T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

CHERRY LEAF MOLD DETECTION USING NON-EXCLUSIVE CLUSTERING

OZAN DORUK YAVUZ

SUPERVISOR
DR. BURCU YILMAZ

GEBZE
2025

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CHERRY LEAF MOLD DETECTION USING NON-EXCLUSIVE CLUSTERING

OZAN DORUK YAVUZ

SUPERVISOR
DR. BURCU YILMAZ

2025
GEBZE

ABSTRACT

This project focuses on detecting mold in cherry leaves using clustering methods. Fuzzy C-Means (non-exclusive clustering) was chosen for its ability to handle overlapping data, while exclusive clustering methods such as DBSCAN, K-Means, and Hierarchical Clustering were used for comparison. Key results include the identification of healthy and diseased leaves, improvements through DBSCAN noise removal. The custom implementations of Fuzzy C-Means and DBSCAN provided flexibility into the dataset.

Keywords: Non-Exclusive Clustering, Cherry Leaf Mold Detection, Computer Vision, Fuzzy C-Means Clustering, Exclusive Clustering, DBSCAN, K-Means Clustering, Hierarchical Clustering, Color Histogram Feature Extraction, Adjusted Rand Index (ARI), Noise Removal with DBSCAN.

ACKNOWLEDGEMENT

I am grateful for the lectures throughout the semester, and thankful for the invaluable support and guidance to my project provided by my supervisor, Dr.Burcu Yılmaz. In addition, I am grateful to my family who owns the cherry garden that inspires me to do this project and for their support.

Ozan Doruk YAVUZ

CONTENTS

1. PROBLEM DEFINITION

Project is inspired from my family's small cherry garden. With knowing the problem with the topic about mold and its effects about the cherry trees, I have decided to implement this project.

In cherry farming, leaf mold significantly impacts tree health and crop yield. Detecting mold-affected leaves early allows timely medication which improves outcomes. This project aims to classify cherry leaves into healthy and molded categories using clustering techniques. Non-exclusive clustering methods, particularly Fuzzy C-Means, were employed to capture overlapping data. With that overlapping data we can detect some missed molded leaves for medication and save some healthy trees from unnecessary medication. Exclusive clustering methods used for comparison.

Figure 1.1: Cherry Tree from our family's garden.

2. THEORETICAL FOUNDATION AND LITERATURE ANALYSIS

2.1. Non-Exclusive clustering

Non-exclusive clustering allows data points to belong to multiple clusters simultaneously, with varying degrees of membership. This flexibility is achieved by assigning membership values to each data point for every cluster, rather than forcing a single assignment. This is particularly useful for datasets where the boundaries between clusters are ambiguous or overlapping.

2.2. Exclusive clustering

Exclusive clustering, assigns each data point to a single cluster. This approach is more straightforward but less flexible, as it assumes that the clusters are distinct and non-overlapping which may end up with lacking point predictions in real life solutions.

2.3. Fuzzy C-Means (FCM)

Introduced by Dunn (1973) and enhanced by Bezdek (1981), FCM is widely used for overlapping datasets. Unlike exclusive methods, FCM assigns membership degrees, enabling partial membership for data points. This is particularly useful for datasets where classifications are ambiguous, such as partially mold-affected leaves.

2.4. DBSCAN

Proposed by Ester et al. (1996), DBSCAN clusters data based on density. It identifies outliers as noise, making it ideal for noisy datasets. It does not require specifying the number of clusters, making it adaptable to real-world scenarios.

2.5. Feature Extraction Using Color Histograms

Color histograms are a fundamental technique in image processing used for feature extraction. They capture the distribution of pixel intensities across color channels,

providing a compact representation of an image. This method is particularly effective in agricultural applications, where color changes often indicate plant health or disease. For this project, color histograms were used to extract features from cherry leaf images. By dividing the RGB channels into 8 bins each, a total of 512 features were generated per image.

2.6. K-Means and Hierarchical Clustering

K-Means clustering is a centroid-based algorithm that partitions data into clusters by minimizing within-cluster variance. While effective for distinct clusters, it struggles with overlapping or non-spherical clusters. Hierarchical clustering, on the other hand, creates a hierarchy of clusters through agglomerative or divisive approaches. This method is particularly useful for smaller datasets. In this project, K-Means and Hierarchical Clustering were implemented as benchmarks for comparison with non-exclusive clustering methods like FCM.

2.7. Adjusted Rand Index (ARI)

The Adjusted Rand Index (ARI) is a measure of agreement between two clusters, considering the assignment of data points to clusters. It evaluates the similarity of two partitions by examining how consistently data points are grouped together in the compared clusters.

3. PROJECT DESIGN

3.1. Dataset Overview

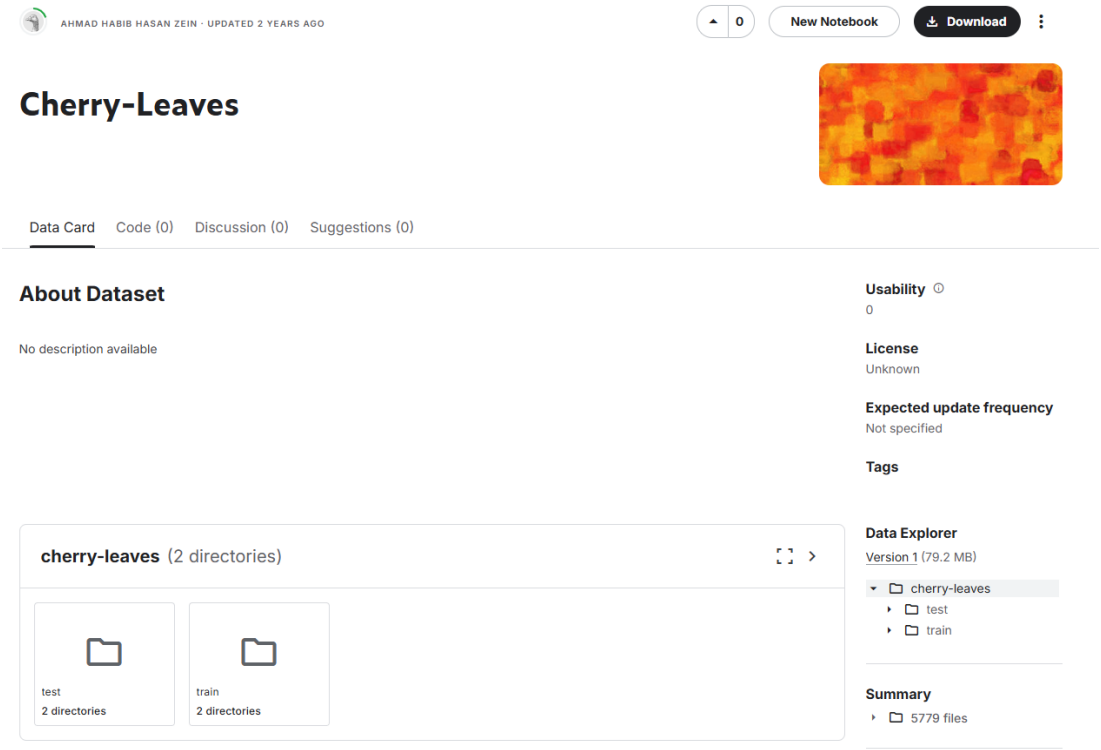

Figure 3.1: Cherry-Leaves Dataset in Kaggle.

The dataset comprises cherry leaf images categorized into healthy and mold-affected classes. A total of 4,205 images were processed. Each image was resized and converted into a feature vector using color histograms. Dataset link is provided in the references part.

3.2. Methodology

3.2.1. Feature Extraction

Feature extraction was performed using color histograms, which capture the color distribution in images. Each image was divided into three color channels (Red, Green, Blue), and histograms were computed with 8 bins per channel. The resulting feature vector for each image was 512-dimensional. (8x8x8)

3.2.2. Pre-Processing

To improve the clustering results, the dataset was cleaned using DBSCAN, a density-based clustering algorithm. DBSCAN helped remove noisy data points that did not belong to any cluster.

```
(cherry_env) PS C:\Users\ozand\Desktop\dataproje\code> python main.py
Loading and preprocessing data...

Cleaning the dataset using DBSCAN...
Original data shape: (4205, 512)
Cleaned data shape: (3928, 512)
Original labels shape: (4205,)
Cleaned labels shape: (3928,)
```

Figure 3.2: Cleaning the dataset with DBSCAN algorithm.

3.2.3. Non-Exclusive Clustering

As our Non-Exclusive method, Fuzzy c-Means (FCM) has been used. In addition it has been tested with different fuzziness metric (m) as 1.2, 1.5, 2.0, 2.5. To show the importance of the mid values which has membership values between 0.4 and 0.6, they have saved in addition to the high-certainty images which has more than 0.9 (%90 certainty).

```
Running fuzzy clustering for different m values...

Fuzziness parameter: m = 1.2
Saved: results/library_fuzzy_clustering_m1.2.png
Saved: results/custom_fuzzy_clustering_m1.2.png
Filtered and saved high-certainty images to 'results/predictions/custom_fuzzy_m1.2'
Filtered 217 images with membership values between 0.4 and 0.6.

Fuzziness parameter: m = 1.5
Saved: results/library_fuzzy_clustering_m1.5.png
Saved: results/custom_fuzzy_clustering_m1.5.png
Filtered and saved high-certainty images to 'results/predictions/custom_fuzzy_m1.5'
Filtered 688 images with membership values between 0.4 and 0.6.

Fuzziness parameter: m = 2.0
Saved: results/library_fuzzy_clustering_m2.0.png
Saved: results/custom_fuzzy_clustering_m2.0.png
Filtered and saved high-certainty images to 'results/predictions/custom_fuzzy_m2.0'
Filtered 1637 images with membership values between 0.4 and 0.6.

Fuzziness parameter: m = 2.5
Saved: results/library_fuzzy_clustering_m2.5.png
Saved: results/custom_fuzzy_clustering_m2.5.png
Filtered and saved high-certainty images to 'results/predictions/custom_fuzzy_m2.5'
Filtered 3928 images with membership values between 0.4 and 0.6.
```

Figure 3.3: Non-Exclusive Fuzzy c-Means run.

3.2.4. Exclusive Clustering

As our Exclusive methods, DBSCAN, KMeans and Hierarchical clustering has been used. Those methods used as comparable methods to our non-exclusive method.

```
Running exclusive clustering (DBSCAN)...
Saved: results/dbscan_clustering.png

Running exclusive clustering (KMeans)...
Saved: results/kmeans_clustering.png

Running exclusive clustering (Hierarchical)...
Saved: results/hierarchical_clustering.png
```

Figure 3.4: Exclusive runs which used as comparison.

3.3. Implementation of the Custom Methods

3.3.1. Custom Fuzzy c-Means (FCM) Implementation

The non-exclusive clustering method ”Custom Fuzzy c-Means” (FCM) we used allowed data points to be belong to multiple clusters with different degrees of membership.

The implementation of Fuzzy C-Means (FCM) is based on the following core formula:

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{||\mathbf{x}_i - \mathbf{v}_j||}{||\mathbf{x}_i - \mathbf{v}_k||} \right)^{\frac{2}{m-1}}}$$

Figure 3.5: Fuzzy C-Means Clustering Algorithm’s membership formula.

We can explain the variables of the formula as:

- Wij: Membership degree of the i-th data point to the j-th cluster.
- xi: i-th data point.
- vj: The centroid of the j-th cluster.
- m: The fuzziness parameter (m ¿ 1) controlling the degree of cluster overlap.
- c: The total number of clusters.

Lets start to explain the code of our implementation.

As initialization, membership matrix is initialized randomly using the Dirichlet distribution, ensuring each data point has valid membership values that sum to 1 across clusters.

Then we calculated the clusters centers, for each cluster j, the centroid vj is updated using a weighted average of data points, where weights are determined by the membership degrees raised to the power of m.

Membership degrees are recalculated for each data point and cluster using the core FCM formula. This ensures that each data point’s membership is updated based on its relative distance to cluster centers.

Then we check the convergence with the comparison of the membership matrix from the previous iteration with current matrix. This step is added for the method to stop iterating infinitely. I have adjusted this one with different values to balance the speed of the run with its accuracy.

Finally, for consistent cluster ordering, cluster centers and membership matrices are sorted based on the first dimension of cluster centers.

3.3.2. Custom DBSCAN Implementation

The ”Density-Based Spatial Clustering of Applications with Noise” (DBSCAN) algorithm is clustering method that identifies clusters based on density connectivity. DBSCAN does not require specifying the number of clusters beforehand, instead, it groups points based on their proximity and density.

Lets explain our code implementation.

As initialization, all data points are initially marked as noise (-1). The algorithm begins with a cluster ID of 0 and dynamically increments it as new clusters are identified.

Then we expand our clusters, starting from a randomly selected point, the algorithm checks its neighbors within a radius (eps). If the number of neighbors meets or exceeds the (min_samples) threshold, the point is marked as a core point, and the cluster is expanded to include its density-reachable neighbors.

Then with the help of our helper function inside our function, we compute the distances between a point and all other points in the dataset. Neighbors within the (eps) radius are returned for further processing.

The algorithm iterates over each point, forming clusters dynamically without requiring the number of clusters as input. Points that fail to meet the density criteria are retained as noise (-1).

4. RESULTS

4.1. Data Before Cleaning

The data had some noise problems and extra outlier photos. Following results are before the cleaning done with the DBSCAN. What they mean and their results will be discussed later with the cleaned data.

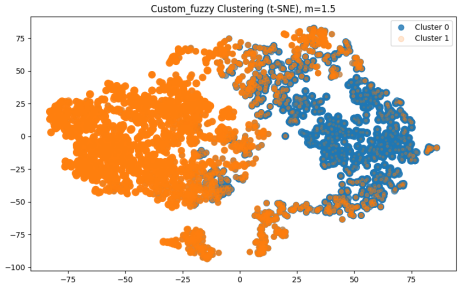

Figure 4.1: Fuzzy Clustering (Custom version) before cleaning. (With fuzziness metric: 1.5)

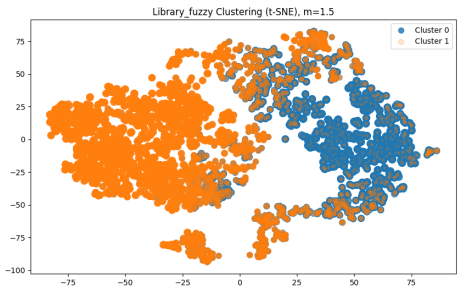

Figure 4.2: Fuzzy Clustering (Library version) before cleaning. (With fuzziness metric: 1.5)

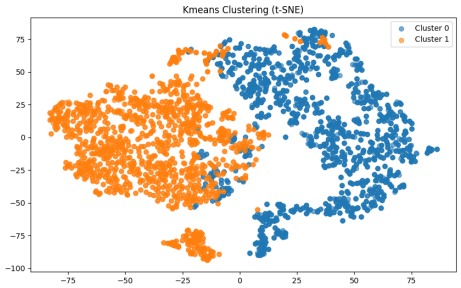

Figure 4.3: K-Means Clustering before cleaning.

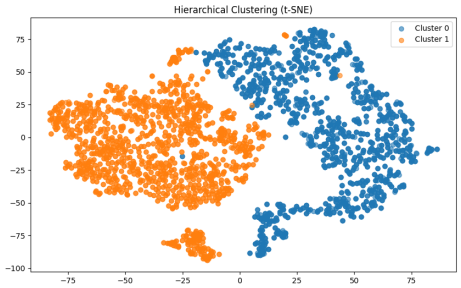

Figure 4.4: Hierarchical Clustering before cleaning.

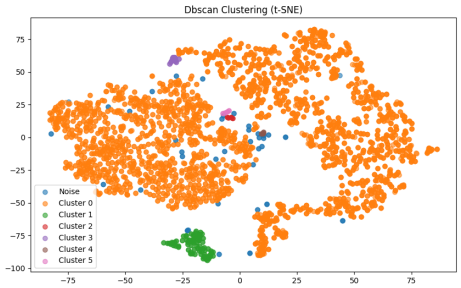

Figure 4.5: DBSCAN Clustering before cleaning.

4.2. After Cleaning

As we can see it from the DBSCAN graph before cleaning, there is main cluster and some outlier clusters and data-points in our uncleaned dataset. With this result, we have decided to clean our dataset with this method.

Here is the DBSCAN after that cleaning which enabled us to see result of our cleaning.

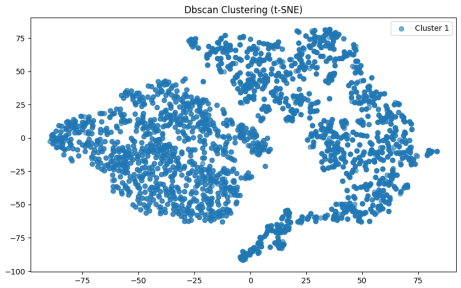

Figure 4.6: DBSCAN Clustering after cleaning with that method.

The rest of the results are made with dataset which are cleaned.

4.3. Non-Exclusive Results

We have used one method for our non-exclusive method as Fuzzy C-Means but we have tested it with two different versions which is our custom method and other one is library function of this method. We have done this implementation to customize this method better and test it with comparing its success to a library method.

Fuzzy parameter (m) which represents the fuzziness has been tested with 4 different values as 1.2, 1.5, 2.0 and 2.5. This metric can be used to balance what is healthy and what is molded for our leaves and it can be adjusted with some professional. As according to graphics, most reasonable results has been achieved between 1.2 and 1.5.

With figures we have saw that our custom clustering method is success as compared to the library one because they are very close to each other. In addition when we decrease the tolerance (tol) and increase the runtime of our model, we had more accurate results.

The following figures shows Fuzzy C-Means runs for both versions with those metric values.

4.3.1. Custom Runs:

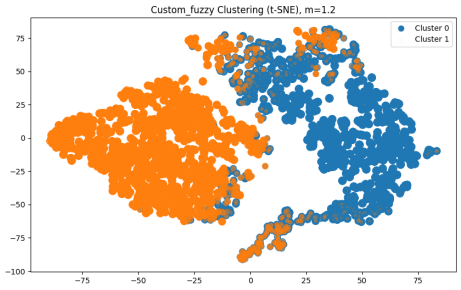

Figure 4.7: Fuzzy Clustering (Custom version). (With fuzziness metric: 1.2)

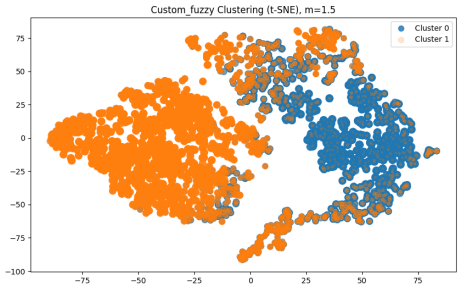

Figure 4.8: Fuzzy Clustering (Custom version). (With fuzziness metric: 1.5)

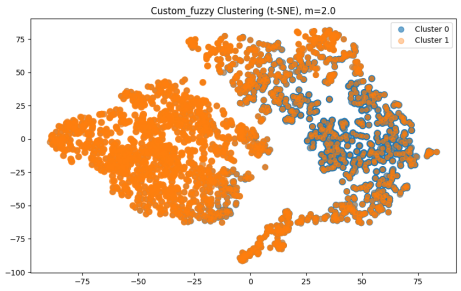

Figure 4.9: Fuzzy Clustering (Custom version). (With fuzziness metric: 2.0)

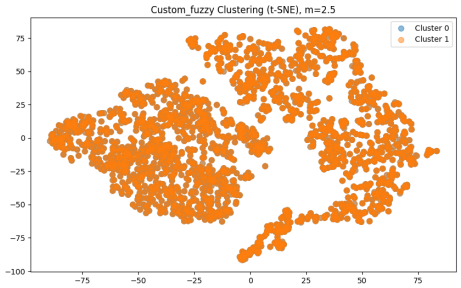

Figure 4.10: Fuzzy Clustering (Custom version). (With fuzziness metric: 2.5)

4.3.2. Library Runs:

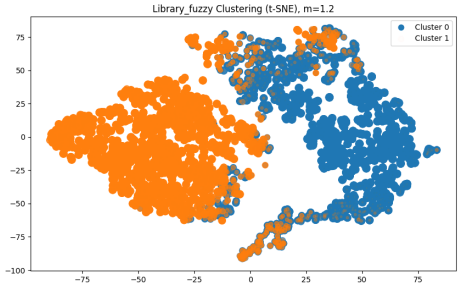

Figure 4.11: Fuzzy Clustering (Library version). (With fuzziness metric: 1.2)

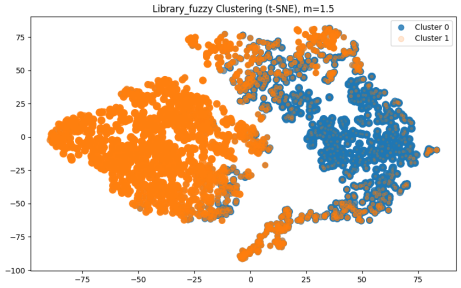

Figure 4.12: Fuzzy Clustering (Library version). (With fuzziness metric: 1.5)

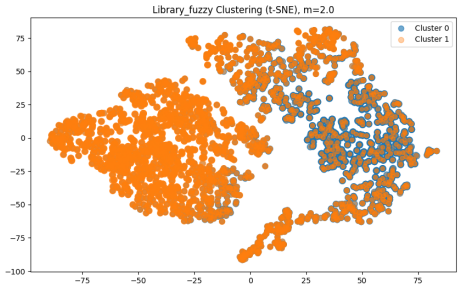

Figure 4.13: Fuzzy Clustering (Library version). (With fuzziness metric: 2.0)

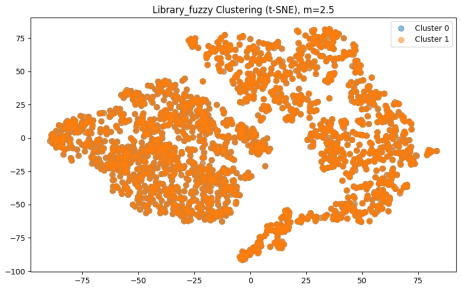

Figure 4.14: Fuzzy Clustering (Library version). (With fuzziness metric: 2.5)

4.4. Exclusive Methods

Exclusive methods has been used to show the results of those methods which very strictly divides the data and makes us lose the information of the leaves between healthy and molded condition.

DBSCAN, Hierarchical and k-Means methods has been used to compare them with each other and with our non-exclusive method. DBSCAN used as custom implementation. We used this method to clean the data so it just show the accuracy of the cleaning.

Those results are shown in following figures.

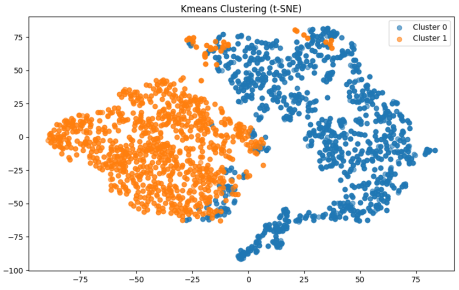

Figure 4.15: k-Means Clustering

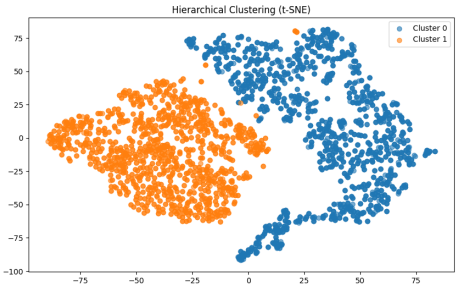

Figure 4.16: Hierarchical Clustering

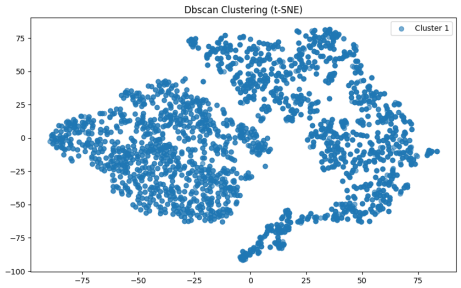

Figure 4.17: DBSCAN Clustering

4.5. Results as ARI Scores for All Clustering Methods

As ARI score shows, Our library and fuzzy ARI scores are close to each other and changes with every run which are very variant but as generally they are really close to each other. This score changes from 0.5 to 0.6 with both of our library and custom methods which means for some data points, those models separates them very sure of itself, but for some leaves, model is unsure which is what we are trying to achieve and see.

In other hand, exclusive methods achieved very high confidence. (Except the DBSCAN which is just here to show the result of cleaning.) This means we have lost some leaves in between molded and healthy leaves.

```
Evaluating clustering results...
Library Fuzzy Clustering - ARI: 0.69
Custom Fuzzy Clustering - ARI: 0.60
Exclusive Clustering (DBSCAN) - ARI: 0.00
Exclusive Clustering (KMeans) - ARI: 0.79
Exclusive Clustering (Hierarchical) - ARI: 0.98

All results saved in 'results/' folder.
(cherry_env) PS C:\Users\ozand\Desktop\dataproje\code>
```

Figure 4.18: ARI Scores of Clustering Methods

4.6. Example of Leaves with different confidence on Custom Fuzzy Algorithm (with fuzziness metric as 1.5)

4.6.1. High (more then %90) Confidence

Healthy and Molded leaves are separated successfully as we can see on the following figures.

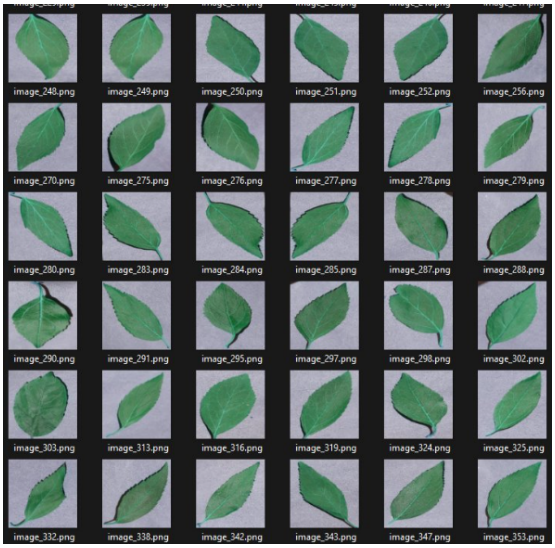

Figure 4.19: Healthy Labeled Leaves on High Confidence.

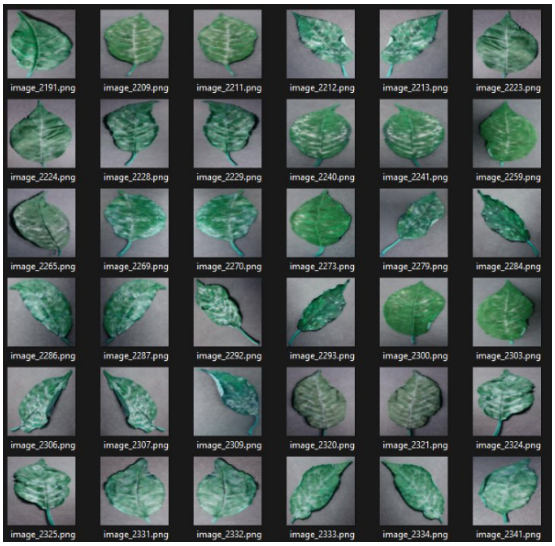

Figure 4.20: Molded Labeled Leaves on High Confidence.

4.6.2. Mid (between %60 and %40) Confidence

Healthy and Molded leaves are not strictly separated as we can see on the following figures which has both of them mixed up with each other.

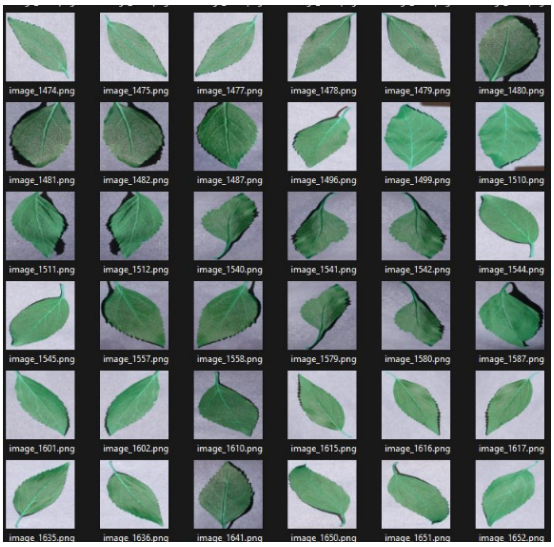

Figure 4.21: Healthy Labeled Leaves on Mid Confidence.

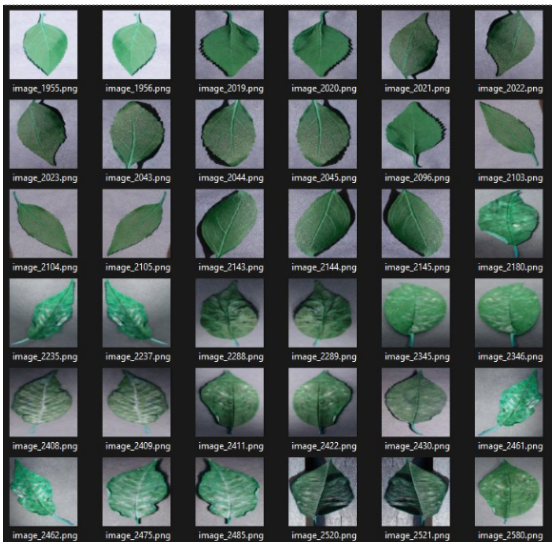

Figure 4.22: Molded Labeled Leaves on Mid Confidence.

5. CONCLUSIONS

Throughout this project, we explored both exclusive and non-exclusive clustering methods to analyze and classify cherry leaf data affected by mold. With fuzzy c-means clustering, we observed how non-exclusive methods can provide deeper insights into datasets without clean-cut boundaries, such as leaves that might partially has mold symptoms. The soft membership assignment allowed us to capture the uncertainty in the dataset and enabled us to filter and analyze data points based on confidence thresholds.

On the other hand, we tested the exclusive clustering methods, such as DBSCAN, k-means, and hierarchical clustering, which provided different results and helped us to cluster assignments and identifying noise.

The comparison of ARI (Adjusted Rand Index) scores demonstrated the differences in how these methods classify data and handle overlap or ambiguity. While non-exclusive methods was good at capturing uncertainty, exclusive clustering provided clear boundaries for different clusters.

With using DBSCAN to clean noisy data, model had improved the quality of results in methods. Additionally, with the visualizations of membership thresholds (e.g., healthy 90%, mold 90%, and mid-range 60%-40% points) we have seen the importance of the non-exclusive clustering for real life datasets with different levels of certainty.

Finally, this project showed the importance of non-exclusive clustering techniques for analyzing mid-data and handling real-world examples, particularly in datasets like our leaves dataset, where clear borders are hard to produce for some values.

BIBLIOGRAPHY

[1] Ahmad Habib Hasan Zein. “Cherry-leaves dataset.” (2025), [Online]. Available: https://www.kaggle.com/datasets/ahmadhabibhasanzein/cherry-leaves/data (visited on 01/18/2025).

[2] JAMES C. BEZDEK, ROBERT EHRLICH, WILLIAM FULL. “Fcm: The fuzzy c-means clustering algorithm.” (2025), [Online]. Available: https://staff.fmi.uvt.ro/~daniela.zaharie/dm2019/EN/projects/biblio/FuzzyCMeans/FCM%20-%20The%20Fuzzy%20c-Means%20Clustering%20Algorithm.pdf (visited on 01/18/2025).

[3] Scikit. “Scikit-learn.” (2025), [Online]. Available: https://scikit-learn.org/ (visited on 01/19/2025).

[4] matplotlib. “Matplotlib.” (2025), [Online]. Available: https://matplotlib.org/ (visited on 01/19/2025).

[5] PyTorch. “Pytorch.” (2025), [Online]. Available: https://pytorch.org/ (visited on 01/19/2025).

[6] DBSCAN. “Dbscan.” (2025), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html (visited on 01/19/2025).

[7] scipy.org. “Hierarchical clustering functions and tools provided in scipy.” (2025), [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.linkage.html (visited on 01/19/2025).

[8] Scikit. “Scikit-learn documentation for k-means.” (2025), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html (visited on 01/19/2025).

[1]–[8]