Disease Detection in Plants

1st Palen Pushkar
B.Tech IT
IIITA
Prayagraj, Uttar Pradesh
iit2017042@iiita.ac.in

2nd Ashish Kumar
B.Tech IT
IIITA
Prayagraj, Uttar Pradesh
iit2020256@iiita.ac.in

3rd Mohd Wasif
B.Tech IT
IIITA
Prayagraj, Uttar Pradesh
iit2020227@iiita.ac.in

4th Rahul
B.Tech IT
IIITA
Prayagraj, Uttar Pradesh
iit2020244@iiita.ac.in

5th Yogiraj Chaudhari
B.Tech IT
IIITA
Prayagraj, Uttar Pradesh
iit2020254@iiita.ac.in

Under the Supervision of

Dr. Manish Kumar

ABSTRACT :

Plant diseases are caused by various pathogens, including bacteria, viruses, fungi, and nematodes, which can cause significant economic losses in agriculture, horticulture, and forestry. These diseases can affect plants at any stage of their life cycle, from seedling to maturity, and can impact plant growth, yield, and quality. Symptoms of plant disease include wilting, yellowing, stunted growth, leaf spots, blight, and rot. This paper is using techniques like image processing and analysis, machine learning, and spectral analysis can be used to extract relevant features from images of leaves and classify the diseases into four classes : Cercospora_leaf_spot Gray_leaf_spot , Common_rust, healthy, and Northern_Leaf_Blight.

INTRODUCTION :

Plant diseases can have a significant impact on crop yields and quality, which can ultimately affect food security and economic stability. In recent years, there has been a growing interest in using image processing and analysis to help diagnose plant diseases. This approach has several advantages over traditional methods of diagnosis, such as visual inspection. Image processing can be used to extract quantitative features from images of leaves, which can be used to train machine learning models to identify diseases. This paper presents a new method for using image processing and analysis to diagnose plant diseases. The method was evaluated on a dataset of images of leaves infected with four different diseases: Cercospora leaf spot, gray leaf spot, common rust, and northern leaf blight. The objective of this project is to apply an ML model that can accurately detect plant diseases from images of plant leaves. The model should be able to classify the type of disease and provide recommendations for appropriate treatment. This method has the potential to be a valuable tool for farmers and other plant growers. It can be used to quickly and accurately identify plant diseases, which can help to prevent the spread of the disease and protect crops. In [1], the application of random forest algorithms to classify diseases in grapes using images obtained from uncontrolled environments gives us great insight for using the the ML models for different plant diseases.

CONTRIBUTION:

Traditional methods of detecting plant diseases are time-consuming, expensive, and require expert knowledge. However, with advances in machine learning (ML) technology, it is possible to automate the process of detecting plant diseases using image processing techniques. This project will contribute to the development of a cost-effective, efficient, and reliable tool for detecting plant diseases, which can help farmers to make informed decisions and prevent significant crop losses.

Related Works :

[1] This paper explores the application of random forest algorithms to classify diseases in grapes using images obtained from uncontrolled environments. The researchers address the challenge of identifying diseases in grapes, which can significantly impact the yield and quality of the fruit. The study focuses on developing an automated system that can accurately detect and classify diseases in grape images. They utilize random forest, a popular machine learning algorithm, to train a classification model. The model is trained on a dataset consisting of images of healthy grapes and grapes affected by various diseases. To evaluate the effectiveness of the proposed method, the researchers compare the classification results with other state-of-the-art algorithms. The experimental results demonstrate that the random forest-based approach achieves superior performance in accurately classifying grape diseases. The algorithm's ability to handle images captured in uncontrolled environments, where lighting and background conditions may vary, highlights its robustness and practical applicability.

Overall, the study highlights the potential of using random forest algorithms for disease classification in grapes, offering a promising solution to support disease management practices in viticulture and contribute to higher crop yields.

[2] This paper presents a study on the application of machine learning techniques for the detection of rice leaf diseases. The researchers address the challenge of timely and accurate identification of diseases in rice plants, which can significantly impact crop productivity. The study focuses on developing an automated system that can effectively detect and classify rice leaf diseases using machine learning algorithms. The researchers compare the performance of various machine learning techniques, including Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (KNN). They train and test the models using a dataset consisting of images of healthy rice leaves and leaves affected by different diseases. The experimental results indicate that the SVM algorithm outperforms the other techniques in terms of accuracy, precision, and recall for rice leaf disease detection. The study demonstrates the potential

of machine learning in accurately identifying and classifying rice leaf diseases, which can assist farmers in taking appropriate measures to prevent or manage these diseases effectively.
In conclusion, the paper highlights the significance of machine learning techniques in the field of agricultural disease detection, specifically focusing on rice leaf diseases. The findings of this study offer valuable insights for developing automated systems to support farmers in making informed decisions and implementing timely interventions to safeguard rice crops.

[3] This project uses Wireless Multimedia Sensor Networks (WMSN) to construct a web-enabled system for plant disease detection in agricultural applications. The researchers address the need for timely and accurate identification of plant diseases to prevent crop losses. They propose a system that utilizes WMSN to monitor and capture images of plants in real-time. The captured images are then processed and analyzed using image processing techniques to detect and classify plant diseases. The study emphasizes the advantages of WMSN, such as low-cost deployment, scalability, and wireless communication capabilities, which enable efficient data collection and transmission from multiple sensor nodes. The web-enabled aspect of the system allows farmers and agricultural experts to access the disease detection results remotely through a web interface. The experimental results demonstrate the effectiveness of the proposed system in accurately detecting and classifying plant diseases. The system offers a convenient and efficient approach to monitor plant health, enabling farmers to take appropriate actions in a timely manner.
In summary, the paper highlights the development of a web-enabled plant disease detection system using WMSN. The system provides a practical solution for monitoring and managing plant diseases in agricultural applications, facilitating informed decision-making and timely interventions for farmers to ensure healthy crop growth and maximize yields.

[4] The creation of an edge intelligent Internet of Things (IoT)-based system for rice leaf disease detection using machine learning techniques is the subject of the study presented in this paper. The necessity for an effective, automated system to identify and treat diseases in rice plants is addressed by the researchers. They propose an architecture that combines IoT devices, edge computing, and machine learning algorithms to enable real-time disease detection and monitoring of rice crops. The system collects data from IoT sensors placed in the field, including images of rice leaves, environmental parameters, and other relevant data. The collected data is processed at the edge nodes using machine learning algorithms, which are trained to classify rice leaf diseases accurately. The study emphasizes the advantages of edge intelligence, such as reduced latency, improved privacy, and reduced network traffic, enabling rapid and localized decision-making. The proposed system offers real-time disease detection, allowing farmers to take immediate action to prevent disease spread and mitigate crop losses. The experimental results demonstrate the effectiveness of the system in accurately detecting and classifying rice leaf diseases. The integration of IoT, edge computing, and machine learning provides a robust and practical solution for disease management in rice crops.
In summary, the paper presents an innovative approach to rice leaf disease detection using an IoT-based system with edge intelligence. The system's real-time capabilities, combined with machine learning algorithms, offer a valuable tool for farmers to monitor and manage diseases in rice crops, enhancing agricultural productivity and ensuring food security.

[5] This paper presents a study on the automatic classification of foliar leaf diseases using statistical and color feature extraction techniques combined with Support Vector Machine (SVM) algorithm. The researchers address the challenge of timely and accurate identification of leaf diseases, which is crucial for effective disease management in plants. They propose a method that extracts statistical and color features from leaf images to characterize different disease classes. These features are then used to train an SVM classifier for automatic disease classification.

The study highlights the importance of both statistical and color features in capturing essential information related to leaf diseases. The combination of these features enhances the accuracy and robustness of the classification model. The experimental results demonstrate the effectiveness of the proposed method in accurately classifying multiple foliar leaf diseases. The paper emphasizes the potential applications of the proposed approach in agriculture, enabling farmers and researchers to identify and classify leaf diseases efficiently. By automating the disease classification process, the method offers a valuable tool for disease management, helping to prevent crop losses and optimize plant health.
In summary, the paper presents an automatic multiclass classification method for foliar leaf diseases, combining statistical and color feature extraction with SVM. The study contributes to the development of effective and reliable disease identification techniques, providing support for improved plant health management in agricultural settings.

[6] This research presents a study on the detection and classification of plant leaf diseases using a color and texture-based approach combined with the K-Nearest Neighbors (KNN) classifier. The researchers address the need for efficient methods to identify and classify plant leaf diseases, which are critical for timely disease management. They propose a method that utilizes color and texture features extracted from leaf images to distinguish healthy leaves from those affected by diseases. These features are then used to train a KNN classifier for disease classification. The study highlights the importance of color and texture features in capturing the unique characteristics of different leaf diseases. The combination of these features enhances the accuracy of disease detection and classification. The experimental results demonstrate the effectiveness of the proposed approach in accurately identifying and classifying plant leaf diseases. The paper emphasizes the practical applicability of the proposed method in the field of agriculture, enabling farmers and researchers to quickly identify and categorize leaf diseases. By automating the disease detection process, the approach offers a valuable tool for effective disease management, aiding in the prevention of crop losses and the improvement of plant health.
In summary, the paper presents a color and texture-based approach for the detection and classification of plant leaf diseases using the KNN classifier. The study contributes to the development of accurate and efficient disease identification techniques, providing support for enhanced plant health management in agricultural contexts.

Comparison Table:

Study	**Cases**	**Method Used**	**Accuracy**		**Drawbacks**

[1]	Grapes dataset.	SVM, PNN, BPNN random forest and GLCM	86 %	Highest disease detection as compared to other models	Dimensionality reduction could improve the accuracy.
[2]	Rice dataset	KNN, decision Tree, Naive Bayes and Logistic Regression	97.90%	Decision tree perform better than the other learning models.	Low quality dataset is used.
[3]	Pomegranate, brinjal and tomato	SVM, GLCM, CS	98.4 %nhi	Various crops are taken for disease detection	Proposed method failed to follow weather conditions
[4]	Rice dataset	Decision tree, random forest, Naive Bayes, SVM, KNN, logistic regression color histogram.	97.5 %	Classification is done at the edge devices, so, it reduce the latency and connectivity over the network	Small dataset is used for classification

[5]	Paper does not specify the exact dataset used for the experiments	Uses Multi Class Support Vector Machine for Image Classification after feature acquisition	95%	Uses Multi Class Support Vector Machine for Image Classification after feature acquisition	Paper does not specify the exact dataset used for the experiments
[6]	It consists of 237 photos of diseased plant leaves. From the two largest plant disease image databases, five different categories of disease-affected photographs have been gathered.	K Nearest Neighbors for Image Classification after feature acquisition	96.76%	Applicable to multiple disease categories.	Evaluation of the proposed approach focuses on a limited set of metrics, such as accuracy, sensitivity, specificity, and F1-score, which may not provide a comprehensive evaluation of the approach.

Project Scope and Objectives :

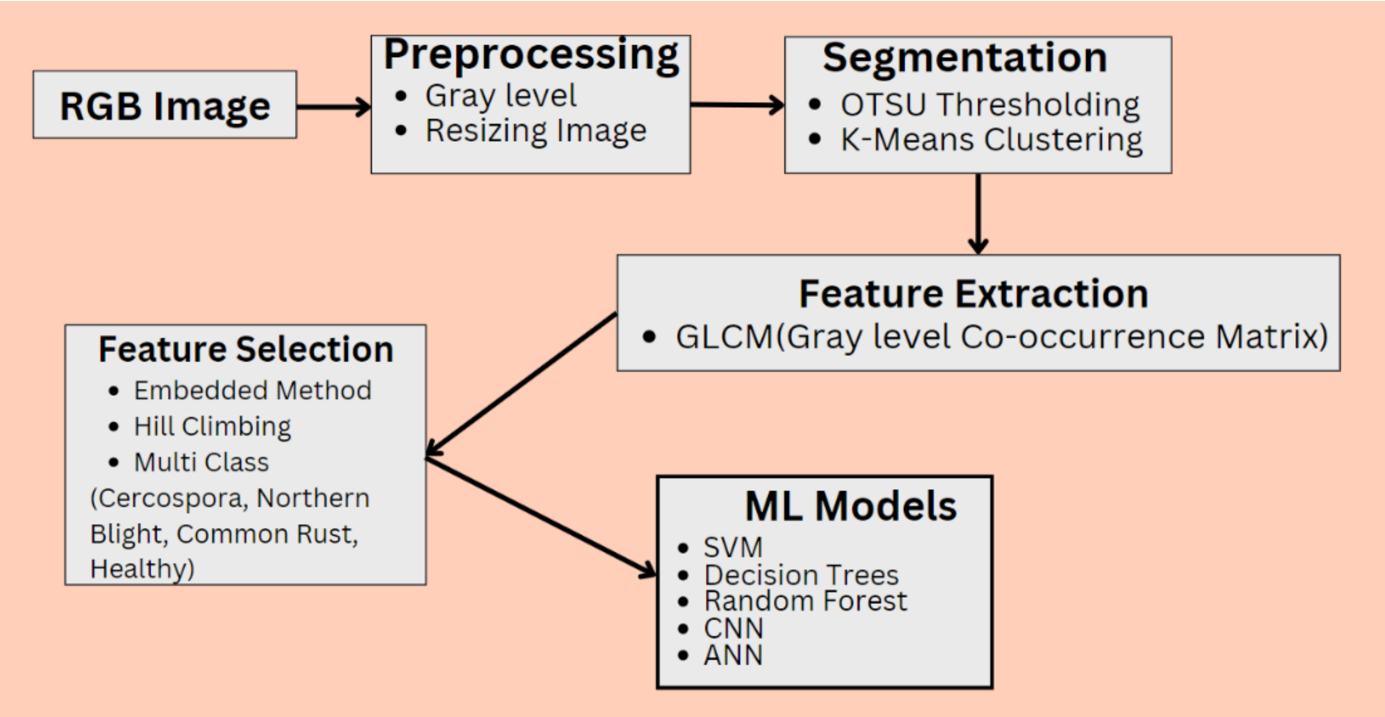
We will develop an automated system for detecting diseases in plants. The objective is to improve crop yield by detecting and diagnosing plant diseases early, allowing for timely treatment and preventing the spread of disease.

Machine learning algorithms can be trained to analyze images of plants and identify the presence of disease based on visual symptoms such as discoloration, lesions, and abnormal growth patterns. By automating the detection process, farmers and agricultural workers can save time and resources by quickly identifying diseased plants, preventing further spread of the disease, and taking the necessary steps to protect their crops.

The project would entail gathering a sizable dataset of pictures of both healthy and diseased plants, labelling the pictures to show which plants are healthy and which are diseased, and then using machine learning algorithms to train a model to recognise the presence of disease based on the visual characteristics of the plants.

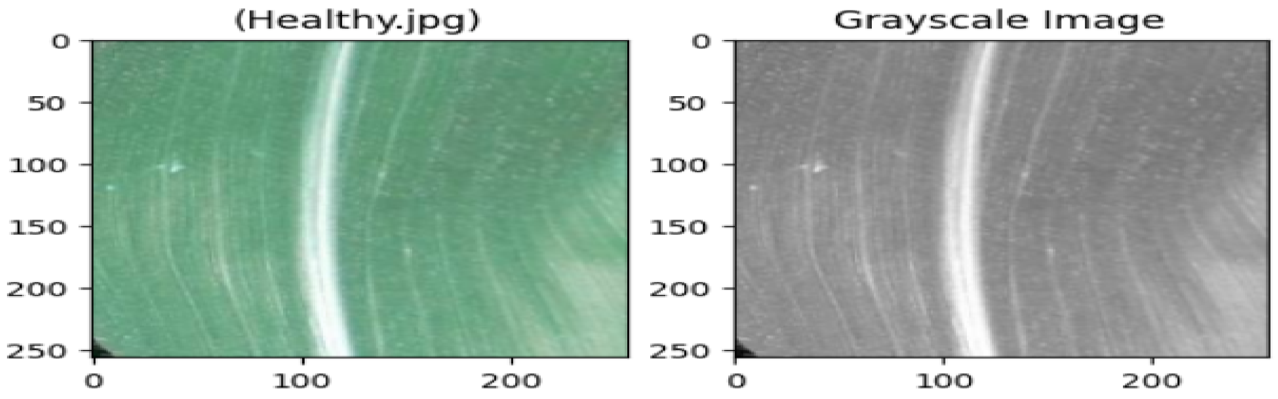
The end result would be a software application or system that can analyze images of plants and provide a diagnosis of any disease present. This system could be integrated into existing agricultural technologies, such as drones or farm equipment, to provide real-time monitoring of crops and help farmers make informed decisions about their land and resources.

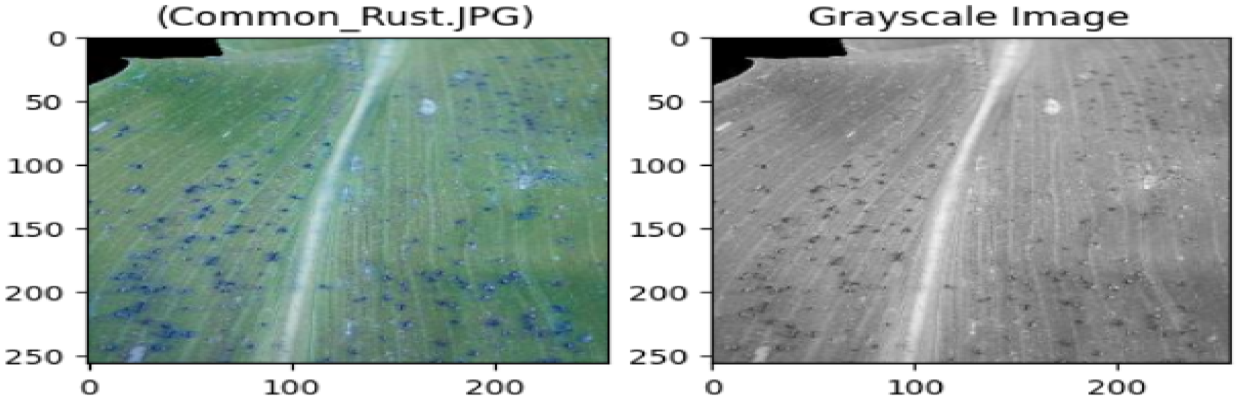
Approach / Methodology :

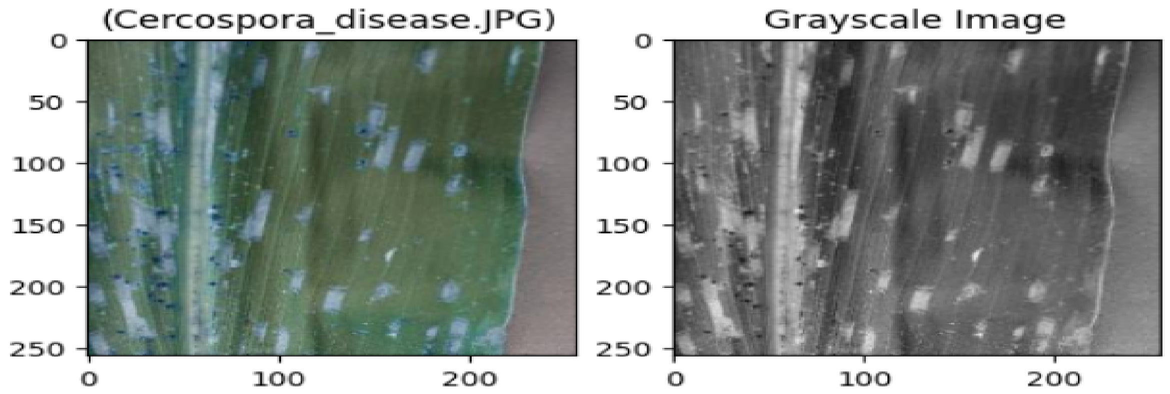


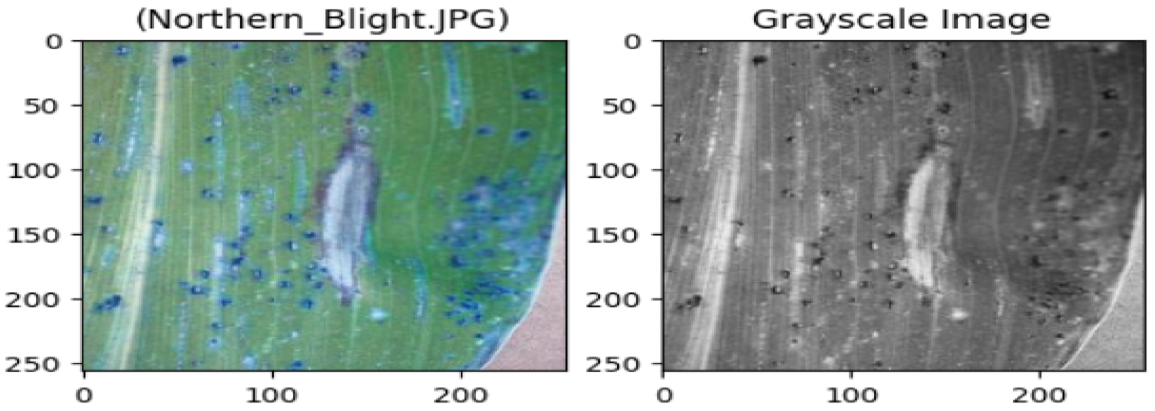
The proposed methodology for disease detection in plants using machine learning involves several steps. First, a dataset of RGB images of healthy plants and plants affected by various diseases, along with their corresponding labels, is collected. Next, the images are preprocessed by converting them to grayscale and resizing them to a standard size for ease of computation.

The below figure shows the conversion of the original image to a Grayscale Image:


(Healthy.jpg)
Grayscale Image
0
50
100
150
200
250
0
100
200

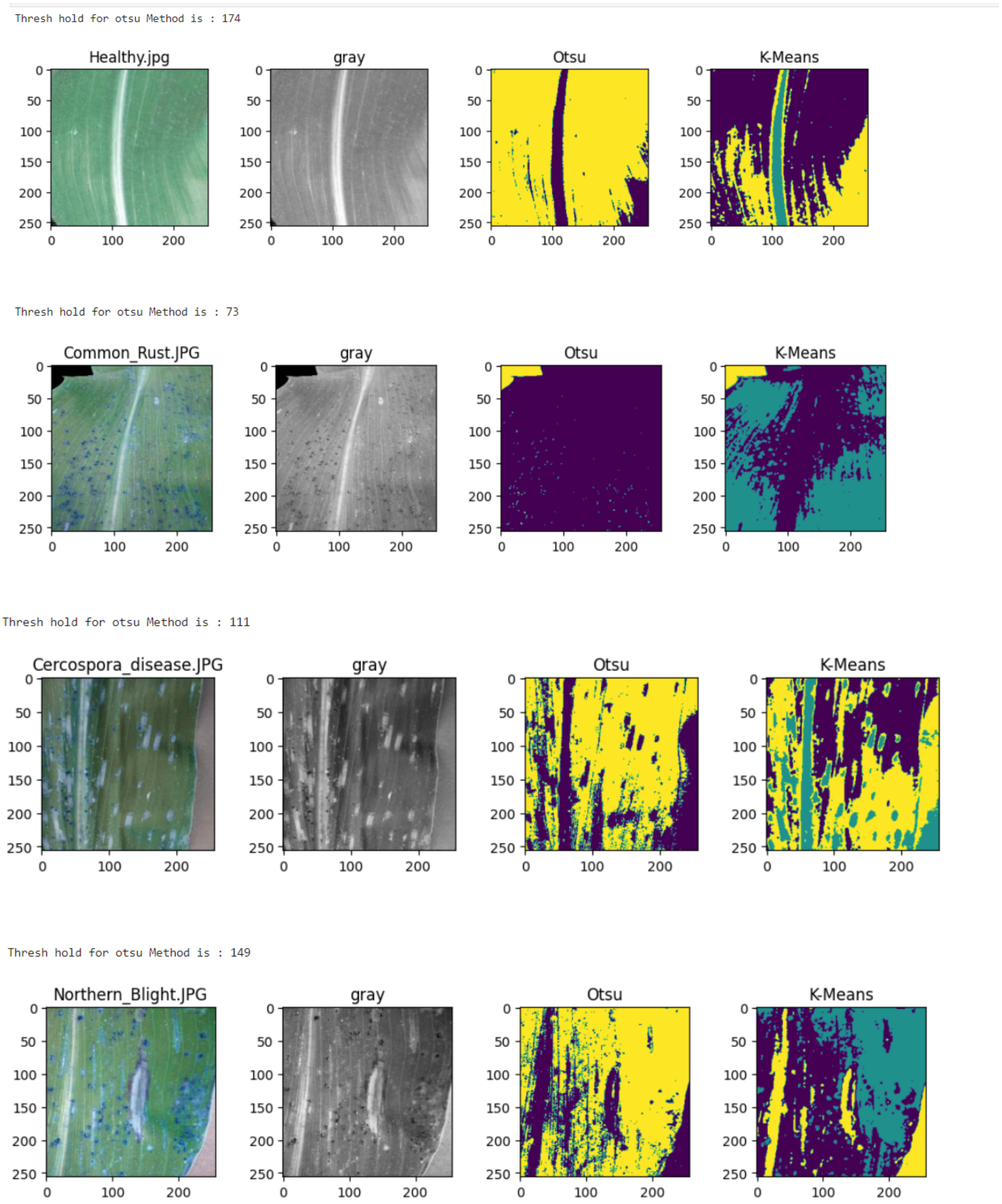

(Common_Rust.JPG)
Grayscale Image
0
50
100
150
200
250
0
100
200


(Cercospora_disease.JPG)
Grayscale Image
0
50
100
150
200
250
0
100
200


(Northern_Blight.JPG)
Grayscale Image
0
50
100
150
200
250
0
100
200

Then, image segmentation techniques such as Otsu thresholding or k-means clustering are applied to separate the plant from the background.

The below figure shows the segmentation of the Grayscale image by Otsu Thresholding and K-Means Clustering:



```
Thresh hold for otsu Method is : 174
```

```
Thresh hold for otsu Method is : 73
```

```
Thresh hold for otsu Method is : 111
```

```
Thresh hold for otsu Method is : 149
```

Equations for Segmentation Technique

Otsu Thresholding :

Let **I** be a grayscale image with **M** rows and **N** columns, and let **p_i** be the probability of gray level **i** in the image. The Otsu thresholding method finds the threshold value t that maximizes the between-class variance, which is given by the following equation:

$$\sigma_b^2(t) = \frac{[\mu_t(1-\mu_t)]^2}{\sigma_w^2(t)}$$

where **μ_t** is the mean gray level of the foreground and background regions separated by the threshold **t**, and **σ_w^2(t)** is the within-class variance of the gray levels in the two regions.

The threshold value **t*** is then selected as the value that maximizes **σ_b^2(t)**:
t* = argmax(t) {σ_b^2(t)}

Once the threshold value is obtained, the image is segmented into foreground and background regions using the following equation:

B(x,y) = {1 if I(x,y) < t*; 0 otherwise}

where **B(x,y)** is the binary image obtained after segmentation.

Following segmentation, pertinent characteristics are derived via methods like the Gray-Level Co-Occurrence Matrix (GLCM).

K-means Clustering:

Let K be the number of clusters, and let I be a grayscale image with M rows and N columns. The image is divided into K clusters using the K-means clustering technique, and each pixel is given the cluster that has the closest centroid. The sum of squared distances between each pixel and its designated centroid is what the algorithm aims to reduce.

Let $\mathbf{x_i}$ be the gray level of pixel **i** in the image, and let $\mathbf{m_i}$ be the centroid of the cluster to which pixel **i** is assigned. The K-means algorithm updates the centroids and assigns pixels to clusters iteratively using the following equations:

$$m_i = (1/n_i) * \sum_{j \in C_i} x_j, \text{ for } i = 1, 2, \ldots, K$$

$$C_i = j|||x_j - m_i||^2 \leq ||x_j - m_k||^2 \text{ for } k \neq i$$

where Ci is the collection of pixels assigned to cluster i and ni is the total number of pixels assigned to cluster i, and **||.||** denotes the Euclidean distance.

The algorithm iterates until convergence, which is typically determined by a maximum number of iterations or a small change in the sum of squared distances between iterations.

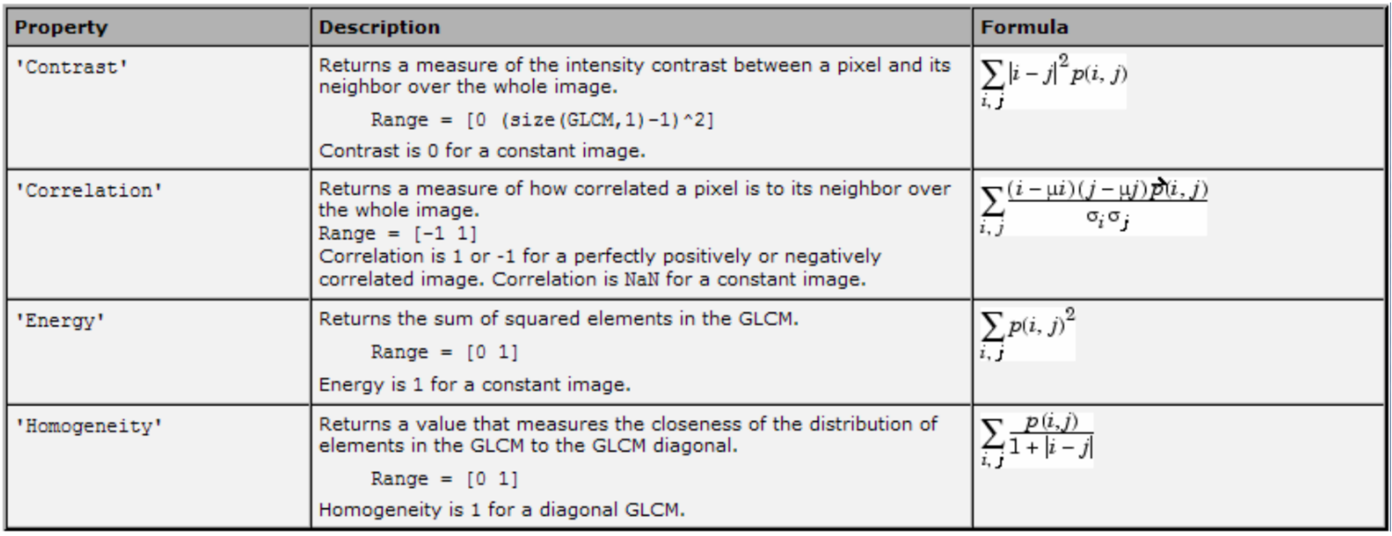
Once the algorithm converges, the image is segmented by assigning each pixel to the cluster with the closest centroid:

$$B(x, y) = argmin(i)||I(x, y) - m_i||^2, \text{ for } i = 1, 2, \ldots, K$$

where **B(x,y)** is the binary image obtained after segmentation.

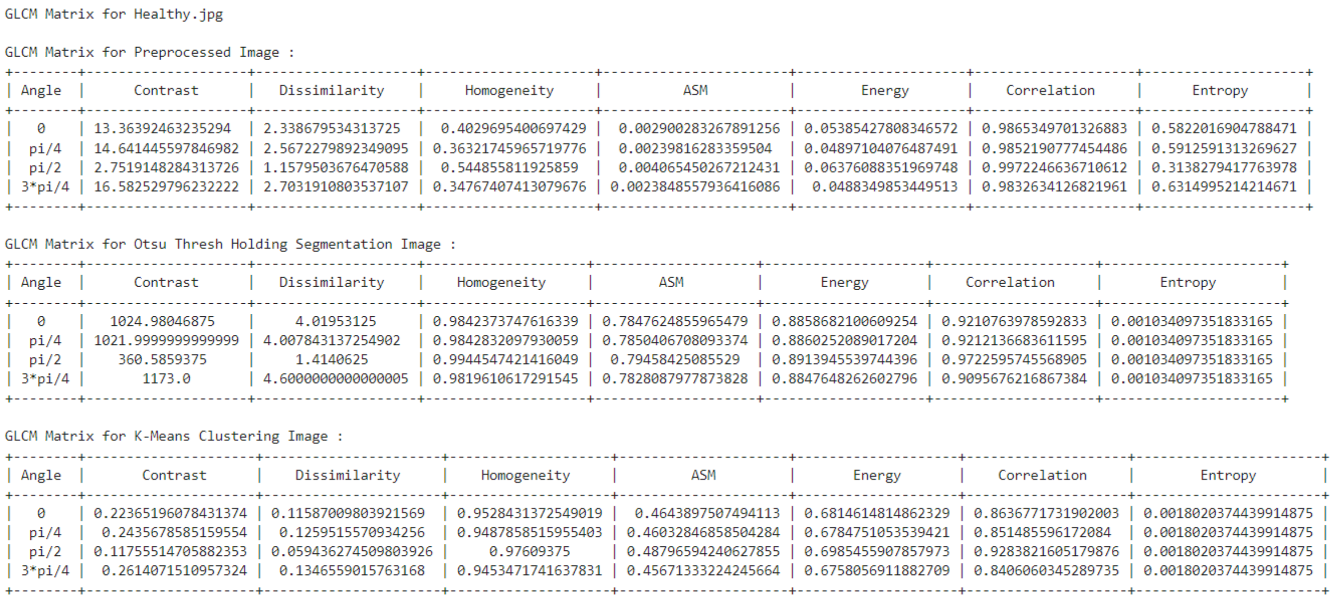
GLCM Matrices:-

The GLCM determines how often a pixel is displayed. A pixel's gray-level (grayscale intensity or level (grayscale intensity or Tone)) value i is used to calculate how often it will appear horizontally, vertically, or diagonally next to pixels with the value j.



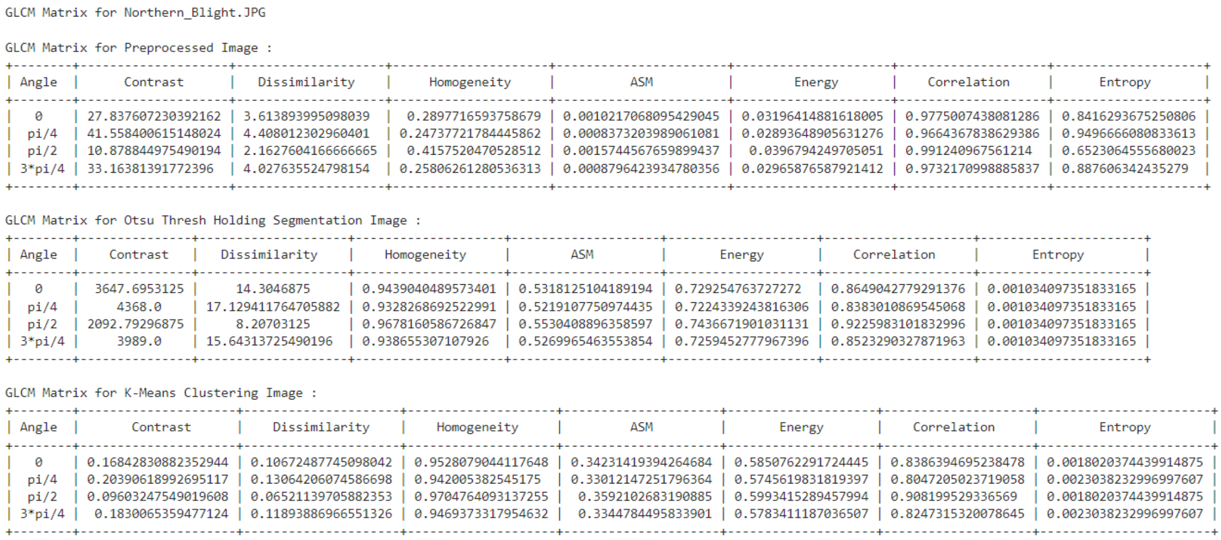
Property	Description	Formula
'Contrast'	Returns a measure of the intensity contrast between a pixel and its neighbor over the whole image. Range = [0 (size(GLCM,1)-1)^2] Contrast is 0 for a constant image.	$\sum_{i,j} \lvert i-j \rvert^2 p(i,j)$
'Correlation'	Returns a measure of how correlated a pixel is to its neighbor over the whole image. Range = [-1 1] Correlation is 1 or -1 for a perfectly positively or negatively correlated image. Correlation is NaN for a constant image.	$\sum_{i,j} \frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j}$
'Energy'	Returns the sum of squared elements in the GLCM. Range = [0 1] Energy is 1 for a constant image.	$\sum_{i,j} p(i,j)^2$
'Homogeneity'	Returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. Range = [0 1] Homogeneity is 1 for a diagonal GLCM.	$\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j \rvert}$

The below figures shows GLCM Matrix Healthy.jpg



```
GLCM Matrix for Healthy.jpg

GLCM Matrix for Preprocessed Image :
+-------+--------------------+--------------------+---------------------+-----------------------+---------------------+--------------------+--------------------+
| Angle |      Contrast      |   Dissimilarity    |     Homogeneity     |          ASM          |        Energy       |    Correlation     |      Entropy       |
+-------+--------------------+--------------------+---------------------+-----------------------+---------------------+--------------------+--------------------+
|   0   | 13.36392463235294  | 2.338679534313725  |  0.4029695400697429 |  0.002900283267891256 | 0.05385427808346572 | 0.9865349701326883 | 0.5822016904788471 |
|  pi/4 | 14.641445597846982 | 2.5672279892349095 | 0.36321745965719776 |  0.00239816283359504  | 0.04897104076487491 | 0.9852190777454486 | 0.5912591313269627 |
|  pi/2 | 2.7519148284313726 | 1.1579503676470588 |  0.544855811925859  |  0.004065450267212431 | 0.06376088351969748 | 0.9972246636710612 | 0.3138279417763978 |
| 3*pi/4| 16.582529796232222 | 2.7031910803537107 | 0.34767407413079676 | 0.0023848557936416086 |  0.0488349853449513 | 0.9832634126821961 | 0.6314995214214671 |
+-------+--------------------+--------------------+---------------------+-----------------------+---------------------+--------------------+--------------------+

GLCM Matrix for Otsu Thresh Holding Segmentation Image :
+-------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
| Angle |      Contrast      |   Dissimilarity    |    Homogeneity     |        ASM         |       Energy       |    Correlation     |       Entropy        |
+-------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
|   0   |    1024.98046875   |     4.01953125     | 0.9842373747616339 | 0.7847624855965479 | 0.8858682100609254 | 0.9210763978592833 | 0.001034097351833165 |
|  pi/4 | 1021.9999999999999 | 4.007843137254902  | 0.9842832097930059 | 0.7850406708093374 | 0.8860252089017204 | 0.9212136683611595 | 0.001034097351833165 |
|  pi/2 |    360.5859375     |     1.4140625      | 0.9944547421416049 |  0.79458425085529  | 0.8913945539744396 | 0.9722595745568905 | 0.001034097351833165 |
| 3*pi/4|       1173.0       | 4.6000000000000005 | 0.9819610617291545 | 0.7828087977873828 | 0.8847648262602796 | 0.9095676216867384 | 0.001034097351833165 |
+-------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+

GLCM Matrix for K-Means Clustering Image :
+-------+---------------------+----------------------+--------------------+---------------------+--------------------+--------------------+-----------------------+
| Angle |       Contrast      |    Dissimilarity     |    Homogeneity     |         ASM         |       Energy       |    Correlation     |        Entropy        |
+-------+---------------------+----------------------+--------------------+---------------------+--------------------+--------------------+-----------------------+
|   0   | 0.22365196078431374 | 0.11587009803921569  | 0.9528431372549019 |  0.4643897507494113 | 0.6814614814862329 | 0.8636771731902003 | 0.0018020374439914875 |
|  pi/4 |  0.2435678585159554 |  0.1259515570934256  | 0.9487858515955403 | 0.46032846858504284 | 0.6784751053539421 | 0.851485596172084  | 0.0018020374439914875 |
|  pi/2 | 0.11755514705882353 | 0.059436274509803926 |     0.97609375     | 0.48796594240627855 | 0.6985455907857973 | 0.9283821605179876 | 0.0018020374439914875 |
| 3*pi/4|  0.2614071510957324 |  0.1346559015763168  | 0.9453471741637831 | 0.45671333224245664 | 0.6758056911882709 | 0.8406060345289735 | 0.0018020374439914875 |
+-------+---------------------+----------------------+--------------------+---------------------+--------------------+--------------------+-----------------------+
```

The below figure shows the GLCM matrix for Norther_Blight.jpg

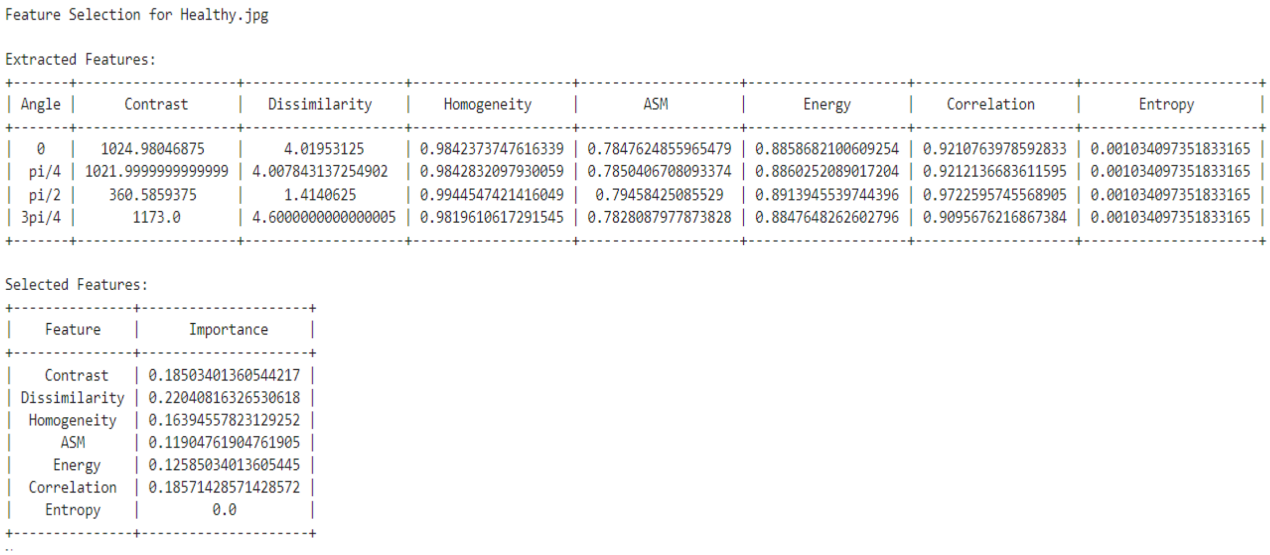


```
GLCM Matrix for Northern_Blight.JPG

GLCM Matrix for Preprocessed Image :
+-------+--------------------+--------------------+---------------------+-----------------------+---------------------+--------------------+--------------------+
| Angle |      Contrast      |   Dissimilarity    |     Homogeneity     |          ASM          |        Energy       |    Correlation     |      Entropy       |
+-------+--------------------+--------------------+---------------------+-----------------------+---------------------+--------------------+--------------------+
|   0   | 27.837607230392162 | 3.613893995098039  |  0.2897716593758679 | 0.0010217068095429045 | 0.03196414881618005 | 0.9775007438081286 | 0.8416293675250806 |
|  pi/4 | 41.558400615148024 | 4.408012302960401  | 0.24737721784445862 | 0.0008373203989061081 | 0.02893648905631276 | 0.9664367838629386 | 0.9496666080833613 |
|  pi/2 | 10.878844975490194 | 2.1627604166666665 |  0.4157520470528512 | 0.0015744567659899437 |  0.0396794249705051 | 0.991240967561214  | 0.6523064555680023 |
| 3*pi/4| 33.16381391772396  | 4.027635524798154  | 0.25806261280536313 | 0.0008796423934780356 | 0.02965876587921412 | 0.9732170998885837 | 0.887606342435279  |
+-------+--------------------+--------------------+---------------------+-----------------------+---------------------+--------------------+--------------------+

GLCM Matrix for Otsu Thresh Holding Segmentation Image :
+-------+---------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
| Angle |    Contrast   |   Dissimilarity    |    Homogeneity     |        ASM         |       Energy       |    Correlation     |       Entropy        |
+-------+---------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
|   0   |  3647.6953125 |     14.3046875     | 0.9439040489573401 | 0.5318125104189194 | 0.729254763727272  | 0.8649042779291376 | 0.001034097351833165 |
|  pi/4 |     4368.0    | 17.129411764705882 | 0.9328268692522991 | 0.5219107750974435 | 0.7224339243816306 | 0.8383010869545068 | 0.001034097351833165 |
|  pi/2 | 2092.79296875 |     8.20703125     | 0.9678160586726847 | 0.5530408896358597 | 0.7436671901031131 | 0.9225983101832996 | 0.001034097351833165 |
| 3*pi/4|     3989.0    | 15.64313725490196  | 0.938655307107926  | 0.5269965463553854 | 0.7259452777967396 | 0.8523290327871963 | 0.001034097351833165 |
+-------+---------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+

GLCM Matrix for K-Means Clustering Image :
+-------+---------------------+---------------------+--------------------+---------------------+--------------------+--------------------+-----------------------+
| Angle |       Contrast      |    Dissimilarity    |    Homogeneity     |         ASM         |       Energy       |    Correlation     |        Entropy        |
+-------+---------------------+---------------------+--------------------+---------------------+--------------------+--------------------+-----------------------+
|   0   | 0.16842830882352944 | 0.10672487745098042 | 0.9528079044117648 | 0.34231419394264684 | 0.5850762291724445 | 0.8386394695238478 | 0.0018020374439914875 |
|  pi/4 | 0.20390618992695117 | 0.13064206074586698 | 0.942005382545175  | 0.33012147251796364 | 0.5745619831819397 | 0.8047205023719058 | 0.0023038232996997607 |
|  pi/2 | 0.09603247549019608 | 0.06521139705882353 | 0.9704764093137255 |  0.3592102683190885 | 0.5993415289457994 | 0.908199529336569  | 0.0018020374439914875 |
| 3*pi/4|  0.1830065359477124 | 0.11893886966551326 | 0.9469373317954632 |  0.3344784495833901 | 0.5783411187036507 | 0.8247315320078645 | 0.0023038232996997607 |
+-------+---------------------+---------------------+--------------------+---------------------+--------------------+--------------------+-----------------------+
```

Feature Selection Using Random Forest :

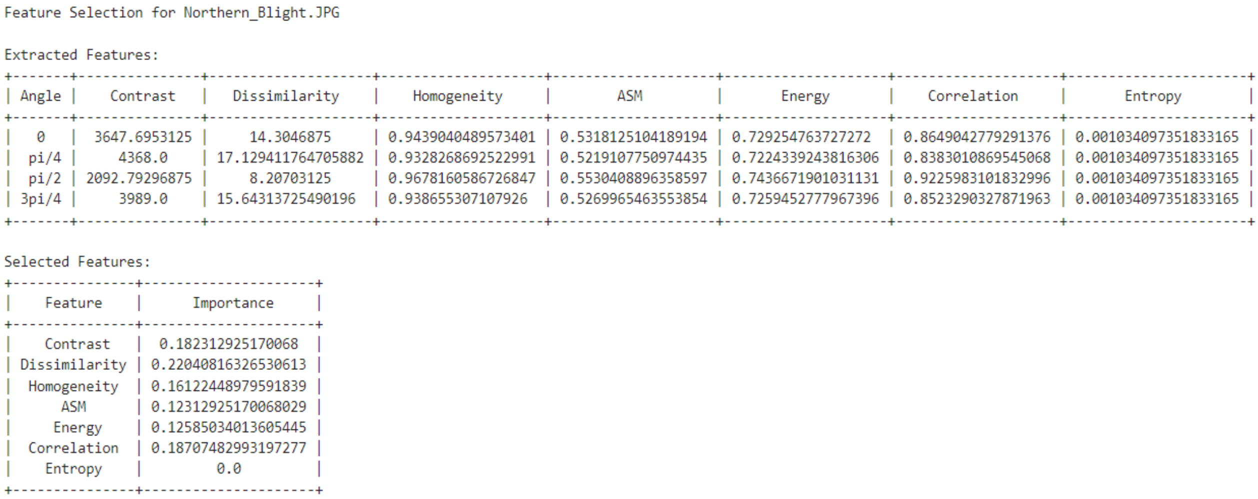
The widely used machine learning technique, It is appropriate to use Random Forest for both classification and regression tasks. One advantage of Random Forest is that it can be used for feature selection, which can help a model perform better by using fewer features during training. The importance of a feature in a random forest model is calculated based on the decrease in impurity that results from splitting the data using that feature. The impurity measure used is typically the Gini impurity or entropy.

Feature Selection for Healthy images:-



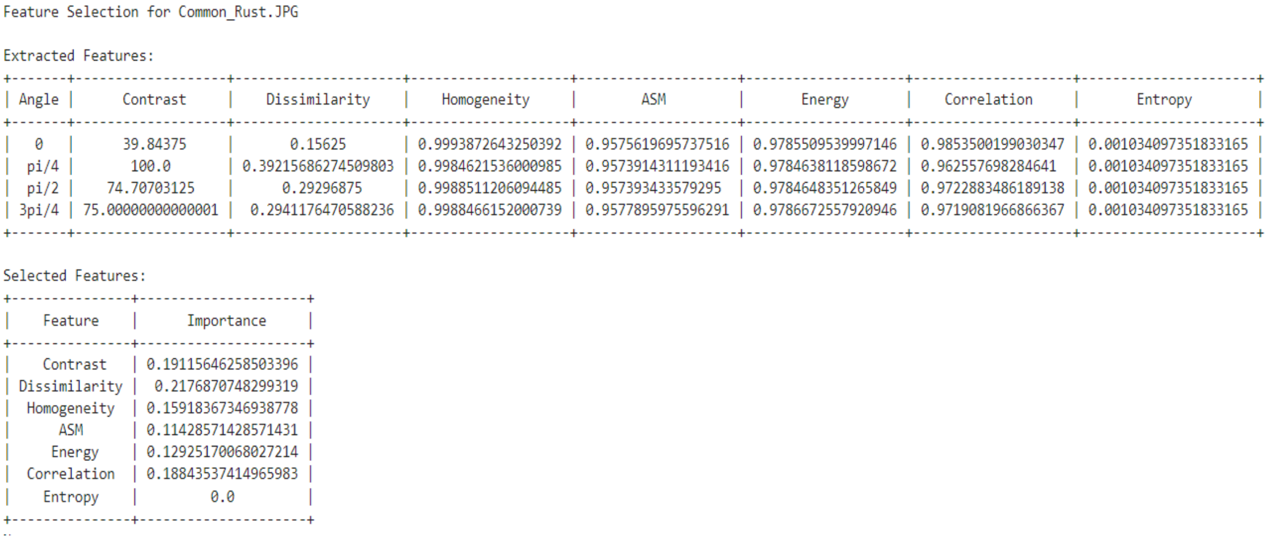
```
Feature Selection for Healthy.jpg

Extracted Features:
+-------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
| Angle |      Contrast      |   Dissimilarity    |    Homogeneity     |        ASM         |       Energy       |    Correlation     |       Entropy        |
+-------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
|   0   |   1024.98046875    |     4.01953125     | 0.9842373747616339 | 0.7847624855965479 | 0.8858682100609254 | 0.9210763978592833 | 0.001034097351833165 |
|  pi/4 | 1021.9999999999999 | 4.007843137254902  | 0.9842832097930059 | 0.7850406708093374 | 0.8860252089017204 | 0.9212136683611595 | 0.001034097351833165 |
|  pi/2 |    360.5859375     |     1.4140625      | 0.9944547421416049 |  0.79458425085529  | 0.8913945539744396 | 0.9722595745568905 | 0.001034097351833165 |
| 3pi/4 |       1173.0       | 4.6000000000000005 | 0.9819610617291545 | 0.7828087977873828 | 0.8847648262602796 | 0.9095676216867384 | 0.001034097351833165 |
+-------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+

Selected Features:
+---------------+---------------------+
|    Feature    |     Importance      |
+---------------+---------------------+
|    Contrast   | 0.18503401360544217 |
| Dissimilarity | 0.22040816326530618 |
|  Homogeneity  | 0.16394557823129252 |
|      ASM      | 0.11904761904761905 |
|     Energy    | 0.12585034013605445 |
|  Correlation  | 0.18571428571428572 |
|    Entropy    |         0.0         |
+---------------+---------------------+
```

Feature Selection for Northern_blight:-



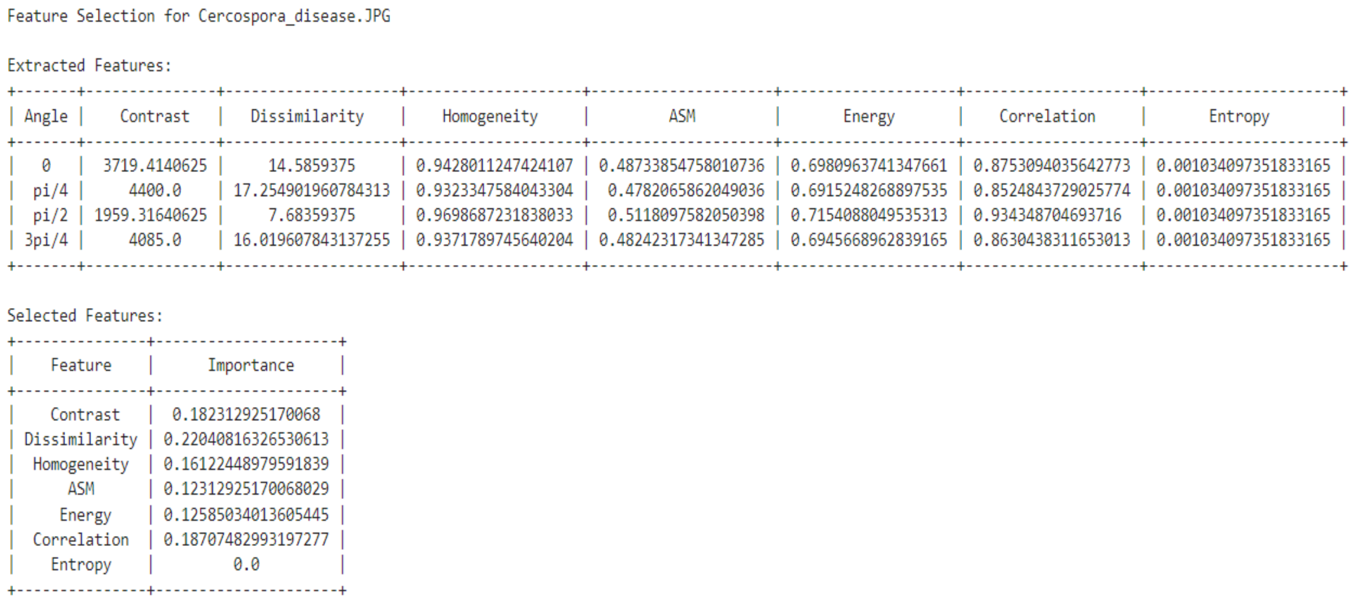
```
Feature Selection for Northern_Blight.JPG

Extracted Features:
+-------+---------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
| Angle |    Contrast   |   Dissimilarity    |    Homogeneity     |        ASM         |       Energy       |    Correlation     |       Entropy        |
+-------+---------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
|   0   |  3647.6953125 |     14.3046875     | 0.9439040489573401 | 0.5318125104189194 | 0.729254763727272  | 0.8649042779291376 | 0.001034097351833165 |
|  pi/4 |     4368.0    | 17.129411764705882 | 0.9328268692522991 | 0.5219107750974435 | 0.7224339243816306 | 0.8383010869545068 | 0.001034097351833165 |
|  pi/2 | 2092.79296875 |     8.20703125     | 0.9678160586726847 | 0.5530408896358597 | 0.7436671901031131 | 0.9225983101832996 | 0.001034097351833165 |
| 3pi/4 |     3989.0    | 15.64313725490196  | 0.938655307107926  | 0.5269965463553854 | 0.7259452777967396 | 0.8523290327871963 | 0.001034097351833165 |
+-------+---------------+--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+

Selected Features:
+---------------+---------------------+
|    Feature    |     Importance      |
+---------------+---------------------+
|    Contrast   |  0.182312925170068  |
| Dissimilarity | 0.22040816326530613 |
|  Homogeneity  | 0.16122448979591839 |
|      ASM      | 0.12312925170068029 |
|     Energy    | 0.12585034013605445 |
|  Correlation  | 0.18707482993197277 |
|    Entropy    |         0.0         |
+---------------+---------------------+
```

Feature Selection for Common_Rust:-



```
Feature Selection for Common_Rust.JPG

Extracted Features:
+-------+-------------------+---------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
| Angle |      Contrast     |    Dissimilarity    |    Homogeneity     |        ASM         |       Energy       |    Correlation     |       Entropy        |
+-------+-------------------+---------------------+--------------------+--------------------+--------------------+--------------------+----------------------+
|   0   |      39.84375     |       0.15625       | 0.9993872643250392 | 0.9575619695737516 | 0.9785509539997146 | 0.9853500199030347 | 0.001034097351833165 |
|  pi/4 |       100.0       | 0.39215686274509803 | 0.9984621536000985 | 0.9573914311193416 | 0.9784638118598672 | 0.962557698284641  | 0.001034097351833165 |
|  pi/2 |    74.70703125    |      0.29296875     | 0.9988511206094485 | 0.957393433579295  | 0.9784648351265849 | 0.9722883486189138 | 0.001034097351833165 |
| 3pi/4 | 75.00000000000001 |  0.2941176470588236 | 0.9988466152000739 | 0.9577895975596291 | 0.9786672557920946 | 0.9719081966866367 | 0.001034097351833165 |
+-------+-------------------+---------------------+--------------------+--------------------+--------------------+--------------------+----------------------+

Selected Features:
+---------------+---------------------+
|    Feature    |     Importance      |
+---------------+---------------------+
|    Contrast   | 0.19115646258503396 |
| Dissimilarity |  0.2176870748299319 |
|  Homogeneity  | 0.15918367346938778 |
|      ASM      | 0.11428571428571431 |
|     Energy    | 0.12925170068027214 |
|  Correlation  | 0.18843537414965983 |
|    Entropy    |         0.0         |
+---------------+---------------------+
```

Feature selection for cercospora disease :-



```
Feature Selection for Cercospora_disease.JPG

Extracted Features:
+-------+---------------+--------------------+--------------------+---------------------+--------------------+--------------------+---------------------+
| Angle |    Contrast   |   Dissimilarity    |    Homogeneity     |         ASM         |       Energy       |    Correlation     |       Entropy       |
+-------+---------------+--------------------+--------------------+---------------------+--------------------+--------------------+---------------------+
|   0   |  3719.4140625 |     14.5859375     | 0.9428011247424107 | 0.48733854758010736 | 0.6980963741347661 | 0.8753094035642773 | 0.00103409735183165 |
|  pi/4 |     4400.0    | 17.254901960784313 | 0.9323347584043304 |  0.4782065862049036 | 0.6915248268897535 | 0.8524843729025774 | 0.00103409735183165 |
|  pi/2 | 1959.31640625 |     7.68359375     | 0.9698687231838033 |  0.5118097582050398 | 0.7154088049535313 | 0.934348704693716  | 0.00103409735183165 |
| 3pi/4 |     4085.0    | 16.019607843137255 | 0.9371789745640204 | 0.48242317341347285 | 0.6945668962839165 | 0.8630438311653013 | 0.00103409735183165 |
+-------+---------------+--------------------+--------------------+---------------------+--------------------+--------------------+---------------------+

Selected Features:
+---------------+---------------------+
|    Feature    |      Importance     |
+---------------+---------------------+
|    Contrast   |  0.182312925170068  |
| Dissimilarity | 0.22040816326530613 |
|  Homogeneity  | 0.16122448979591839 |
|      ASM      | 0.12312925170068029 |
|     Energy    | 0.12585034013605445 |
|  Correlation  | 0.18707482993197277 |
|    Entropy    |         0.0         |
+---------------+---------------------+
```

The equation for calculating the importance score of a feature in a random forest model is as follows :

importance_score(feature) = sum((impurity_before_split - impurity_after_split) / num_trees)

where:

importance_score(feature): the importance score of the feature

impurity_before_split: the impurity measure of the node before the split

impurity_after_split: the impurity measure of the node after the split

num_trees: the total number of trees in the random forest model

The importance score of each feature is then normalized by dividing by the sum of all importance scores, so that the sum of all importance scores is equal to 1.

Feature Selection using Hill Climbing :

Hill climbing is a search algorithm that can be used for feature selection in machine learning. The basic idea behind hill climbing is to start with an initial set of features and iteratively add or remove features in a way that maximizes the performance of a model.The algorithm starts with an initial solution, which can be a subset of the available features. Then, it evaluates the performance of the solution using a performance metric, such as accuracy or AUC. The algorithm then generates a set of neighboring solutions by adding or removing a single feature from the current solution. It evaluates the performance of each neighboring solution and selects the best one. If the performance of the best neighboring solution is better than the current solution, the algorithm moves to the best neighboring solution. Otherwise, it stops and returns the current solution.The equation for selecting the best neighboring solution in hill climbing for feature selection is as follows: **best_solution = argmax(S) f(S)**

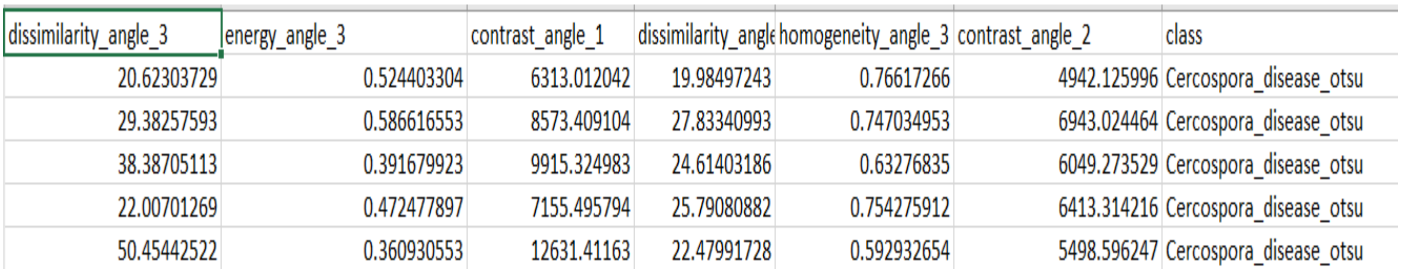
where:

best_solution: the best neighboring solution

S: the set of neighboring solutions

f(S): the performance metric evaluated on each neighboring solution

Feature values obtained by applying apply hill-climbing to all 4-classes for feature selection:



dissimilarity_angle_3	energy_angle_3	contrast_angle_1	dissimilarity_angl	homogeneity_angle_3	contrast_angle_2	class
20.62303729	0.524403304	6313.012042	19.98497243	0.76617266	4942.125996	Cercospora_disease_otsu
29.38257593	0.586616553	8573.409104	27.83340993	0.747034953	6943.024464	Cercospora_disease_otsu
38.38705113	0.391679923	9915.324983	24.61403186	0.63276835	6049.273529	Cercospora_disease_otsu
22.00701269	0.472477897	7155.495794	25.79080882	0.754275912	6413.314216	Cercospora_disease_otsu
50.45442522	0.360930553	12631.41163	22.47991728	0.592932654	5498.596247	Cercospora_disease_otsu

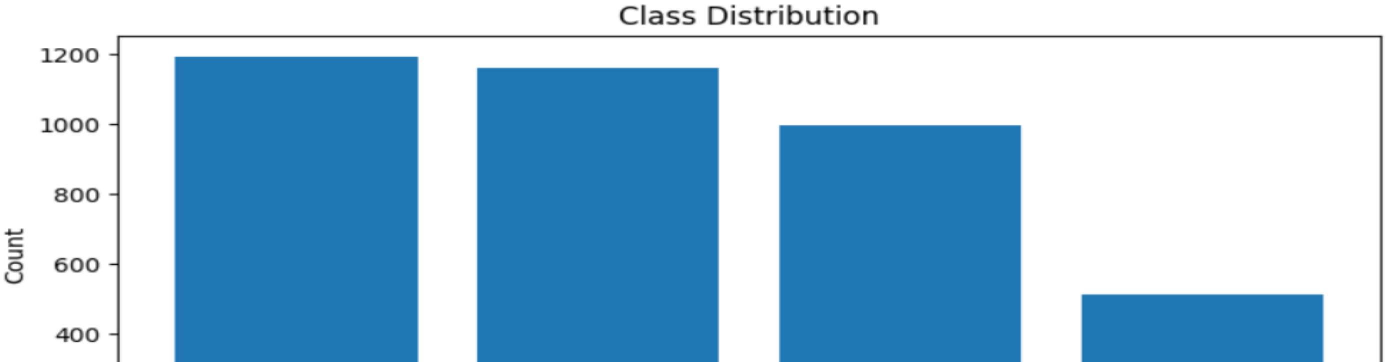
Selected features: ['dissimilarity_angle_3', 'energy_angle_3', 'contrast_angle_1', 'dissimilarity_angle_2', 'homogeneity_angle_3', 'contrast_angle_2']

Accuracy: 0.5948253557567917

Visualisation of the Selected features on our Preprocessed Dataset

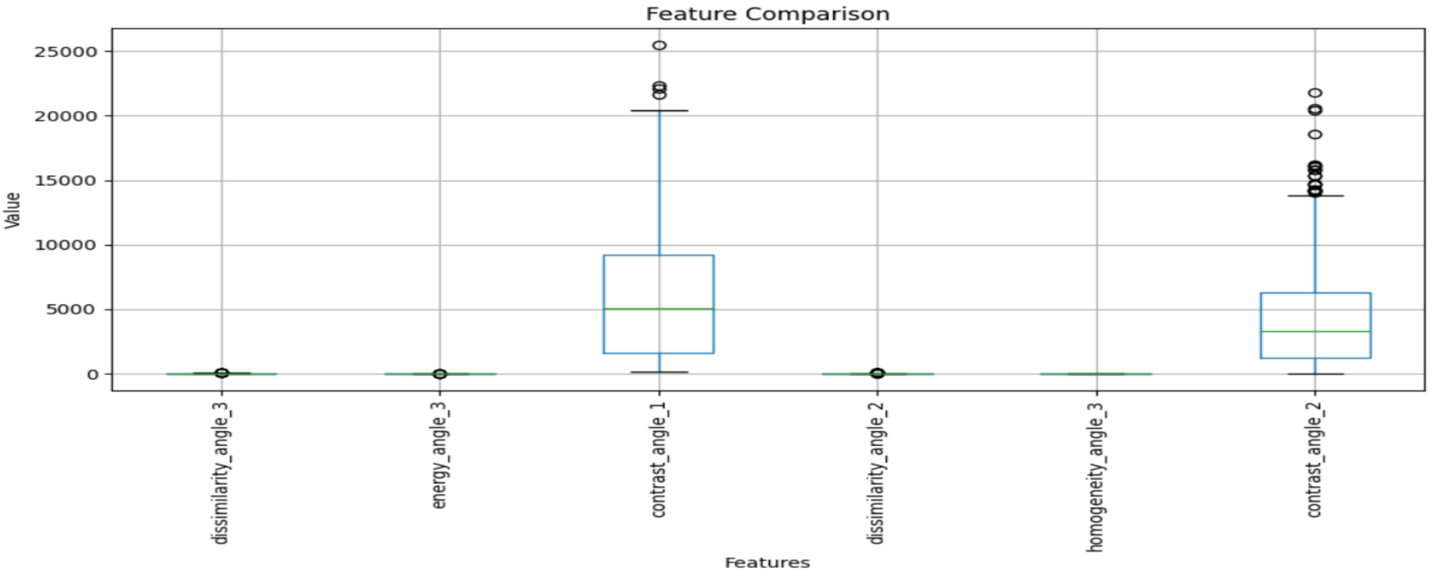
Bar Chart: Class Distribution

This will display a bar chart showing the count of samples for each class:



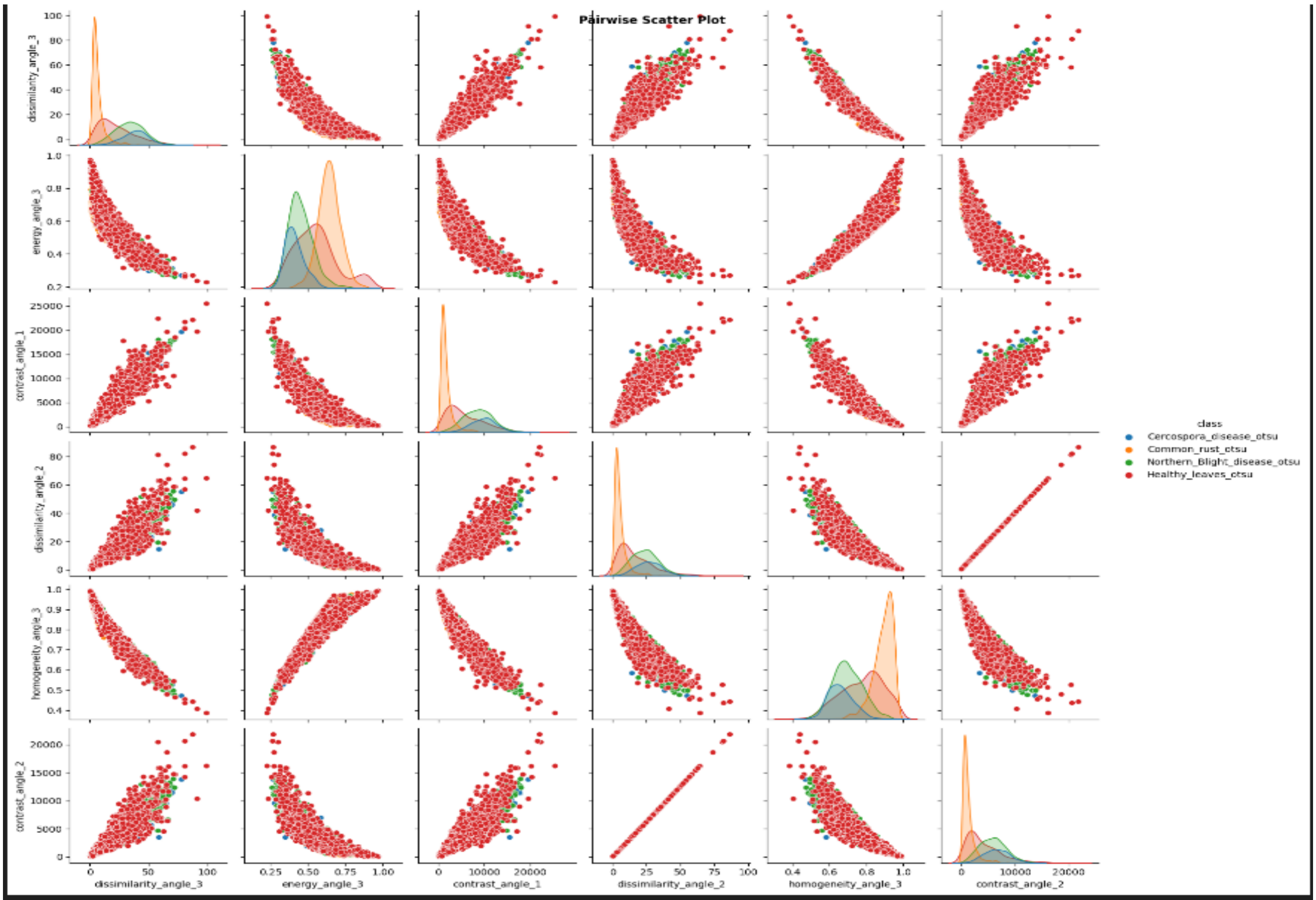
Box Plot: Feature Comparison

This will show a box plot for each feature, grouped by class, allowing us to compare the distributions and identify any potential differences.



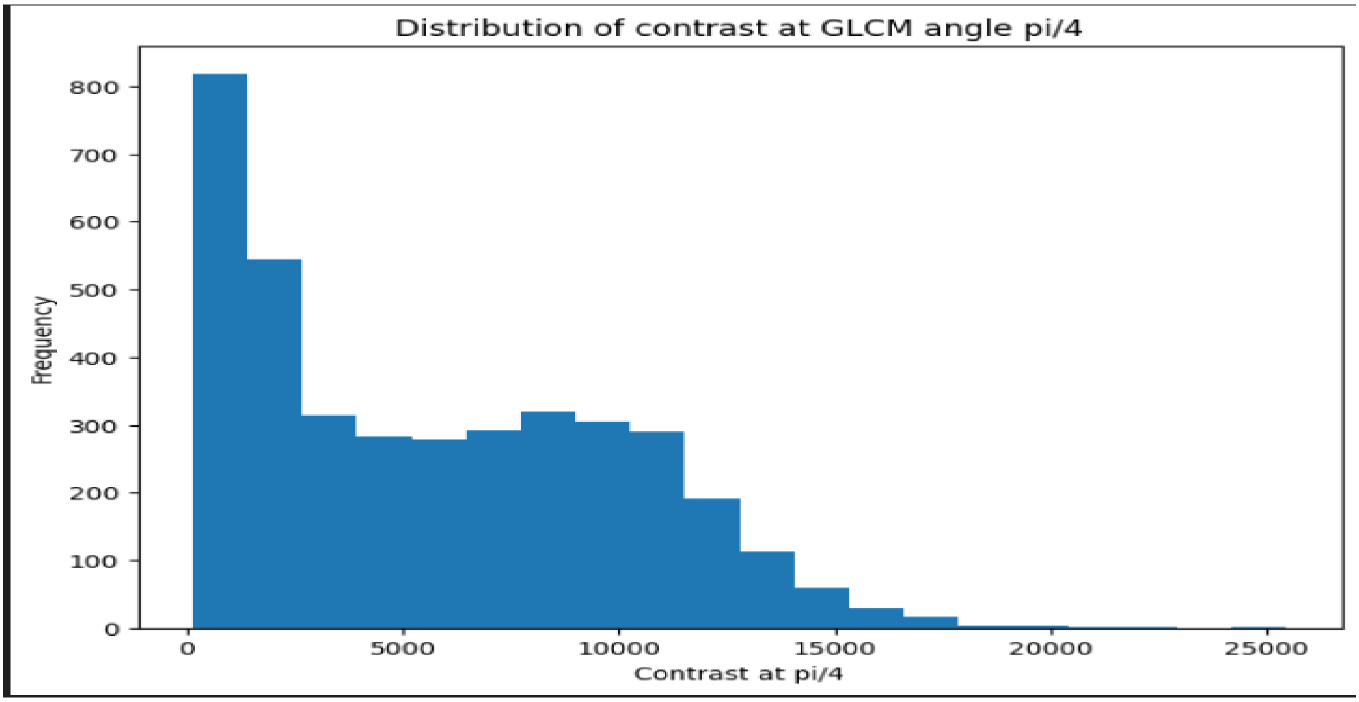
Pairwise Scatter Plot: Feature Relationships

This will generate a scatter plot matrix where each point represents the relationship between two features, with different colors indicating different classes.



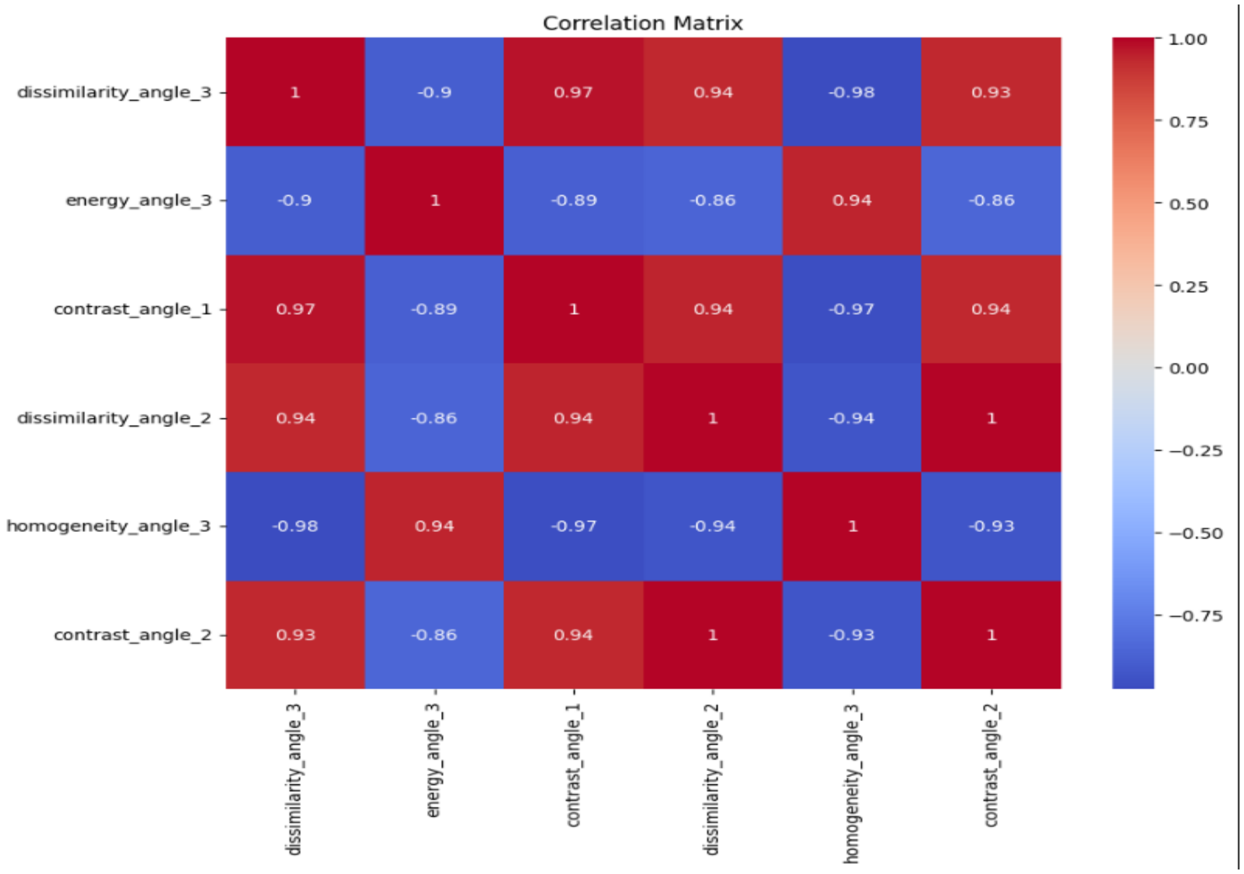
Histogram: Distribution of a Single Feature

This will display a histogram showing the distribution of the 'contrast' feature.

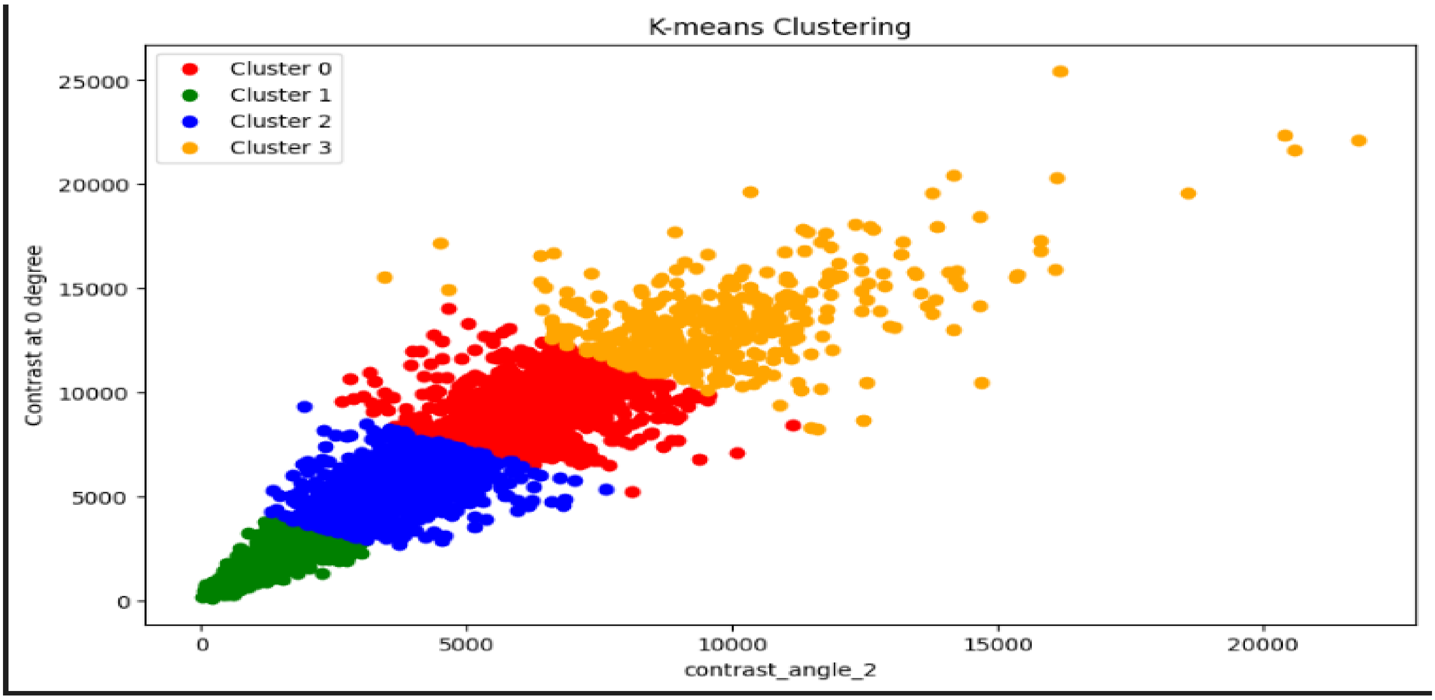


Heatmap: Correlation Matrix

This will generate a heatmap where each cell represents the correlation between two features. Higher values indicate a stronger positive correlation, while lower values indicate a negative correlation.

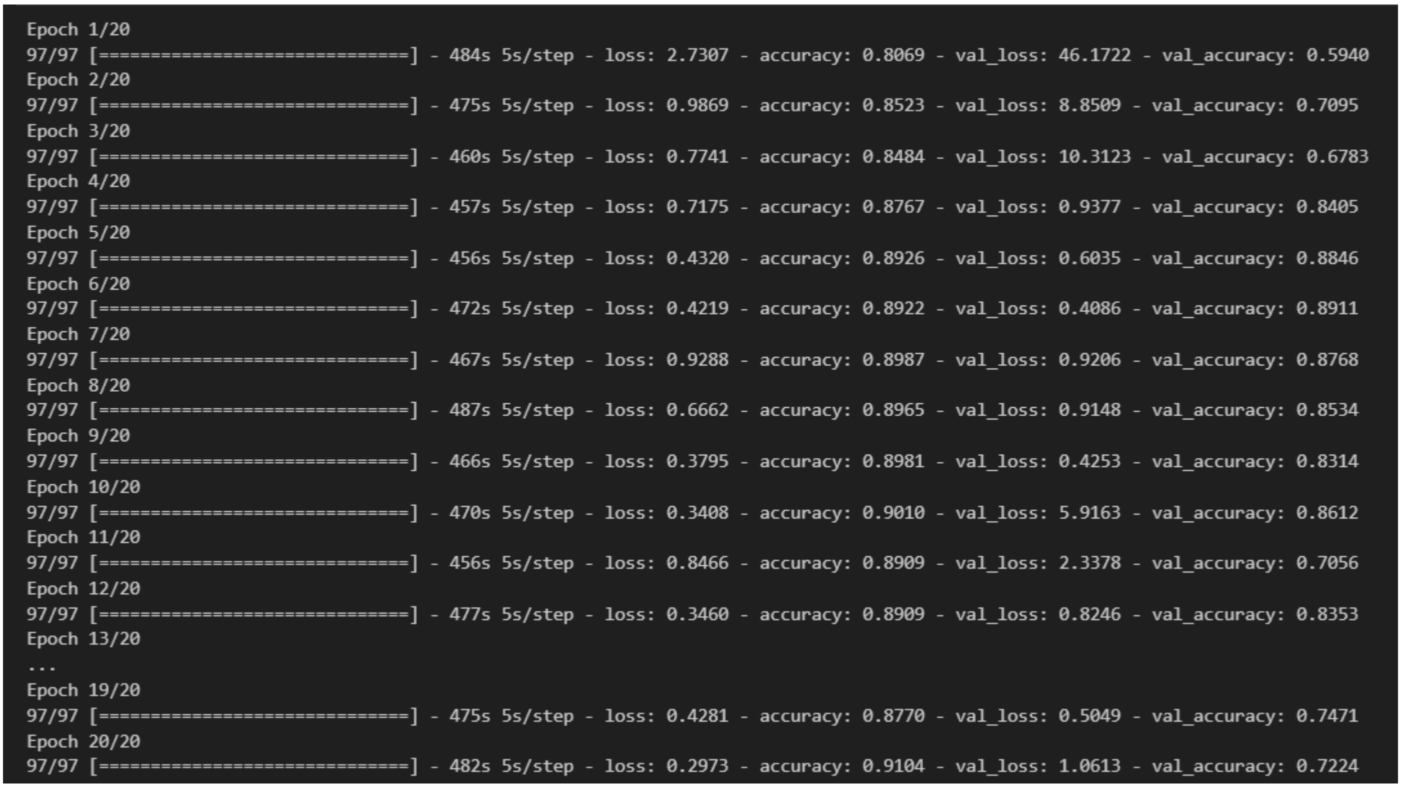


K-Means Clustering :



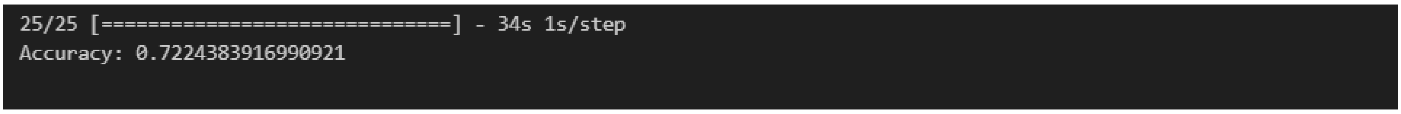
Training by Machine Learning and CNN Models

Model 1: CNN for 25 epochs



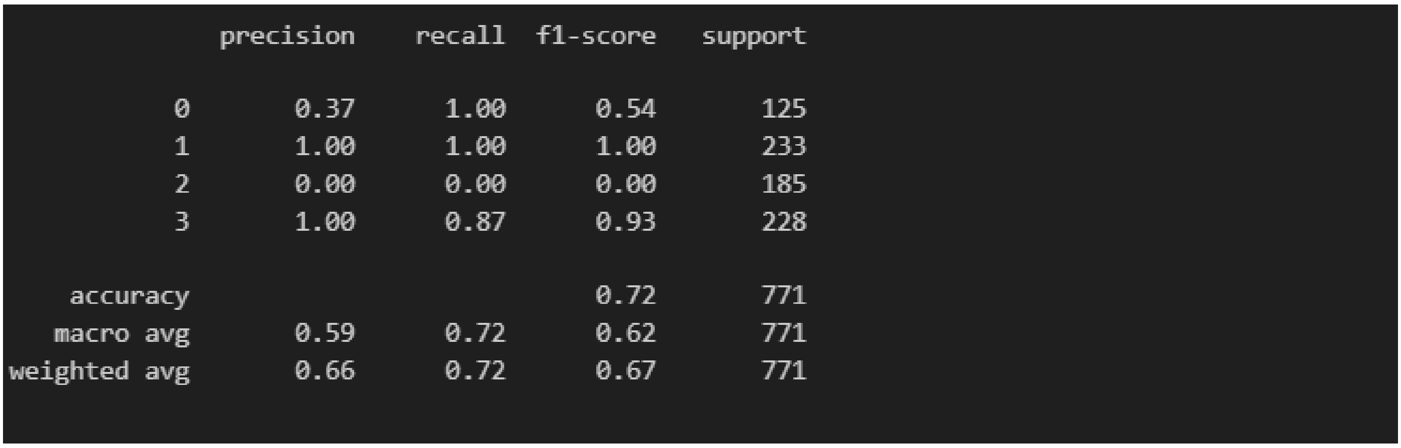
```
Epoch 1/20
97/97 [==============================] - 484s 5s/step - loss: 2.7307 - accuracy: 0.8069 - val_loss: 46.1722 - val_accuracy: 0.5940
Epoch 2/20
97/97 [==============================] - 475s 5s/step - loss: 0.9869 - accuracy: 0.8523 - val_loss: 8.8509 - val_accuracy: 0.7095
Epoch 3/20
97/97 [==============================] - 460s 5s/step - loss: 0.7741 - accuracy: 0.8484 - val_loss: 10.3123 - val_accuracy: 0.6783
Epoch 4/20
97/97 [==============================] - 457s 5s/step - loss: 0.7175 - accuracy: 0.8767 - val_loss: 0.9377 - val_accuracy: 0.8405
Epoch 5/20
97/97 [==============================] - 456s 5s/step - loss: 0.4320 - accuracy: 0.8926 - val_loss: 0.6035 - val_accuracy: 0.8846
Epoch 6/20
97/97 [==============================] - 472s 5s/step - loss: 0.4219 - accuracy: 0.8922 - val_loss: 0.4086 - val_accuracy: 0.8911
Epoch 7/20
97/97 [==============================] - 467s 5s/step - loss: 0.9288 - accuracy: 0.8987 - val_loss: 0.9206 - val_accuracy: 0.8768
Epoch 8/20
97/97 [==============================] - 487s 5s/step - loss: 0.6662 - accuracy: 0.8965 - val_loss: 0.9148 - val_accuracy: 0.8534
Epoch 9/20
97/97 [==============================] - 466s 5s/step - loss: 0.3795 - accuracy: 0.8981 - val_loss: 0.4253 - val_accuracy: 0.8314
Epoch 10/20
97/97 [==============================] - 470s 5s/step - loss: 0.3408 - accuracy: 0.9010 - val_loss: 5.9163 - val_accuracy: 0.8612
Epoch 11/20
97/97 [==============================] - 456s 5s/step - loss: 0.8466 - accuracy: 0.8909 - val_loss: 2.3378 - val_accuracy: 0.7056
Epoch 12/20
97/97 [==============================] - 477s 5s/step - loss: 0.3460 - accuracy: 0.8909 - val_loss: 0.8246 - val_accuracy: 0.8353
Epoch 13/20
...
Epoch 19/20
97/97 [==============================] - 475s 5s/step - loss: 0.4281 - accuracy: 0.8770 - val_loss: 0.5049 - val_accuracy: 0.7471
Epoch 20/20
97/97 [==============================] - 482s 5s/step - loss: 0.2973 - accuracy: 0.9104 - val_loss: 1.0613 - val_accuracy: 0.7224
```

Accuracy :



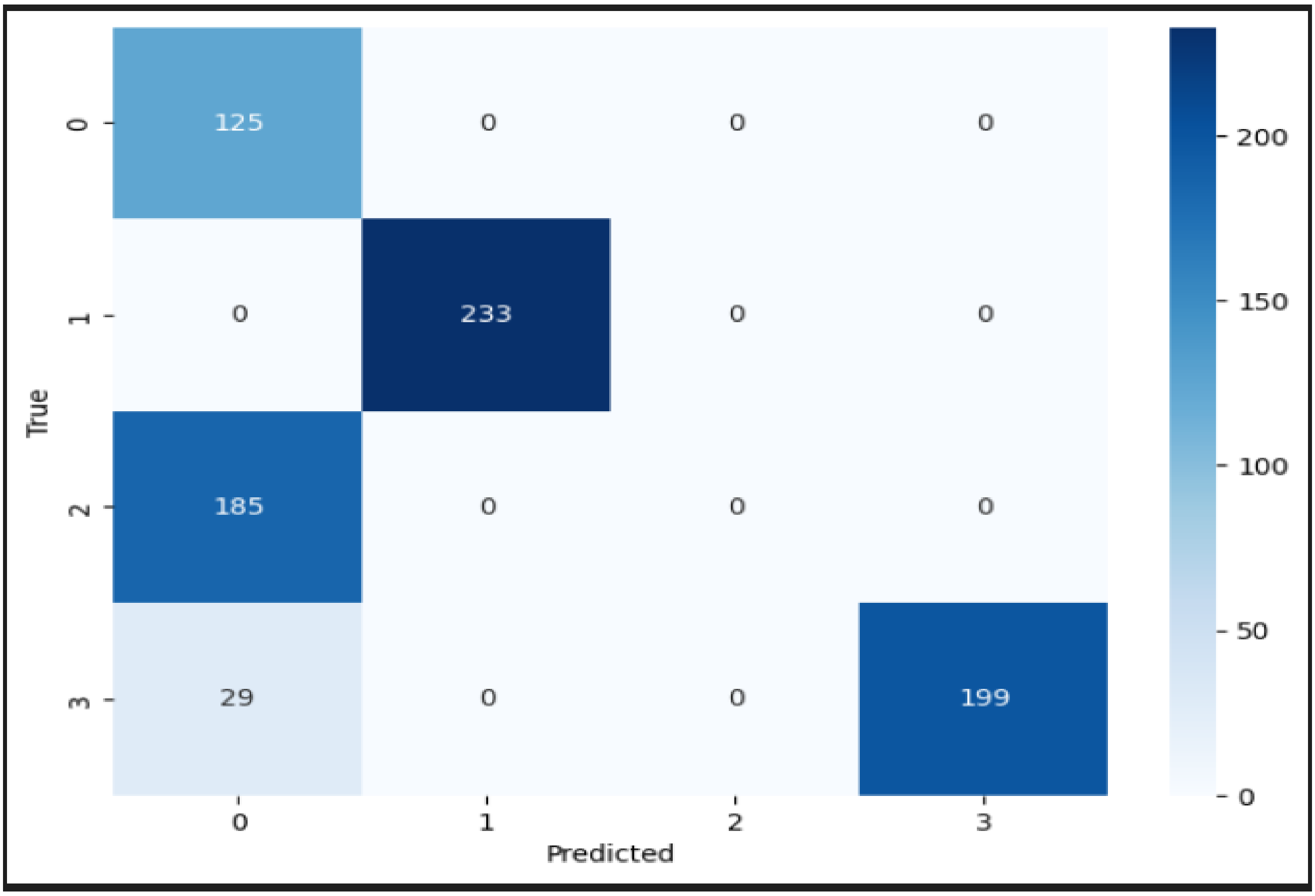
```
25/25 [==============================] - 34s 1s/step
Accuracy: 0.7224383916990921
```

Classification Report:

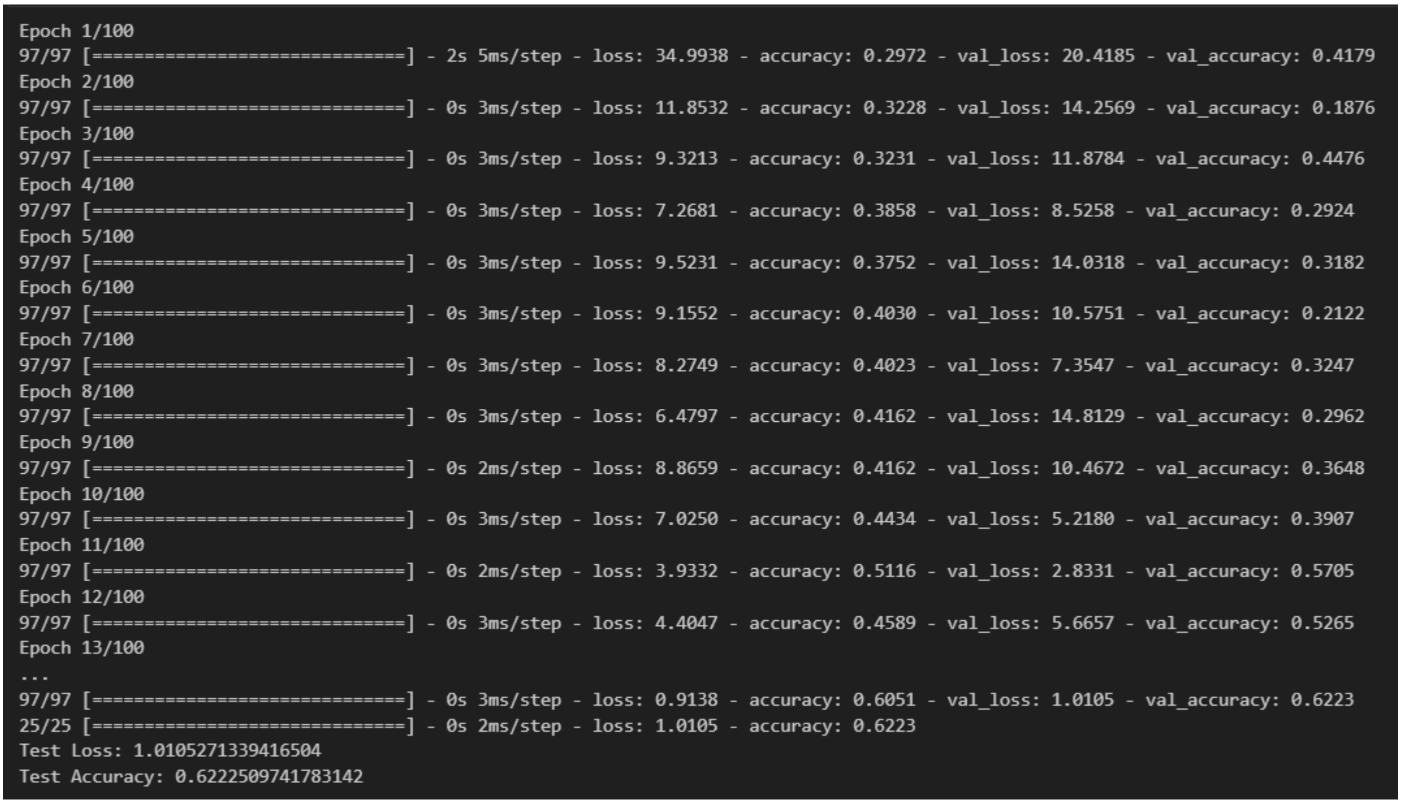


```
              precision    recall  f1-score   support

           0       0.37      1.00      0.54       125
           1       1.00      1.00      1.00       233
           2       0.00      0.00      0.00       185
           3       1.00      0.87      0.93       228

    accuracy                           0.72       771
   macro avg       0.59      0.72      0.62       771
weighted avg       0.66      0.72      0.67       771
```

Confusion matrix :

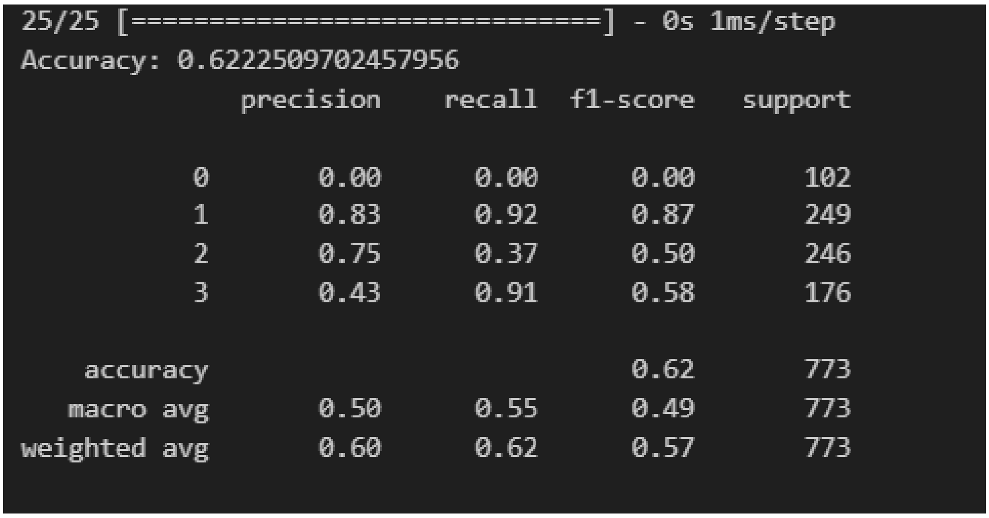


Model 2: ANN for 25 epochs:



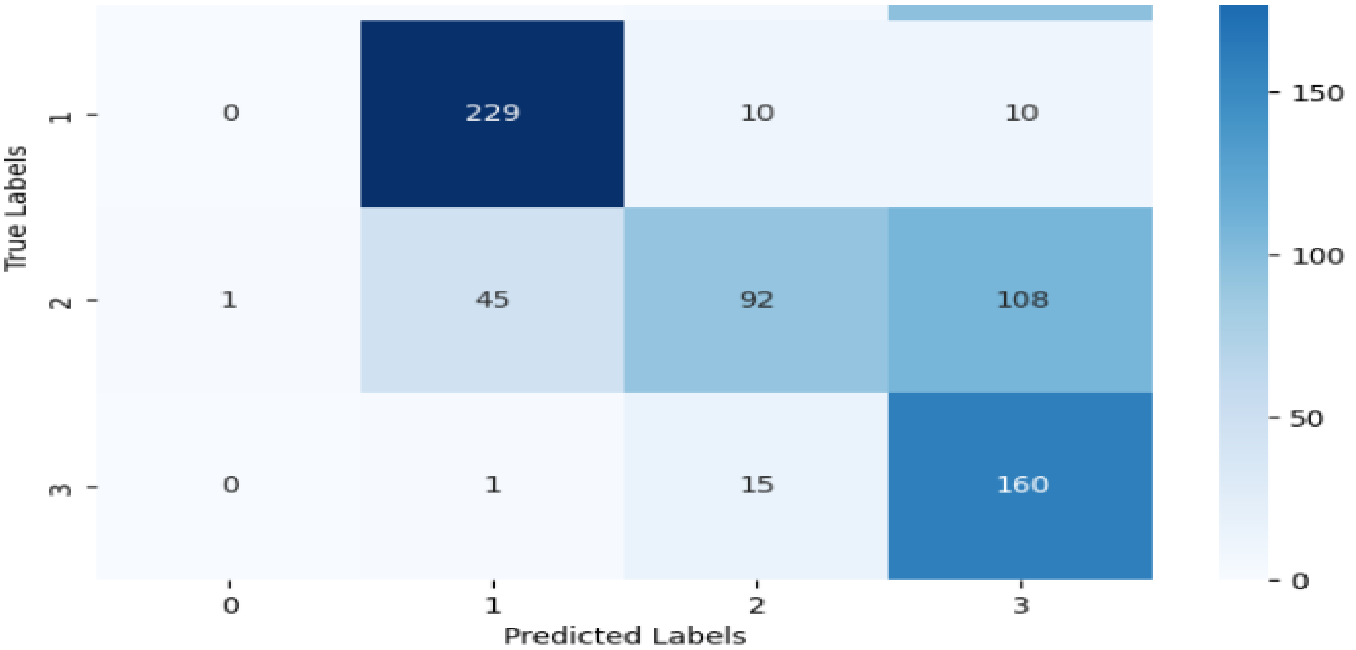
```
Epoch 1/100
97/97 [==============================] - 2s 5ms/step - loss: 34.9938 - accuracy: 0.2972 - val_loss: 20.4185 - val_accuracy: 0.4179
Epoch 2/100
97/97 [==============================] - 0s 3ms/step - loss: 11.8532 - accuracy: 0.3228 - val_loss: 14.2569 - val_accuracy: 0.1876
Epoch 3/100
97/97 [==============================] - 0s 3ms/step - loss: 9.3213 - accuracy: 0.3231 - val_loss: 11.8784 - val_accuracy: 0.4476
Epoch 4/100
97/97 [==============================] - 0s 3ms/step - loss: 7.2681 - accuracy: 0.3858 - val_loss: 8.5258 - val_accuracy: 0.2924
Epoch 5/100
97/97 [==============================] - 0s 3ms/step - loss: 9.5231 - accuracy: 0.3752 - val_loss: 14.0318 - val_accuracy: 0.3182
Epoch 6/100
97/97 [==============================] - 0s 3ms/step - loss: 9.1552 - accuracy: 0.4030 - val_loss: 10.5751 - val_accuracy: 0.2122
Epoch 7/100
97/97 [==============================] - 0s 3ms/step - loss: 8.2749 - accuracy: 0.4023 - val_loss: 7.3547 - val_accuracy: 0.3247
Epoch 8/100
97/97 [==============================] - 0s 3ms/step - loss: 6.4797 - accuracy: 0.4162 - val_loss: 14.8129 - val_accuracy: 0.2962
Epoch 9/100
97/97 [==============================] - 0s 2ms/step - loss: 8.8659 - accuracy: 0.4162 - val_loss: 10.4672 - val_accuracy: 0.3648
Epoch 10/100
97/97 [==============================] - 0s 3ms/step - loss: 7.0250 - accuracy: 0.4434 - val_loss: 5.2180 - val_accuracy: 0.3907
Epoch 11/100
97/97 [==============================] - 0s 2ms/step - loss: 3.9332 - accuracy: 0.5116 - val_loss: 2.8331 - val_accuracy: 0.5705
Epoch 12/100
97/97 [==============================] - 0s 3ms/step - loss: 4.4047 - accuracy: 0.4589 - val_loss: 5.6657 - val_accuracy: 0.5265
Epoch 13/100
...
97/97 [==============================] - 0s 3ms/step - loss: 0.9138 - accuracy: 0.6051 - val_loss: 1.0105 - val_accuracy: 0.6223
25/25 [==============================] - 0s 2ms/step - loss: 1.0105 - accuracy: 0.6223
Test Loss: 1.0105271339416504
Test Accuracy: 0.6222509741783142
```

Accuracy and Classification Report:



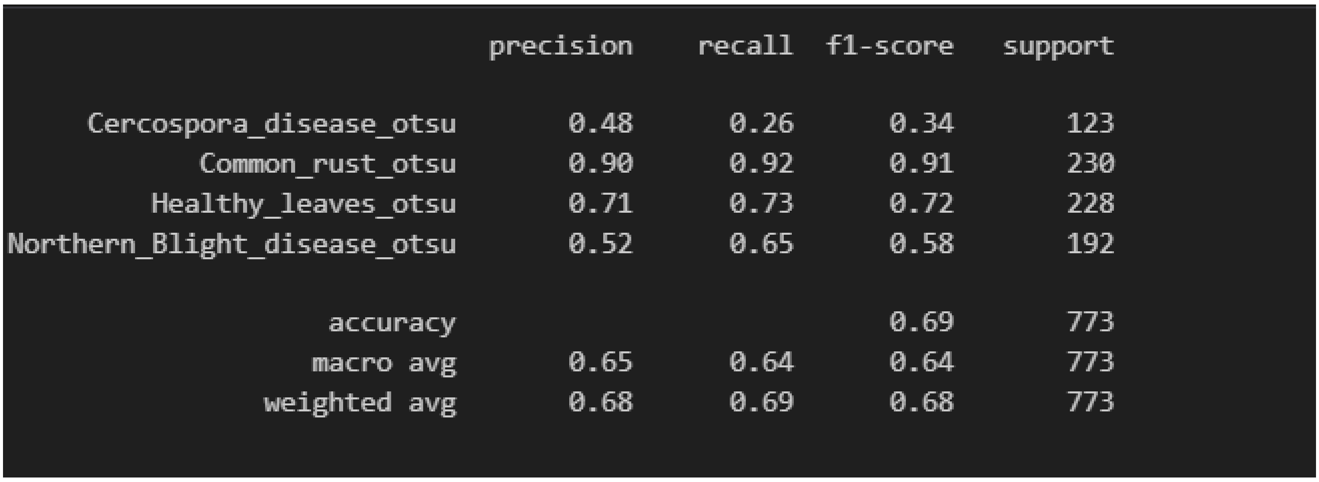
```
25/25 [==============================] - 0s 1ms/step
Accuracy: 0.6222509702457956
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       102
           1       0.83      0.92      0.87       249
           2       0.75      0.37      0.50       246
           3       0.43      0.91      0.58       176

    accuracy                           0.62       773
   macro avg       0.50      0.55      0.49       773
weighted avg       0.60      0.62      0.57       773
```

Confusion matrix :



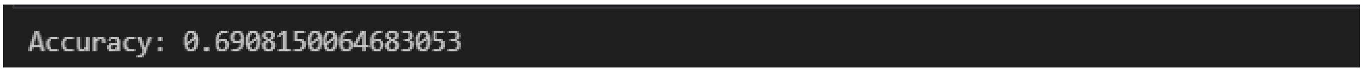
Model 3: Random Forest

Classification Report:



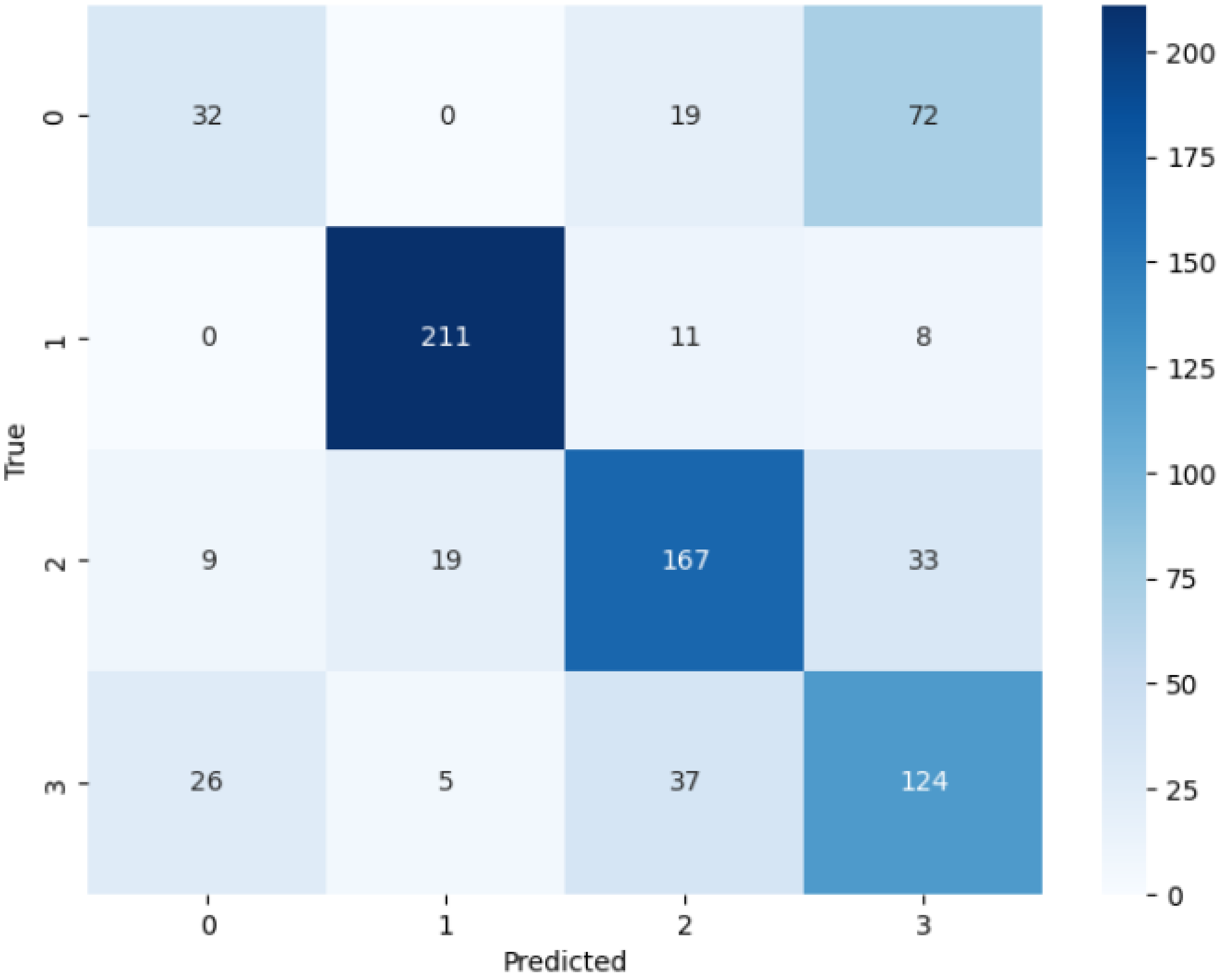
```
                             precision    recall  f1-score   support

     Cercospora_disease_otsu       0.48      0.26      0.34       123
            Common_rust_otsu       0.90      0.92      0.91       230
         Healthy_leaves_otsu       0.71      0.73      0.72       228
Northern_Blight_disease_otsu       0.52      0.65      0.58       192

                    accuracy                           0.69       773
                   macro avg       0.65      0.64      0.64       773
                weighted avg       0.68      0.69      0.68       773
```

Accuracy:



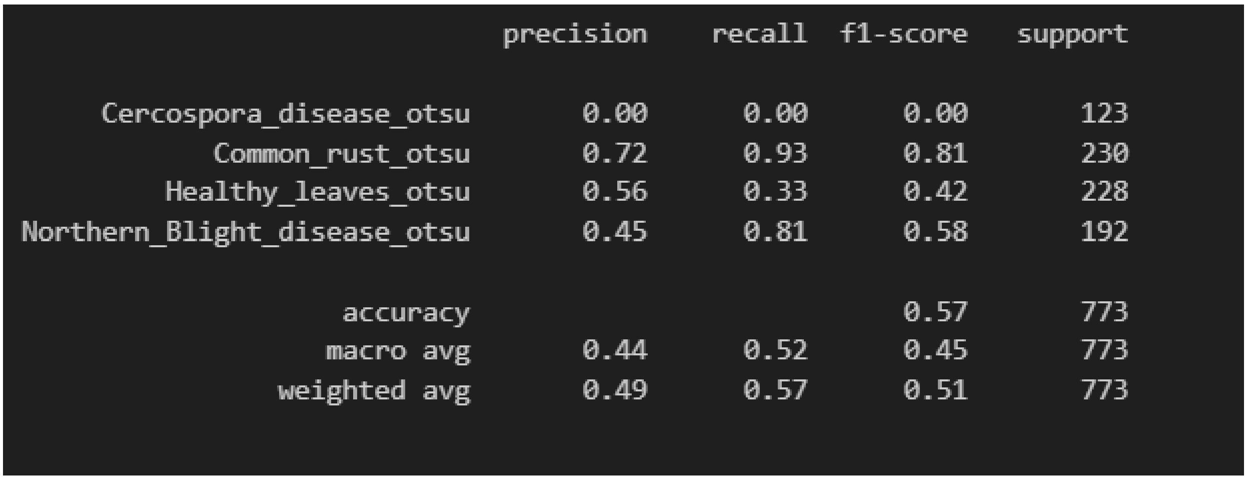
```
Accuracy: 0.6908150064683053
```

Confusion Matrix:



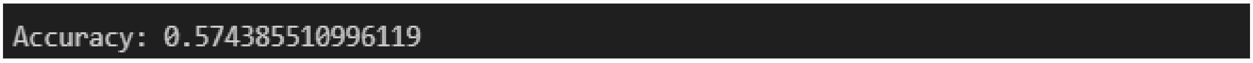
Model 4: SVM

Classification Report:



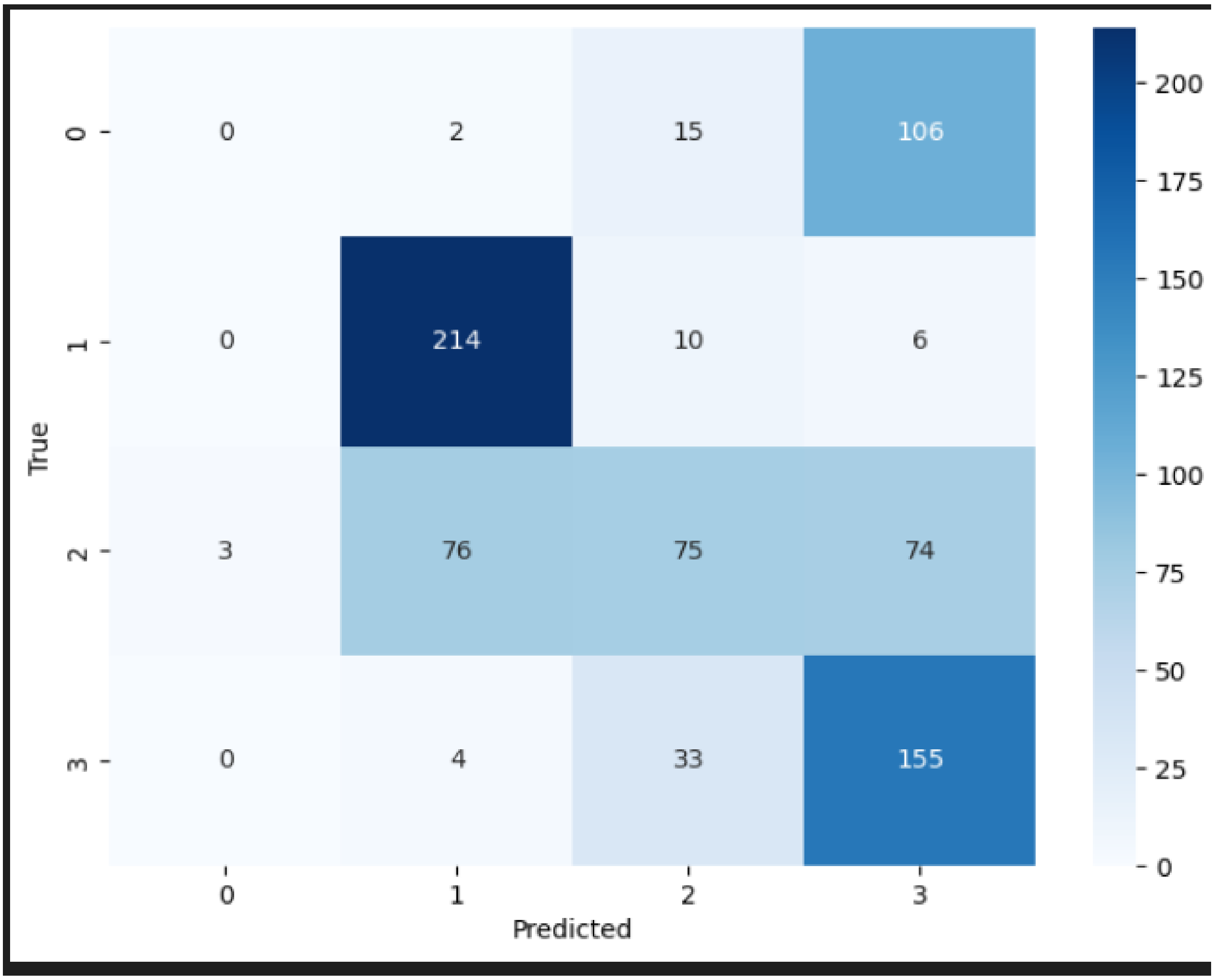
```
                            precision    recall  f1-score   support

    Cercospora_disease_otsu       0.00      0.00      0.00       123
           Common_rust_otsu       0.72      0.93      0.81       230
        Healthy_leaves_otsu       0.56      0.33      0.42       228
Northern_Blight_disease_otsu       0.45      0.81      0.58       192

                   accuracy                           0.57       773
                  macro avg       0.44      0.52      0.45       773
               weighted avg       0.49      0.57      0.51       773
```

Accuracy:



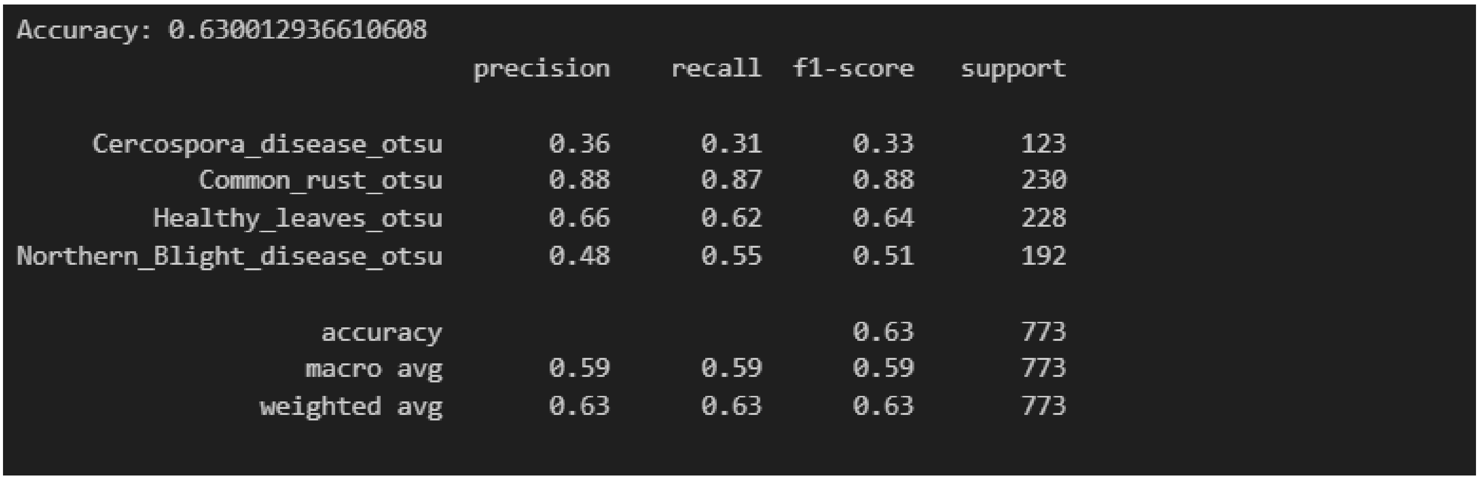
```
Accuracy: 0.574385510996119
```

Confusion Matrix :



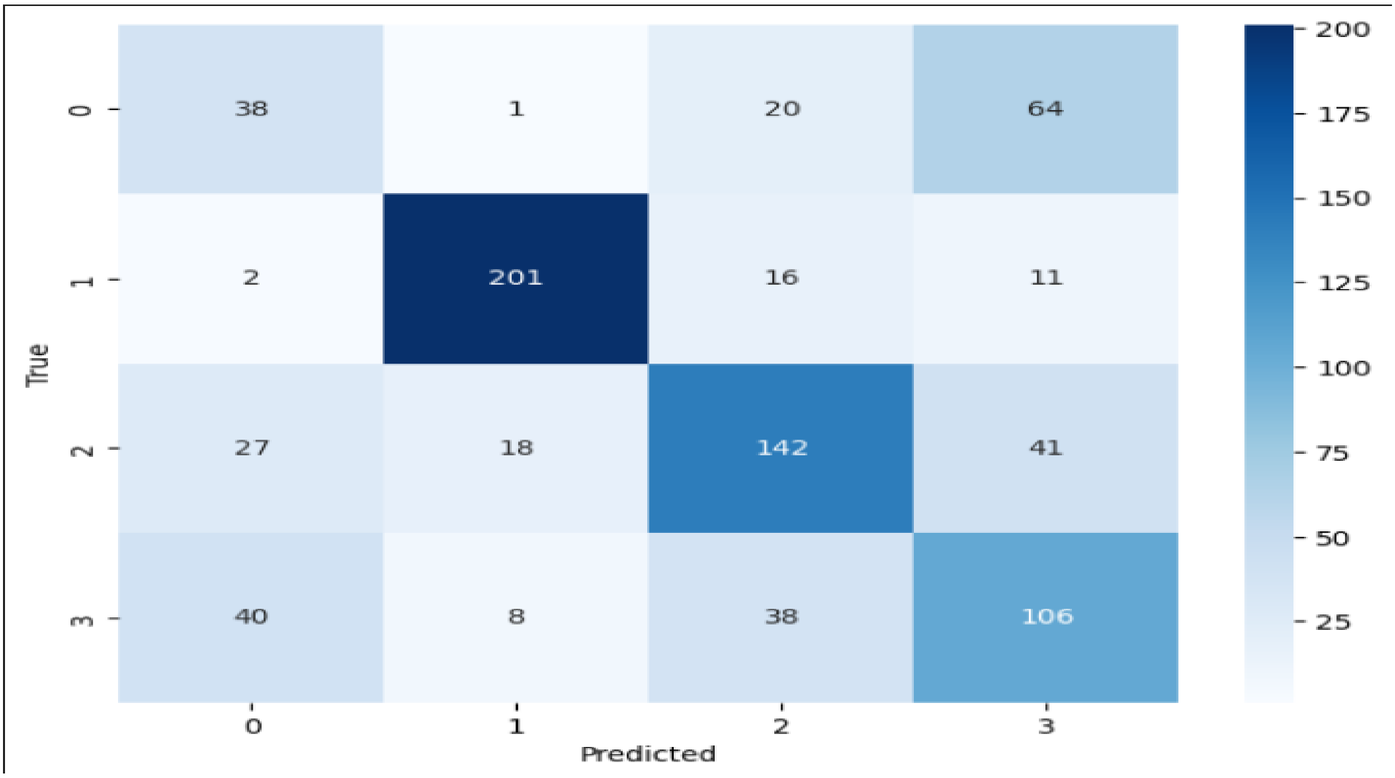
Model 5: Decision Tree

Classification Report & Accuracy:



```
Accuracy: 0.630012936610608
                             precision    recall  f1-score   support

     Cercospora_disease_otsu       0.36      0.31      0.33       123
            Common_rust_otsu       0.88      0.87      0.88       230
         Healthy_leaves_otsu       0.66      0.62      0.64       228
Northern_Blight_disease_otsu       0.48      0.55      0.51       192

                    accuracy                           0.63       773
                   macro avg       0.59      0.59      0.59       773
                weighted avg       0.63      0.63      0.63       773
```

Confusion Matrix:



Comparative Analysis :

S.No	Model Used	Accuracy	F1-Score	Classwise-Prediction
1.	CNN	72%	67%	i.) common_rust : 100% ii.) northern blight disease : 100% iii.) healthy : 0% iv.) cercospora disease : 37%
2.	ANN	62%	57%	i.) common_rust : 83% ii.) northern blight disease : 43% iii.) healthy : 76% iv.) cercospora disease : 0%
3.	Random Forest	69%	68%	i.) common_rust : 90% ii.) northern blight disease :53% iii.) healthy : 72% iv.) cercospora disease : 48%
4.	SVM	57%	51%	i.) common_rust : 72% ii.) northern blight disease : 46% iii.) healthy : 57% iv.) cercospora disease : 0%
5.	Decision Tree	63%	63%	i.) common_rust : 88% ii.) northern blight disease : 48% iii.) healthy : 66% iv.) cercospora disease : 36%

From the above comparative analysis table we can say that CNN (Convolutional Neural Network) is the best ML model to predict the disease in plants using leaves images as dataset.

As we can see, the Common_rust and Northern Blight are the best predicted class among all classes with accuracy of 100% using CNN model. But the total accuracy is much lower because the data is unbalanced. There are more samples in Common_rust and Northern Blight and less number of samples in the remaining two classes.

Future Upgradation :

From the result we can surely conclude that accuracy is not up to the mark and it is much lower. It happens because the data is unbalanced i.e some classes are having a high number of sample sizes compared to others. Hence to increase the accuracy of models we need to get a balanced dataset with nearly the same sample sizes belonging to each class.

Test Data Selection : The PlantVillage, which is openly available for research, provided the dataset for our study. It has a good number of leaf photos for each of the 14 different plant varieties.

Here, we are utilizing Corn_(maize)_Cercospora_leaf_spot Gray_leaf_spot, Corn_(maize)_Common_rust_, Corn_(maize)_healthy and Corn_(maize)_Northern_Leaf_Blight. Here Healthy has 1161 images, Common_rust has 1191 images, Cercospora has 512 images, and Northan_Blight has 984 image

REFERENCES :

[1] Sandika, B., Avil, S., Sanat, S., & Srinivasu, P. (2016). Random forest based classification of diseases in grapes from images captured in uncontrolled environments. *2016 IEEE 13th International Conference on Signal Processing (ICSP).*

[2] Ahmed, K., Shahidi, T. R., Irfanul Alam, S. M., & Momen, S. (2019). Rice leaf disease detection using machine learning techniques. *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI).*

[3] Aasha Nandhini, S., Hemalatha, R., Radha, S., & Indumathi, K. (2017, December 4). Web Enabled Plant Disease Detection System for Agricultural Applications Using WMSN. *Wireless Personal Communications*, *102*(2), 725–740. https://doi.org/10.1007/s11277-017-5092-4

[4] Rumy SSH, Hossain MIA, Jahan F, and Tanvin T (2021) has done an IoT-based System with Edge Intelligence for Rice Leaf Disease Detection using Machine Learning.

[5] Datta, A., Dey, A., & Dey, K. N. (2019). Automatic multiclass classification of foliar leaf diseases using statistical and color feature extraction and support vector machine. In *Communications in Computer and Information Science* (pp. 3–15). Springer Singapore.

[6] Hossain, E., Hossain, M. F., & Rahaman, M. A. (2019). A color and texture based approach for the detection and classification of plant leaf disease using KNN classifier. *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE).*

[7] H. Wang, G. Li, Z. Ma, and X. Li, "Application of neural networksto image recognition of plant diseases," in Systems and Informatics(ICSAI), 2012 International Conference on. IEEE, 2012, pp. 2159–2164.

[8] S. Biswas, B. Jagyasi, B. P. Singh, and M. Lal, "Severity identification ofpotato late blight disease from crop images captured under uncontrolledenvironment," in Humanitarian Technology Conference-(IHTC), 2014IEEE Canada International. IEEE, 2014, pp. 1–5.

[9] P. Soille, Morphological image analysis: principles and applications.Springer Science & Business Media, 2013.