers and agricultural experts identify plant diseases quickly and accurately using deep learning. By using computer vision techniques, especially Convolutional Neural Networks (CNNs), we can build a model that automatically detects plant leaf diseases from images.

The overall goal of this project is to create a smart and easy-to-use system that can predict plant diseases by analyzing images of leaves. This will support farmers in taking fast and correct actions to protect their crops.

1.2 Problem Statement

The problem we are addressing is the automatic detection and classification of plant leaf diseases using images. Farmers often struggle to detect and identify plant diseases accurately without expert assistance, especially in large-scale farming operations. Traditional methods of disease identification are not only time-consuming but also require a significant amount of expertise, making them less practical for large farms where early and accurate detection is crucial for preventing widespread damage.

One of the key challenges in this problem is that many plant diseases have symptoms that look strikingly similar to one another, making visual differentiation difficult. Additionally, environmental factors such as varying lighting conditions, weather, and the quality of images taken with different devices can significantly affect the accuracy of disease detection. These factors pose further difficulties for ensuring that a model can reliably detect and classify plant diseases from real-world images, which are often taken under suboptimal conditions.

This problem is critical because plant diseases, if left unchecked, can severely harm crop yields, reduce food production, and result in significant financial losses for farmers. Early detection of plant diseases can help in implementing timely control measures, which is vital for minimizing the impact on crop quality and quantity. The aim of this project is to address these issues and improve the efficiency of disease detection in agriculture.

The objectives of the project are as follows: First, we aim to train a deep learning model, specifically a Convolutional Neural Network (CNN), to classify plant leaf diseases with high accuracy. Next, we intend to develop a user-friendly system that allows farmers or users to upload images of plant leaves and receive real-time predictions of the diseases affecting them. Additionally, the system will provide treatment suggestions in multiple languages to ensure accessibility for farmers across different regions. Finally, the application will include features like text-to-speech, which will read out the disease diagnosis and treatment suggestions, and a confidence level display, offering users transparency and trust in the predictions made by the model.

Methodology

2.1 Dataset

For this project, we utilized the PlantVillage dataset, which is publicly available on Kaggle and widely recognized for plant disease classification tasks. The dataset consists of over 54,000 images of both healthy and diseased plant leaves. It includes 38 different classes, representing various plant diseases and healthy conditions of crops such as tomatoes, apples, grapes, potatoes, corn, and more. The images are provided in JPEG format, with varying sizes, but all images were resized to 224x224 pixels to ensure uniformity for input into the deep learning model.

To prepare the dataset for training, we applied several preprocessing steps. First, all images were resized to 224x224 pixels to ensure consistency across the dataset. We then normalized the pixel values by scaling them between 0 and 1, which helps stabilize the training process. Data augmentation techniques were also used to increase the model's ability to generalize. These included random rotation, zoom, and horizontal flipping of the images. These transformations allowed the model to learn more robust features and avoid overfitting by introducing variety into the training data. Finally, the dataset was split into a training set and a validation set, with 80% of the data used for training and 20% for validation. This split ensured that the model was trained on a large portion of the data while having a separate set for evaluation, which is crucial for monitoring performance and preventing overfitting.

2.2 Approach/Algorithm

For this project, we used a Convolutional Neural Network (CNN), which is a deep learning algorithm particularly effective for image classification tasks. The high-level steps followed in the process are straightforward. First, the input image is preprocessed by resizing and normalizing it. The CNN model then extracts features from the image through convolution layers, which help in detecting patterns and important characteristics. These features are passed through pooling layers to reduce the size and computational complexity of the data. Finally, the model outputs a prediction of the disease class, along with a confidence score, indicating the likelihood of the predicted class.

The CNN architecture used in this project consists of several layers. The input layer accepts a 224x224 RGB image, which is processed through three convolutional layers. The first convolutional layer has 32 filters with a 3x3 kernel and uses ReLU activation. The first max pooling layer follows, with a 2x2 pool size to downsample the image. The second convolutional layer uses 64 filters with a 3x3 kernel and is followed by another max pooling layer. A third convolutional layer with 128 filters and a 3x3 kernel comes next, followed by another max pooling layer. After the convolutional and pooling layers, the model flattens the 2D features into a 1D array using the flatten layer. This is followed by a dense layer with 256 neurons and ReLU activation, and the final output layer uses softmax activation to predict one of the 38 disease classes.

The reason we chose CNN for this project is that it automatically learns relevant features from images without requiring manual feature extraction. CNNs outperform traditional machine learning algorithms on image data, as they are designed to detect hierarchical patterns, which is particularly useful for complex image datasets. Furthermore, CNNs are both efficient and accurate when handling large datasets, making them ideal for this plant disease classification task.

To implement and train the model, we used TensorFlow and Keras, popular libraries for building deep learning models. For visualization purposes, we used Matplotlib to plot the accuracy and loss graphs. PIL (Python Imaging Library) was used for image processing tasks, while NumPy handled numerical operations. For the web interface, Streamlit was used to create a simple, user-friendly platform. Additionally, Googletrans and pyttsx3 were incorporated for language translation and text-to-speech features, allowing the system to provide treatment suggestions in multiple languages and audibly read the results.

3.1 Experimental Setup

The project was developed and tested using the following hardware and software setup. The hardware used for the project consisted of a laptop with an Intel Core i5 processor, 8 GB of RAM, and integrated GPU (no external GPU). The storage capacity of the laptop was 256 GB SSD, which provided sufficient space for handling the dataset and project files. In terms of software, the development environment was set up on Windows 11, and Google Colab was used for model training due to its cloud-based resources and GPU support for deep learning tasks. For deploying the model as a web application, Streamlit was used to build an intuitive and user-friendly interface. The programming language used throughout the project was Python 3.8, which is widely adopted for machine learning tasks and offers rich libraries for deep learning, image processing, and web app development.

3.2 Results

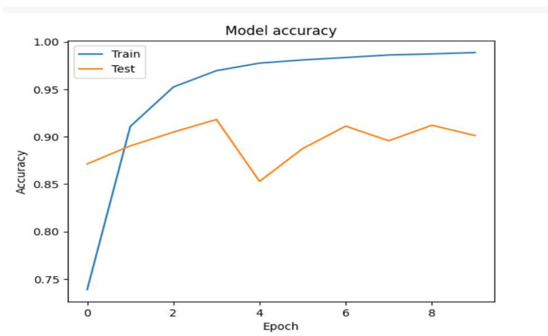
The trained Convolutional Neural Network (CNN) model was evaluated on a validation set to assess its performance. The evaluation was based on accuracy, which measures the percentage of correctly classified images. After training, the model achieved a validation accuracy of 90.12%, indicating that approximately 9 out of every 10 leaf images were correctly identified. The validation loss, which quantifies the model's error on the validation set, was found to be 0.5225, reflecting a reasonable level of performance in terms of minimizing errors during prediction. These results demonstrate that the model performs well in classifying plant leaf diseases with high accuracy.

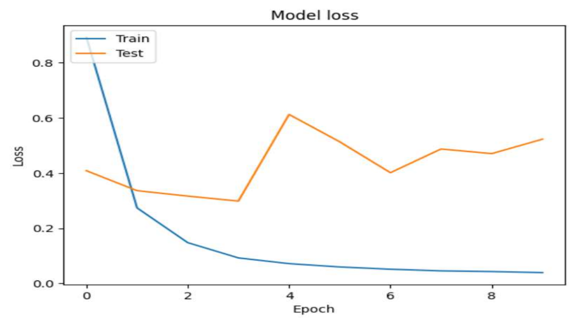
Model Evaluation Output:

```
Evaluating model..
339/339 ━━━━━━━━━━━━━━━━━━━━ 19s 57ms/step - accuracy: 0.8999 - loss: 0.5270
Validation Accuracy: 0.9011799693107605
Validation Loss: 0.5225481986999512
```

Graphical Representation:

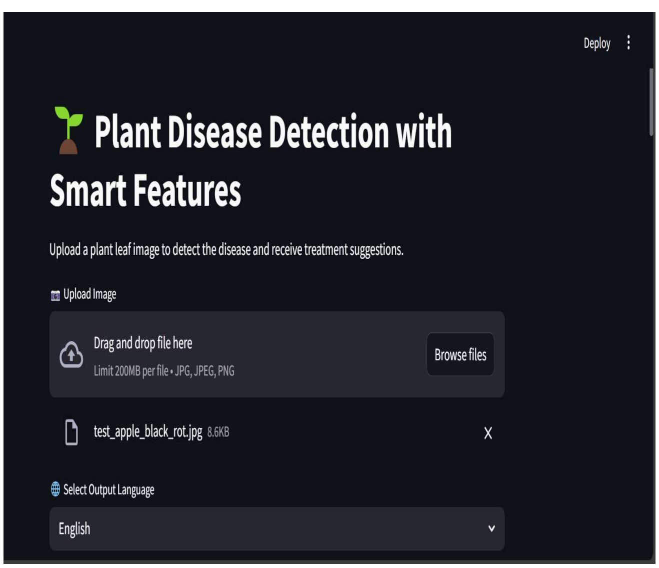
Graphs of accuracy and loss over the course of training were plotted to visualize the model's learning process and performance. These graphs help in understanding how the model improved over time and how well it generalized to the validation data.

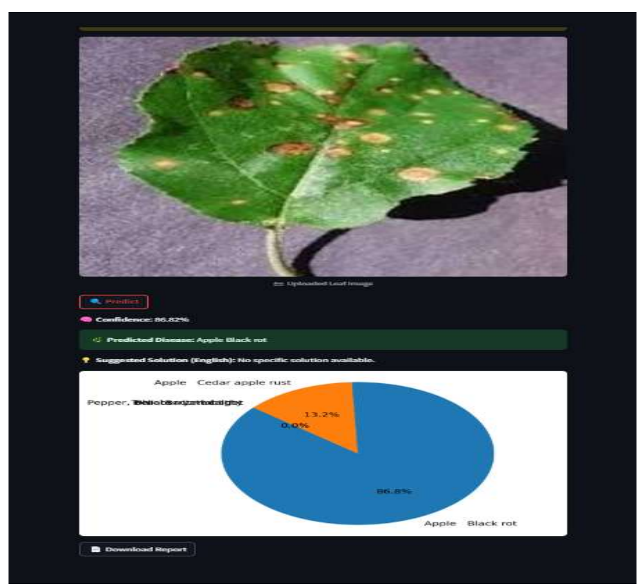




Sample Output from webApp :

Upon uploading a leaf image to the web application, the user receives an output displaying the predicted disease class along with a confidence score, providing both the diagnosis and the certainty of the prediction. This allows users to easily interpret the results and take the necessary actions based on the diagnosis.





3.3 Discussion

The results of this project indicate that the CNN model performs exceptionally well in identifying various plant diseases from leaf images. With a validation accuracy of approximately 90%, the model proves to be reliable and suitable for real-world applications such as in farms and gardens. The high accuracy suggests that the model can be effectively used for early detection of plant diseases, enabling farmers and gardeners to address problems promptly and accurately. This early detection can help reduce crop damage and improve overall productivity by facilitating quick and correct treatment.

One of the strengths of this approach is its high accuracy, which remains consistent across different plant types. The user-friendly nature of the Streamlit app, which allows users to upload images, receive translated treatment solutions, and even hear the diagnosis via text-to-speech, enhances accessibility and user experience. Additionally, visual reports like pie charts and the option to download predictions provide clear insights into the results, making it easier for users to understand and act upon the diagnosis.

However, there are certain limitations to the system. In some cases, particularly with unclear or blurry images, the model showed low confidence in its predictions, highlighting a potential weakness in handling poor-quality images. Furthermore, the model is limited to the diseases represented in the PlantVillage dataset, meaning it may not be able to detect new or rare diseases that were not included in the training data. The translation feature also depends on an internet connection, as it utilizes Google Translate, which can be a barrier in areas with limited internet access.

In comparison to traditional methods, such as rule-based approaches or manual inspections, our CNN-based model offers significant improvements in speed and accuracy. Deep learning techniques, particularly CNNs, outperform older image processing methods used in past research, providing better results for plant disease detection. Despite the success, there were challenges faced during the project. Training the model required substantial computational power and time, and integrating the model into the app, especially ensuring compatibility for both gallery and camera images, posed technical difficulties. Moreover, incorporating multi-language translation and text-to-speech support required additional effort to ensure smooth functionality.

4.1 Conclusion

In conclusion, this project successfully developed a Plant Disease Detection System using deep learning and computer vision techniques. The primary goal was to identify plant diseases by analyzing leaf images and offer users actionable treatment suggestions. By training a Convolutional Neural Network (CNN) on the PlantVillage dataset, the model achieved a commendable validation accuracy of around 90%. Additionally, a user-friendly web application was built using Streamlit, which allows users to upload leaf images, receive predictions, and access solutions in multiple languages, including a text-to-speech feature. This makes the app more accessible and useful to farmers and agricultural workers from diverse regions. The project not only met its objectives but also demonstrated how AI and computer vision can be applied to solve real-world agricultural challenges, ultimately promoting early disease detection and reducing crop loss, thus improving food production.

4.2 Future Work

Looking ahead, there are several ways to enhance the system. One improvement could be to expand the model to include more plant species and a broader range of diseases, enhancing its capability to detect additional plant health issues. Another potential development is to make the app available as a mobile application, enabling easier access for users, particularly in rural areas where smartphones are more common than computers. Further, a real-time camera detection feature could be integrated, along with GPS-based location tagging to help track disease outbreaks in specific areas. Additionally, exploring more advanced models, such as EfficientNet or Vision Transformers, could further enhance accuracy and performance. Future research could also focus on incorporating multispectral images or drone-based data for large-scale disease monitoring, allowing for more comprehensive and efficient disease detection in agricultural fields.

4.3 References

1. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). *Using deep learning for image-based plant disease detection.* Frontiers in Plant Science.

   https://doi.org/10.3389/fpls.2016.01419

2. PlantVillage Dataset - https://www.kaggle.com/datasets/emmarex/plantdisease

3. TensorFlow Documentation - https://www.tensorflow.org/

4. Streamlit Documentation - https://docs.streamlit.io/

5. Googletrans Python Library - https://pypi.org/project/googletrans/