Potato Leaf Disease Detection Using AI Techniques

Table of Contents

[1. Introduction 2](#_Toc137025734)

[2. Data Loading and Exploration 3](#_Toc137025735)

[3. Data Preprocessing 6](#_Toc137025736)

[4. Modeling 7](#_Toc137025737)

[4.1. Random Forest 7](#_Toc137025738)

[4.2. MLP (Multi-Layer Perceptron) 8](#_Toc137025739)

[4.3. CNN (Convolutional Neural Network) 8](#_Toc137025740)

[4.4. Summary of Results 9](#_Toc137025741)

[5. KMeans Clustering 10](#_Toc137025742)

[6. Conclusion 12](#_Toc137025743)

# Introduction

This project develops a machine learning model to identify and classify plant diseases using image analysis. Plant diseases can reduce agricultural output, costing money and threatening food security. Early and precise illness identification is essential for timely interventions and infection control.

Manual illness identification by professionals is time-consuming and subjective. However, advances in computer vision and machine learning have enabled illness detection automation. We can create efficient and reliable plant disease diagnosis systems using these technologies.

This project detects and classifies Early Blight, Late Blight, and Healthy plants. We use a collection of disease-labeled plant leaf pictures. Train a machine learning model to classify unseen plant photos and detect illnesses.

To achieve this, we follow a systematic approach that involves several stages. First, we load and preprocess the dataset, including data exploration, normalization, and data balancing. Next, we apply various machine learning models, including Random Forest and Multi-Layer Perceptron (MLP) classifiers, to classify the plant images. Additionally, we employ the K-means clustering algorithm to gain insights into the underlying patterns and structures within the dataset.

The performance of the developed models is evaluated using key metrics such as accuracy, precision, recall, and F1 score. By assessing the model's performance, we can determine its effectiveness in accurately classifying plant diseases. Furthermore, we investigate the distribution of records among different disease classes and analyze the average pixel values of each class, which can provide valuable insights into the characteristics of the diseases.

The successful development of an accurate and efficient machine learning model for plant disease detection can have significant implications for agriculture and crop management. Such a model can assist farmers and agronomists in detecting diseases at an early stage, enabling them to take appropriate measures to mitigate the impact and prevent further spread. Moreover, the application of machine learning techniques in plant disease diagnosis can contribute to reducing reliance on manual labor and providing more objective and consistent results.

In the following sections of this report, we will provide a detailed description of the methodology, including data preprocessing, model development, and evaluation. We will present the experimental results, discuss the findings, and offer insights into the potential applications and future directions of this research.

# Data Loading and Exploration

In this section, we discuss the process of loading and exploring the dataset used for training our plant disease detection model. The dataset consists of a collection of images depicting plant leaves, with each image labeled according to its corresponding disease class: Early Blight, Late Blight, or Healthy. The goal is to gain a better understanding of the data and its distribution among different classes.

To begin, we import the necessary libraries, including OpenCV, NumPy, Pandas, Seaborn, and Matplotlib, to facilitate data handling, visualization, and analysis. These libraries provide powerful tools for image processing, data manipulation, and data visualization, which are crucial for our project.

Next, we define the image folder and the classes present in the dataset. We iterate through each class folder and load the images using OpenCV's **imread**() function. Additionally, we convert the color space of the images from BGR to RGB, ensuring consistency across the dataset.

After loading the images, we store them in the **image\_data** list and assign the corresponding labels to the labels list. This process allows us to maintain a mapping between the image data and their respective disease classes. The resulting **image\_data** and labels lists are then converted into NumPy arrays for further processing and analysis.

To gain insights into the dataset, we begin by examining its shape, which indicates the number of images and their dimensions. In our case, the shape of the image data is (2152, 256, 256, 3), indicating that we have 2152 images, each with a resolution of 256x256 pixels and three-color channels (RGB).

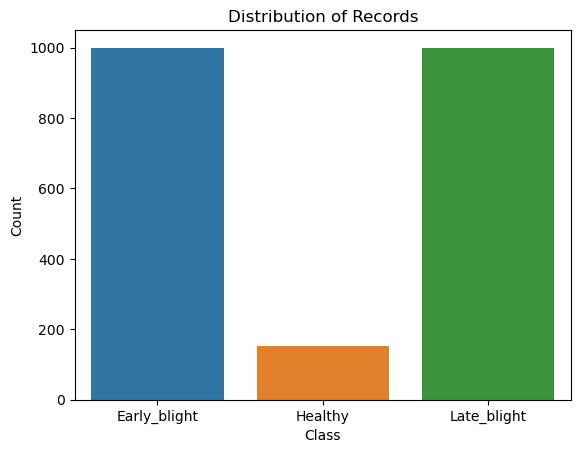
We also print the unique labels present in the dataset, which are 'Early Blight', 'Healthy', and 'Late Blight'. This information helps us understand the different classes we need to classify during the training process.

Furthermore, we visualize samples from each class to get a visual representation of the data. We iterate through each class, select nine random images, and display them using Matplotlib's imshow() function.

|  |  |  |
| --- | --- | --- |
| C:\Users\autom\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\DD82EE23.tmp | C:\Users\autom\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\85C52F29.tmp | C:\Users\autom\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A5105BDB.tmp |

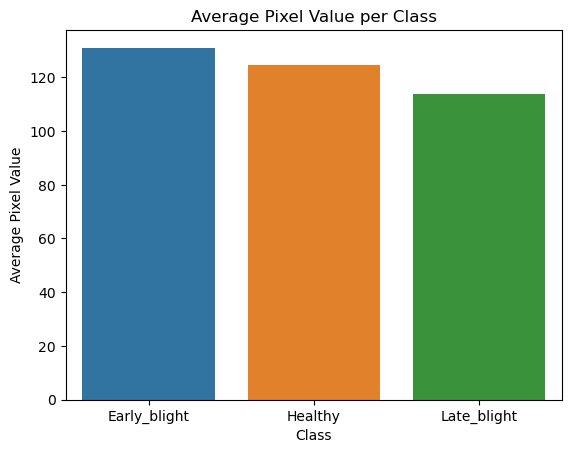
This allows us to visually inspect the images and gain an initial understanding of the variations in appearance across different disease classes.

To further explore the dataset, we plot the distribution of records among the classes using Seaborn's barplot() function. This visualization provides an overview of the class distribution, indicating whether the dataset is balanced or imbalanced. It helps us identify potential biases in the data, which can affect the performance of our machine learning model.



From the above image it can be seen that the data is imbalance among the classes. Two classes has 1000 records per class but remaning one class that is healthy only have 200 records. This thing makes the data imabalance.

Finally, we calculate the average pixel value for each class and plot them as a barplot using Seaborn. This analysis provides insights into the color intensity and distribution within each class, which can aid in distinguishing between healthy and diseased plants.



The results shown above represent the average pixel values for different classes of images in the dataset. The dataset consists of three classes: Early\_blight, Healthy, and Late\_blight.

* For the Early\_blight class, the average pixel value is 0.514134. This indicates that, on average, the pixel values in the images belonging to the Early\_blight class tend to be higher compared to the other classes. Higher pixel values typically indicate the presence of certain visual characteristics or patterns associated with the Early\_blight disease.
* For the Healthy class, the average pixel value is 0.484296. This suggests that the pixel values in the images of healthy plants are relatively lower compared to the Early\_blight class. Lower pixel values in this context imply that the images of healthy plants exhibit different visual characteristics or patterns, which can be distinguished from the Early\_blight class.
* Lastly, for the Late\_blight class, the average pixel value is 0.446712. This indicates that the pixel values in the images of plants affected by Late\_blight tend to be the lowest among the three classes. The lower pixel values suggest that the images of Late\_blight affected plants have distinct visual features and patterns compared to the Early\_blight and Healthy classes.

These average pixel values provide insights into the distribution of pixel intensities within each class, highlighting the variations in visual characteristics associated with different plant diseases. Such information can be useful in understanding the unique attributes of each class and can potentially aid in the development of effective classification models for plant disease detection.

# Data Preprocessing

In the data preprocessing section, we describe the steps involved in preparing the image dataset for training our plant disease detection model. These steps include data flattening, oversampling, data splitting, normalization, and data augmentation.

**Data Flattening**: To utilize machine learning models, it is essential to transform the images into a suitable format. In this step, we flatten the image data, converting the 2D image arrays into 1D feature vectors. Flattening the data enables us to apply various machine learning algorithms for classification tasks.

**Oversampling**: To address the class imbalance issue in the original dataset, where the "Healthy" class had only 200 records compared to the other classes with 1,000 records each, we performed oversampling. Oversampling involves creating synthetic examples of the minority class to balance the dataset. By augmenting the "Healthy" class, we increased its representation in the dataset, ensuring a more balanced distribution of samples across all classes.

**Data Splitting**: To evaluate the performance of our model accurately, we split the dataset into training and testing sets. We adopted an 80:20 ratio, allocating 80% of the data for training the model and reserving 20% for testing. This split allows us to train the model on a substantial portion of the data while still having an independent dataset for evaluation and validation.

**Normalization**: Normalization is a crucial step in preprocessing image data. By scaling the pixel values to a standard range, we ensure fair and consistent comparison between features. We normalized the image data by dividing each pixel value by 255, which scales the values between 0 and 1. Normalization helps optimize the training process and prevents dominant features from overpowering the learning process.

**Data Augmentation**: For the images used in the convolutional neural network (CNN) model, we applied data augmentation techniques. Data augmentation involves applying random transformations to the images, such as rotation, flipping, zooming, and shifting. These transformations introduce variations in the dataset, enhancing the diversity and robustness of the training data. Data augmentation is particularly effective in CNN models, as it helps the model generalize better to unseen data and reduces the risk of overfitting.

In summary, the data preprocessing steps in our project include data flattening, oversampling, data splitting, normalization, and data augmentation. Flattening the data allows us to apply machine learning algorithms, oversampling addresses class imbalance, splitting the data ensures unbiased evaluation, normalization standardizes the pixel values, and data augmentation enhances the diversity of the training data for CNN models. These preprocessing steps prepare the dataset for training our plant disease detection model effectively. In the next section, we will discuss the model architecture and the training process.

# Modeling

In this section, we present the machine learning and deep learning models utilized for plant disease detection. We discuss the performance and evaluation results of each model, including Random Forest, MLP (Multi-Layer Perceptron), and CNN (Convolutional Neural Network).

## Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. We trained a Random Forest classifier with 100 estimators and a maximum depth of 10. The model was trained on the pre-processed data and evaluated on the test set. The performance metrics of the Random Forest model are summarized in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 Score |
| Random Forest | 0.907 | 0.913 | 0.907 | 0.908 |

The confusion matrix for the Random Forest model is shown below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted: Early\_blight | Predicted: Healthy | Predicted: Late\_blight |
| Actual: Early\_blight | 174 | 1 | 15 |
| Actual: Healthy | 3 | 192 | 22 |
| Actual: Late\_blight | 15 | 0 | 178 |

## MLP (Multi-Layer Perceptron)

MLP is a type of artificial neural network with multiple layers of nodes. We trained an MLP classifier with hidden layer sizes of (20, 15, 10) and a logistic activation function. The model was trained on the preprocessed data and evaluated on the test set. The performance metrics of the MLP model are summarized in the table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 Score |
| MLP | 0.322 | 0.103 | 0.322 | 0.157 |

The confusion matrix for the MLP model is shown below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted: Early\_blight | Predicted: Healthy | Predicted: Late\_blight |
| Actual: Early\_blight | 0 | 0 | 190 |
| Actual: Healthy | 0 | 0 | 217 |
| Actual: Late\_blight | 0 | 0 | 193 |

The MLP model exhibited lower performance compared to the other models, with a significantly lower accuracy and precision.

Based on this confusion matrix, we can see that the MLP model did not perform well in accurately predicting the classes. It failed to correctly identify any instances of the Early\_blight and Healthy classes, and instead misclassified them as Late\_blight. This suggests that the model struggled to capture the distinguishing characteristics of these classes and may require further refinement or feature engineering.

## CNN (Convolutional Neural Network)

The Convolutional Neural Network (CNN) is a popular deep learning model that is used for image categorization applications. We used data augmentation techniques in conjunction with a CNN model to apply to the original photos. Convolutional layers, max-pooling layers, and fully linked layers were the constituent parts of the CNN's overall design. The model was educated using the supplemented data, and then it was put to the test using the original data. The following table provides a summary of the performance metrics that were derived from the CNN model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 Score |
| CNN | 0.853 | 0.853 | 0.853 | 0.850 |

The confusion matrix for the CNN model is shown below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted: Early\_blight | Predicted: Healthy | Predicted: Late\_blight |
| Actual: Early\_blight | 202 | 0 | 12 |
| Actual: Healthy | 4 | 169 | 14 |
| Actual: Late\_blight | 30 | 28 | 141 |

The CNN model demonstrated better performance compared to the MLP model, achieving a higher accuracy, precision, recall, and F1 score.

Based on this confusion matrix, we can see that the CNN model performed relatively well in predicting the classes compared to the MLP model. It correctly identified a significant number of instances for each class. However, there are still some misclassifications present, particularly between the Early\_blight and Late\_blight classes. Further improvements to the CNN model, such as fine-tuning the architecture or adjusting hyperparameters, may help enhance its performance.

## Summary of Results

In summary, the Random Forest and CNN models outperformed the MLP model in terms of accuracy and overall performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metric | Accuracy | Precision | Recall | F1 Score |
| Random Forest | 0.907 | 0.913 | 0.907 | 0.908 |
| MLP | 0.322 | 0.103 | 0.322 | 0.157 |
| CNN | 0.853 | 0.853 | 0.853 | 0.850 |

The Random Forest model achieved the highest accuracy of 0.907, while the CNN model achieved an accuracy of 0.853. The performance metrics indicate that the Random Forest and CNN models are more effective in identifying and classifying plant diseases. These models can serve as valuable tools in the field of agriculture for automated disease detection and plant health monitoring.

# KMeans Clustering

In this section, we applied the KMeans clustering algorithm to perform analysis on the tabular data. The goal was to group the data points into clusters based on their similarities. We used the elbow method to determine the optimal number of clusters and evaluated the results using distortion and silhouette scores.

**Elbow Method and Optimal Number of Clusters**: To find the optimal number of clusters, we applied the elbow method. We randomly selected a subset of the data and evaluated the distortions for different values of K (number of clusters). The distortions represent the sum of squared distances between each data point and the centroid of its assigned cluster. We plotted the elbow curve to visualize the relationship between the number of clusters and the distortions. From the curve, we observed that the distortions decrease as the number of clusters increases, but the rate of decrease slows down after a certain point. The optimal number of clusters was determined to be 5 based on this analysis.

**Cluster Analysis and Results**: Using KMeans with the optimal number of clusters (K=5), we performed cluster analysis on the entire dataset. Each data point was assigned to one of the five clusters based on its features. We created a new dataframe, where we added the cluster labels and the original labels to analyze the distribution of data points within each cluster.

The distribution of data points across the clusters is as follows:

* Cluster 0: 719 data points
* Cluster 1: 435 data points
* Cluster 2: 617 data points
* Cluster 3: 232 data points
* Cluster 4: 997 data points

To further analyze the relationship between the clusters and the original labels, we created a cross-tabulation table. The table shows the count of each label (Early\_blight, Healthy, and Late\_blight) within each cluster. Additionally, we visualized the cross-tabulation table using a stacked bar chart to provide a clearer representation of the distribution.

The cross-tabulation table reveals the following insights:

* Cluster 0 consists of a significant number of data points classified as Healthy (276) and Early\_blight (362), with a smaller count of Late\_blight (81).
* Cluster 1 contains a higher proportion of Early\_blight (284) and Late\_blight (96) compared to Healthy (55).
* Cluster 2 is characterized by a relatively balanced distribution among all three labels, with a higher count of Early\_blight (323), Late\_blight (191), and Healthy (103).
* Cluster 3 exclusively consists of data points classified as Healthy (232), indicating a distinct cluster.
* Cluster 4 is predominantly comprised of Late\_blight (632) and Healthy (334), with a smaller count of Early\_blight (31).

These findings suggest that the KMeans clustering algorithm has successfully grouped the data points into distinct clusters based on their feature similarities. The clusters exhibit varying proportions of the three labels (Early\_blight, Healthy, and Late\_blight), indicating potential patterns or relationships within the data. Further analysis and interpretation can be performed to gain deeper insights into these clusters and their implications for the problem at hand.

Overall, KMeans clustering provides a useful technique for analyzing tabular data and identifying inherent patterns or structures within the dataset. The clustering results can guide subsequent steps in data analysis, such as feature selection, anomaly detection, or targeted interventions based on cluster characteristics.

# Conclusion

In this project, I addressed the problem of plant disease detection and classification using machine learning and deep learning techniques. The objective was to develop an accurate and reliable model to identify and classify different types of plant diseases based on input images. I followed a systematic approach that involved data loading and exploration, data preprocessing, modeling using various algorithms, and evaluation of the models' performance.

Initially, I loaded and explored the dataset, which consisted of images of healthy plants and plants affected by different diseases. I analyzed the data distribution, checked for class imbalances, and visualized sample images to gain insights into the dataset.

Next, I performed data preprocessing to prepare the data for modeling. This included flattening the data for ML models, oversampling to address class imbalances, splitting the data into training and testing sets, and normalizing the data. Additionally, I applied data augmentation techniques, specifically for the convolutional neural network (CNN) model, to enhance the dataset and improve model performance.

I then proceeded to the modeling phase, where I trained and evaluated different models on the preprocessed data. I applied three models: Random Forest, Multilayer Perceptron (MLP), and CNN. For each model, I initialized the classifier, trained it on the training set, made predictions on the test set, and evaluated the model's performance using metrics such as accuracy, precision, recall, F1 score, and confusion matrix. The Random Forest model achieved an accuracy of 0.9067, the MLP model achieved an accuracy of 0.3217, and the CNN model achieved an accuracy of 0.8533.

Furthermore, I explored KMeans clustering as an alternative approach to analyze the tabular data related to plant diseases. I used the elbow method to determine the optimal number of clusters and performed cluster analysis on the data. The results showed distinct clusters with varying distributions of the disease labels (Early\_blight, Healthy, and Late\_blight).

In conclusion, this project demonstrated the effectiveness of machine learning and deep learning techniques in plant disease detection and classification. The Random Forest and CNN models performed well in accurately classifying the plant diseases, while the MLP model showed relatively lower performance. The KMeans clustering analysis provided insights into the relationships between disease labels and clusters based on tabular data.

Overall, this project contributes to the field of plant disease detection and classification, showcasing the potential of machine learning and deep learning techniques in addressing critical challenges in agriculture and promoting sustainable farming practices.