Apple Disease Detection and Classification using Random Forest (One-vs-All)

Zengming Wen[1], Hong Lan[1], Muhammad Asim Khan[1,*]

[1]School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi, 341000, China.

Abstract

Fruit diseases detection and recognition are a common problem worldwide. Fruit Disease detection is a hot topic among researchers and is very complex due to structure and color similarity factors. In this research we proposed a new model to detect and classify the apple disease with the help of digital image processing and machine learning. First, image processing techniques were applied to enhance the image contrast and remove the noise, which helped to segment the region of interest accurately also help to extract feature without garbage data. Then K-means clustering technique with fuzzy C-mean method was implemented to segment the images. GLCM feature extraction was used after the segmentation. Real images of apple disease with multi-disease regions were used in the research method. These features were preprocessed with methods like LDA. K-Fold cross validation was used for training and testing, with combination of random forest machine learning method. The result showed high accuracy with comparison of existing techniques.

Keywords: Fruit diseases detection, digital image processing, GLCM feature extraction, LDA, random forest.

Received on 04 12 2024; accepted on 13 01 2025; published on 22 01 2025

doi:10.4108/airo.8041

1. Introduction

Originating in Central Asia and belonging to the Rosaceae family (like roses and cherries), the apple (Malus domestica) has been a dietary staple for millennia. Cultivated worldwide in temperate zones, it's one of the most widely available fruits. Thousands of varieties exist, categorized for eating fresh or various culinary uses like cider, juice, jams, and compotes [1]. The apple is a significant source of nutrients, rich in carbohydrates, vitamins, and minerals including sugars, organic acids, pectin, and dietary fiber. This composition contributes to its well-rounded nutritional profile. While traditionally valued for its potential health benefits, modern evidence-based medicine has confirmed its positive impact on human health [2].

Traditionally, the fruit industry relied on a method of disease detection and identification that centered around visual inspection by experts. This process necessitated the deployment of technical specialists to fields, where they would meticulously examine the outward appearance of the fruit to diagnose specific disease types. However, now considered outdated, this approach suffers from two significant limitations time-consuming and economically costly compare to our proposed method. Furthermore, inconsistencies in the level of expertise among agricultural technicians can introduce inaccuracies into the diagnoses, potentially leading to missed or misidentified diseases [3]. As a result, there is a growing need to transition towards automated detection methods. These methods leverage the power of cutting-edge technologies like computer vision and deep learning, which have demonstrably achieved exceptional accuracy and impressive efficiency in disease identification [4]. The fruit industry has already begun to embrace this technology, and its application has resulted in significant advancements in both the overall quality of fruits and the productivity of growers [5].

This study focuses on the following objective:

- Developing a method to sort apples automatically. The process utilizes digital image processing

[*]Corresponding author. Email: m.asimkhattak@gmail.com

techniques to identify and separate rotten and deteriorated apples from healthy ones.

- The following three diseases are considered for this proposed method. Because those three diseases are the most common also the shape and color of the affected regions are confusing for the formers to identify and cure in advance :[29]
  - Apple scab: The most widespread thrives in cool, wet springs and infects leaves, fruits, and flowers, reducing yield and fruit quality[6].
  - Apple rot: Apple rot, caused by the fungus Botryosphaeria obtusa, attacks leaves, stems, and fruits, with initial purple spots developing on leaves that later expand and turn brown[6].
  - Apple blotch: Apple blotch, caused by two fungal culprits, results in dark greenish-blue blemishes on the fruit surface, impacting both economic value and marketability [6].
- Image segmentation with complex background.
- Classification of 3-disease.

The structure of this paper is outlined as follows: Section 2 presents a literature review. Section 3 introduces data processing techniques, feature extraction techniques and other methods used. Section 4 discusses the classification results and analysis. Finally, Section 5 presents the conclusions.

2. Literature Review

Traditionally, identifying apple diseases was a time-consuming and error-prone task. However, the field is embracing a new era of efficiency and accuracy with the rise of image-based detection methods powered by cutting-edge computer vision and machine learning technologies[29].

In the realm of image recognition, extracting features is an essential step. These features, like color, texture, and shape, help computers understand the image. Some technique are proposed that leverage K-Means clustering and feature fusion to detect apple diseases. This approach, combining color and texture information, significantly boosts the accuracy about 80 percent of classification [7]. An another method based on gray-level co-occurrence matrix (GLCM) features, the limitation of this paper is they use 4 feature from GLCM (contrast, correlation, energy and homogeneity) while the accuracy drop to 50 percent with the complex background [8].

The field of fruit disease classification utilizes a variety of techniques, including support vector machines (SVMs), decision trees, random forests, and neural networks. One such example is the method for identifying apple diseases. This approach, based on HOG features and bagged decision trees, effectively categorizes apples as healthy or defective with an impressive accuracy of 96% [9]. Also a new technique achieve 90% of accuracy by using multiclass SVM and fuzzy logic for citrus dataset [10].

An improved YOLOv3 algorithm is used for accurate apple detection in natural environments, achieving an F1 value above 91.1% under varying lighting conditions [11]. An another method introduced a tomato disease detection method utilizing convolutional neural networks (CNNs). This method leverages data augmentation techniques to significantly boost classification accuracy [12].

Using image analysis, researchers have proposed various techniques for detecting apple and grape leaf diseases. A researcher introduced introduced a fusion model combining CNN and LSTM networks specifically for apple disease classification, achieving a remarkable 99.02% accuracy in identifying initial disease severity [13]. Leveraged transfer learning with models like ResNet and MobileNet to achieve high accuracy in apple disease detection using a standard image dataset [14]. For apple leaf disease detection, combines ant colony optimization with CNNs, reaching a performance of 98.5% on the ALDD database [15]. In one new research in the same era, a fine-tuned VGG-16 network was used on apple and grape leaf diseases and achieve 97.87% accuracy [16].

3. Proposed Architecture

The proposed system is designed to differentiate between healthy and diseased apples. A dataset containing various apple disease categories, including those with apple scab, apple rot and apple blotch, was compiled from Kaggle data-source [35]. To initiate the process, image acquisition is completed. Initially, the gathered material is transformed from RGB color space to Lab color space. These image processing techniques are applied to enhance contrast and remove noise, ultimately aiding in the segmentation of the region of interest. Subsequently, the images are segmented using the K-means clustering technique with fuzzy C-mean method and auto thresholding. Grey Level Co-occurrence Matrix (GLCM) features are then extracted to obtain the desired information. These features are preprocessed using the Linear Discriminant Analysis (LDA) method. Finally, a random forest classifier is implemented, following the one-against-all approach, in conjunction with K-Fold cross validation for test dataset classification.

The structural diagram of our work is shown in Fig 1.

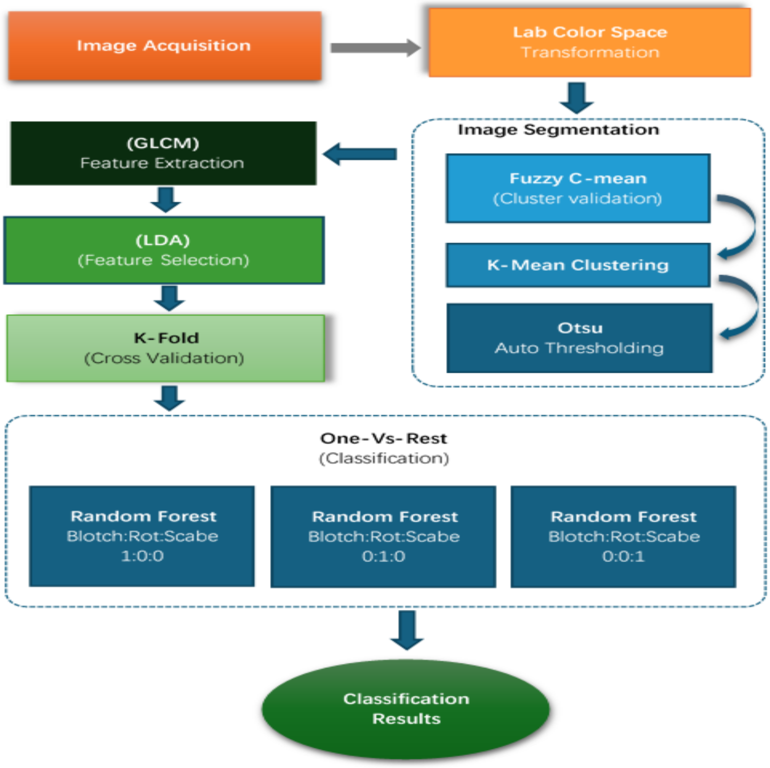
Image Acquisition
Lab Color Space Transformation
Image Segmentation
Fuzzy C-mean (Cluster validation)
K-Mean Clustering
Otsu Auto Thresholding
(GLCM) Feature Extraction
(LDA) (Feature Selection)
K-Fold (Cross Validation)
One-Vs-Rest (Classification)
Random Forest Blotch:Rot:Scabe 1:0:0
Random Forest Blotch:Rot:Scabe 0:1:0
Random Forest Blotch:Rot:Scabe 0:0:1
Classification Results

Figure 1. Structure diagrams with details steps of the proposed methodology.

3.1. Image Pre-processing

To prepare images for computer vision and machine learning, we use image pre-processing, a critical first step. This involves techniques like Lab color space conversion. These techniques address challenges like noise, brightness, contrast, image size, and color format. By applying these methods, we essentially refine the raw images into a format that makes analysis by machine learning models more effective [17].

Lab color space transformation. Lab color space transformation is a technique used in image pre-processing to improve image analysis. the visual results of RGB to Lab color space conversion are shown in the Figure 2 (b). It involves converting an image from its original color space (like RGB) into Lab space (Lab*) [17]. This process separates lightness information (L*) from color information (a* and b*), allowing independent adjustments of brightness and color. To achieve this, the conversion involves optional pre-processing of the RGB data, followed by transforming it into XYZ tristimulus values (representing perceived red, green, and blue light), and finally calculating the Lab* values based on a reference white point [18]. Here's what each component represents:

- L* (Lightness): Ranges from 0 (black) to 100 (white), representing the perceived lightness of a color.
- a*: Represents the green-red color component, with positive values leaning towards red and negative values towards green.
- b*: Represents the blue-yellow color component, with positive values shifting towards yellow and negative values towards blue.

The L* channel captures brightness and contrast, the a* and b* channels hold color and texture information, providing rich image details [19]. We prioritize the a* and b* channels for K-means clustering during image segmentation. Conversion of RGB color space to Lab are explained with the help of following equation:

- First convert RGB to XYZ:

$$XYZ = M \times RGB \tag{1}$$

  - M is a 3 × 3 matrix that depends on the specific RGB color space (e.g., sRGB, Adobe RGB).
  - RGB is a 3 × 1 column vector of RGB values (normalized to the range 0-1).

- In the next step we are performing the conversion of XYZ to CIE 1976 Luv:

$$L^{*} = 116 \times (\frac{Y}{Y_n})^{1/3} - 16 \tag{2}$$

$$u^* = 13 \times L \times (u - u_0) \tag{3}$$

$$v^* = 13 \times L \times (v - v_0) \tag{4}$$

Where Y_n is the reference white point luminance, u and v are the CIE 1931 chromaticity coordinates, u_0 and v_0 are the chromaticity coordinates of the reference white point.

- In the last, we are doing the conversion from CIE 1976 Luv to LAB*:

$$L = L^* \tag{5}$$

$$a = 500 \times (u^* - u_0^*) \tag{6}$$

$$b = 500 \times (v^* - v_0^*) \tag{7}$$

3.2. Image Segmentation

In computer vision, image segmentation acts like a digital scalpel, separating an image into its meaningful components. Techniques like Fuzzy C-means, K-means clustering, and auto thresholding can be used to achieve this. By dividing the image into regions with similar characteristics, like individual objects or groups of pixels, segmentation simplifies analysis. This allows us to isolate specific objects or regions of interest within the image.The segmented region results are shown in the Figure 2 (c) after applying fuzzy c-mean, K-Mean clustring with Otsu's auto threshold technique.

Fuzzy C-Means (FCM). Fuzzy C-Means (FCM) is a popular clustering algorithm that assigns each data point to multiple clusters with varying degrees of membership. While primarily used for clustering itself, it can also be adapted for cluster validation. The algorithm iteratively refines these membership degrees and cluster centers until they stabilize, leading to the formation of well-defined clusters [20].

In this method we calculate Partition Coefficient (PC). Which Measures the degree of fuzziness in the clustering. A value closer to 1 indicates more crisp clusters.

$$PC = \frac{\sum(\sum u_{ij})^n}{m \times n} \tag{8}$$

Where m is the fuzziness exponent (typically between 1 and 2), n is the number of data points and u_{ij} is the membership of data point i to cluster j.

In the next step we calculate Fuzzy Hypervolume (FH). Measures the volume occupied by the clusters in the data space. A smaller FH suggests more compact clusters.

$$FH = \sum(\sum u_{ij}^m \times \|x_i - v_j\|^2) \tag{9}$$

Where x_i is the i-th data point and v_j is the centroid of cluster j.

In the End, we calculate Entropy to Measure the uncertainty in the clustering. A lower entropy indicates more distinct clusters.

$$Entropy = \frac{\sum(\sum(u_{ij} \times \log u_{ij}))}{n \log c} \tag{10}$$

Where c is the number of clusters. This will be passed as an argument to the K-Mean Clustering algorithm for a dynamic number of clustering techniques because fix number of clustering values are the big drawback of K-mean clustering.

K-Means. K-Means clustering is a popular unsupervised learning algorithm that groups unlabeled data points into predefined clusters [21]. It iteratively assigns data points to the closest cluster based on distance metrics, recalculates cluster centers, and repeats until the clusters stabilize. This approach effectively uncovers hidden structures and patterns within the data by grouping similar data points together. This work leverages K-means clustering to analyze image data. The data, derived from the a* and b* channels of the Lab color space, is first reshaped into a two-dimensional matrix. The selection of the cluster number is informed by the data features extracted through the prior application of Fuzzy C-Means (FCM).

Otsu's Auto Threshold. In image processing, auto thresholding tackles the challenge of automatically separating an image into foreground (objects of interest) and background. It achieves this by analyzing the image's histogram, a graph depicting the frequency of each intensity level (brightness). By examining the histogram's shape, auto thresholding algorithms can identify valleys or dips that potentially separate foreground and background pixels [22]. This work leverages Otsu's method, a popular auto thresholding technique. Otsu's method iterates through all possible threshold values and calculates the between-class variance for each threshold. This variance measures the separation between foreground and background pixels in the histogram. Otsu's method ultimately selects the threshold that maximizes this variance, leading to a clear distinction between the two regions in the resulting binary image (where pixels are classified as either 0 or 1 based on the chosen threshold) [23].

3.3. Feature Extraction

In our work, we specifically use feature extraction to analyze images by extracting Gray-Level Co-occurrence Matrices (GLCMs) features. In our proposed methodology we use 13 Gray-Level Co-occurrence Matrices (GLCMs) feature for machine learning classification.

Gray-Level Co-occurrence Matrices (GLCMs).. In image processing, texture plays a crucial role in tasks like classification and segmentation. GLCM Features, derived from Gray-Level Co-occurrence Matrices (GLCMs), provide a powerful way to quantify this texture information [24]. From GLCM, various statistical features that quantify different aspects of the texture can be extracted [25]. The GLCM feature extraction steps are define in the following.

1. **Specifying Direction and Distance**: Choose the directions (denoted by θ) and how far apart (distance,d) wanted to examine pixels in the image. Popular directions include horizontal (straight across, 0 degrees), vertical (up and down, 90 degrees), and diagonal (either at a 45-degree or 135-degree angle).

2. **Building the Co-occurrence Matrix:** For each direction (θ) and distance (d) picked, a special matrix called a Gray Level Co-occurrence Matrix (GLCM) is created, written as $P(i, j)$. This matrix shows how often pairs of pixels with specific gray levels (i and j) appear next to each other in the chosen direction and at the chosen distance. Here's the formula used to calculate a specific value (i, j)within the GLCM.

$$\frac{of\ times(I(p,q) = i, I(p + d \times \cos(\theta), q + d \times \sin(\theta)) = j\ \ )}{Total\ number\ of\ valid\ pixel\ pair} \tag{11}$$

   In a grayscale image, each pixel's brightness is represented by its intensity value, denoted by $I(p, q)$ for the pixel at location (p, q). The gray level co-occurrence matrix (GLCM) analyzes how often pixels with specific intensities (i and j) appear next to each other in a chosen direction (like horizontal or diagonal) and at a specific distance (d). This frequency is captured by the number of times (i, j) in the GLCM. However, the total number of valid neighboring pixel pairs which can be considered depends on the image size itself and the distance (d) chosen between pixels.

3. Following the generation of the Gray Level Co-occurrence Matrices (GLCMs), a set of statistical features can be extracted to quantify various aspects of the image texture. These features provide numerical descriptors that capture the spatial distribution of gray levels within the image. Common examples of these features, along with their corresponding mathematical formulas, are presented in table 1:

To conduct a quantitative analysis of textural properties within our image dataset, thirteen informative features were extracted from the Gray Level Co-occurrence Matrix (GLCM) for each image. This extensive set of features encompassed characteristics such as contrast, homogeneity, and entropy, thereby comprehensively capturing the diverse textural variations exhibited within the images. The extracted features were subsequently meticulously organized and stored within a well-structured comma-separated values (CSV) table. This approach facilitated efficient analysis and exploration of the textural characteristics across the entirety of the dataset.

3.4. Feature Selection

Classification tasks are designed to assign data points to predefined categories. However, raw data frequently exhibits inconsistencies, contains missing information, or may not be presented in a format suitable for analysis by classification algorithms. Data pre-processing emerges as a critical step in addressing these challenges [26]. It refers to the process of transforming raw data into a cleansed, consistent, and well-structured format that is optimal for use with classification models. The fundamental purpose of data pre-processing is to elevate the quality and efficacy of the classification process. In our technique we are using LDA method for feature selection.

Linear Discriminant Analysis (LDA). Within data science, Linear Discriminant Analysis (LDA) offers a powerful and multifaceted solution [27]. It tackles high-dimensional data by projecting it onto a lower-dimensional space, prioritizing features that best distinguish categories. This makes LDA valuable for both dimensionality reduction and supervised classification. LDA excels at separating data points based on class labels, proving useful in pattern recognition (e.g., image classification) and feature selection for robust classification models. Additionally, its linear nature allows for some interpretability of the factors driving class separation. LDA demonstrates particular efficacy in supervised classification tasks [28]. The LDA contains following steps:

1. Mean Vector Calculation: For each class $c(c = 1, 2, ..., C)$, calculate the mean vector (μ_c):

$$\mu_c = \frac{1}{N_c} \sum_{i=1}^{N_c} x_i \tag{12}$$

   N_c is the number of data points in class c , and x_i represents the i-th data point.

2. Scatter Matrices:Compute the within-class scatter matrix (S_w):

$$S_w = \sum_{c=1}^{C} \sum_{i=1}^{N_c} (x_i - \mu_c)(x_i - \mu_c)^T \tag{13}$$

Table 1. GLCM Feature Extracted for the proposed technique with detail information and mathamatic equations

Name	Equation	Detail
Contrast	$\sum_i \sum_f ((i-j)^2 \times P(i,j))$	Measures the intensity variation between neighboring pixels.
Homogeneity	$\sum_i \sum_j (\frac{1}{1+(i-j)^2} \times P(i,j))$	Captures the uniformity of the texture.
Entropy	$-\sum_i \sum_j (P(i,j) \times \log_2 (P(i,j)+\epsilon))$	Represents the randomness or complexity of the texture. ϵ is a small value to avoid issues with log 0.
Energy	$\sum_i \sum_j P(i,j)^2$	Indicates the prevalence of repetitive patterns in the texture.
Correlation	$\sum_i \sum_j \frac{(i-\mu_i)\times(j-\mu_j)\times P(i,j)}{\sigma_i \times \sigma_j}$	Captures the linear dependency between neighboring pixels. μ_i and σ_i represent the mean and standard deviation of the i-th row of the GLCM, and similarly for μ_j and σ_j.
Mean	$\sum_i \sum_j (i \times j \times P(i,j))$	Summarizes the overall gray level intensity distribution for a specific direction and distance in the image.
Variance	$\sum_i \sum_j ((i \times j - Mean)^2 \times P(i,j))$	Reflects the texture's uniformity - high variance indicates more variation (less uniform), while low variance suggests a more consistent texture.
Standard Deviation (STD)	$\sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$	Quantifies how spread out the intensity variations (i,j) are around the average.
Smoothness	$1 - \frac{1}{1+a}$	Describes the uniformity of pixel intensities, with higher smoothness indicating a less textured and more consistent appearance. The value a represents the overall intensity of the image.
Root Mean Square (RMS)	$\sqrt{\sum_i \sum_j ((i-j)^2 \times P(i,j))}$	Measures the contrast variation within a specific direction in an image by calculating the average intensity difference between neighboring pixels.
Kurtosis	$\sum_i \sum_j \frac{(i-Mean)^4 \times P(i,j)}{STD^4} - 2$	Describes the shape of the distribution of intensity values within the GLCM, comparing it to a normal distribution (bell-shaped curve). By subtracting 2, the equation effectively centers the excess kurtosis around 0.
Skewness	$\sum_i \sum_j \frac{(i-Mean)^3 \times P(i,j)}{STD^3}$	Reveals the imbalance in the intensity distribution, favoring brighter or darker areas of the image.
Inverse Difference Moment (IDM)	$\sum_i \sum_j \frac{P(i,j)}{1+(i-j)^2}$	Quantifies local texture uniformity by favoring neighboring pixels with similar intensities.

$S_w$captures the variance of data points within each class. Compute the between-class scatter matrix (S_b):

$$S_b = \sum_{c=1}^{C} N_c(\mu_c - \mu)(\mu_c - \mu)^T \tag{14}$$

μ is the overall mean vector of all data points. S_b highlights the variance between the class means themselves.

3. Maximizing Class Separation: LDA seeks a set of linear transformations (w) that project the data onto a lower-dimensional subspace while maximizing the ratio of the determinant of S_b to the determinant of S_w:

$$J(w) = \frac{w^T S_b w}{w^T S_w w} \tag{15}$$

This ratio essentially measures the separation between classes in the projected space.

4. Eigenvalue Decomposition: To achieve the maximization goal, perform eigenvalue decomposition on $S_w^{-1} S_b$:

$$S_w^{-1} S_b = W \lambda W^T \tag{16}$$

W is a matrix containing the eigenvectors, and Λ is a diagonal matrix containing the corresponding eigenvalues (λ_i).

5. Selecting Discriminant Directions: The eigenvectors represent the directions of projection, and the eigenvalues represent the variance along those directions. Rank the eigenvectors based on their corresponding eigenvalues, with λ_1 being the largest. Select the top k eigenvectors with the highest eigenvalues (corresponding to the greatest separation between classes) to form the projection matrix (W_k).

6. Project the original data (x) onto the lower-dimensional subspace defined by W_k:

$$z = W_k^T x \tag{17}$$

   This results in a reduced-dimensionality representation (z) of the data that emphasizes class separation. Finally, a classification model can be built using this projected data (z) to classify new, unseen data points.

3.5. Classification

K-Fold Cross Validation. In proposed method we use K-Fold validation independently for testing and training data set. K-Fold Cross Validation (K-Fold CV) [29] is a powerful technique that addresses two crucial issues in machine learning: overfitting and unreliable performance estimates. It works by splitting the data into k equal-sized folds. Here's the magic: K-Fold CV iterates k times, each time using one fold for validation (testing) and the remaining k-1 folds for training. The integration of K-Fold Cross Validation (K-Fold CV) into the machine learning workflow facilitates a more confident understanding of a model's performance and generalization ability [30]. This ultimately results in the generation of more reliable and trustworthy results. In our proposed work, we employed K-Fold Cross Validation (K-Fold CV) and fine-tuned its parameters to achieve an optimal data split for model training and evaluation. This approach ensured us to leverage our data most effectively, leading to superior model performance and the best possible results.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \tag{18}$$

where: n, Total number of observations, yi The response value of the ith observation, f(xi) The predicted response value of the ith observation

Random Forest. Random Forest is an ensemble method that relies on building and combining predictions from multiple decision trees [31]. It relies on building and combining predictions from multiple decision trees. However, each decision tree within a random forest utilizes mathematical concepts to make decisions [32]. Random forest achieves this diversity by employing randomness during training. For each tree, a random subset of features from the entire feature set is chosen. This prevents any single tree from relying too heavily on a specific feature, reducing the risk of overfitting. Additionally, data points are sampled with replacement. This means a data point can be selected multiple times for inclusion in a single tree, further enhancing the diversity of the forest. Each decision tree in the forest goes through a standard decision tree training process using its unique subset of features and data points. Random forests can effectively handle missing data points within the dataset. The randomness injected during training helps prevent the model from overfitting on the training data, leading to better performance on unseen data. Essentially, random forest aggregates the predictions from multiple, diverse decision trees, reducing variance and achieving a more robust and accurate outcome compared to a single decision tree.

We employed a single-disease focus for each iteration of our random forest classification process. After feature preparation, we trained the model multiple times. This iterative approach enabled us to pinpoint the configuration that achieved the optimal classification accuracy for the specific disease under investigation.

One-vs-Rest (OvR). One-vs-Rest (OvR), also known as One-vs-All (OvA), is a technique employed in machine learning to address multi-class classification problems using binary classification algorithms [33]. The core concept involves converting the original multi-class classification problem (where each data point is assigned to one of several classes) into a series of binary classification problems. With C classes present, C separate binary classification models are trained by One-vs-Rest, one for each class. All data points belonging to that specific class are labelled as positive examples within each model. All data points from the remaining $(C-1)$ classes are grouped together and labelled as negative examples. When presented with a new, unseen data point, each of the C trained models predicts a probability score indicating the likelihood of the point belonging to its corresponding class. The class associated with the highest predicted probability score from any of the models is assigned to the new data point. In essence, One-vs-Rest decomposes a complex multi-class problem into a sequence of simpler binary classification problems.

When dealing with Multi classes, OvR can be computationally efficient [34].

The idea of One-vs-Rest is implemented in our proposed work to perform the classification task. By understanding the method's strengths and weaknesses, effective implementation can be achieved to attain the desired results.

4. Results and Analysis

4.1. Dataset

This study utilizes a dataset of apple imagery curated from Kaggle, Apple Diseases Image Dataset Dataset[35]. The dataset encompasses approximately 500 images, ranging in dimension from 100 × 100 to 1920 × 1280 pixels. These images specifically depict three distinct apple disease classifications: A pple Rot, Apple Scab, and Apple Blotch.

4.2. Result

Figure 2 shows the visual result of image segmentation. In our research, we implemented three classification method, including Random Forest, Decision Tree and Support Vector Machine (SVM) to make a comparison. The table 2 and figure 3 is the result we get.

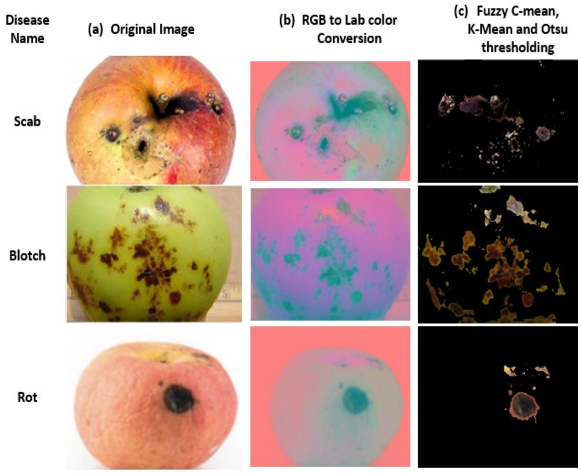
Disease Name
(a) Original Image
(b) RGB to Lab color Conversion
(c) Fuzzy C-mean, K-Mean and Otsu thresholding
Scab
Blotch
Rot

Figure 2. Image segmentation results of 3 target disease with original image and after segmentation image using digital image processing with K-means and Fuzzy C-means, the results show optimal accuracy of segmentation.

Table 2. Accuracy of three classification methods

Splits	Decision Tree	SVM	Random Forest
11	0.81081	0.91892	0.89189
12	0.82353	0.91176	0.88235
13	0.86667	0.90323	0.90323
14	0.85714	0.89655	0.89655
15	0.88889	0.88889	0.92593
16	0.84	0.92	0.88
17	0.86957	0.91667	0.91667
18	0.86957	0.91304	0.91304
19	0.90476	0.94238	0.95248

Three machine learning algorithms that are used in the suggested research with the highest accuracy are compared in table 2. The findings show that the random forest algorithm outperforms SVM and decision trees by a significant margin.

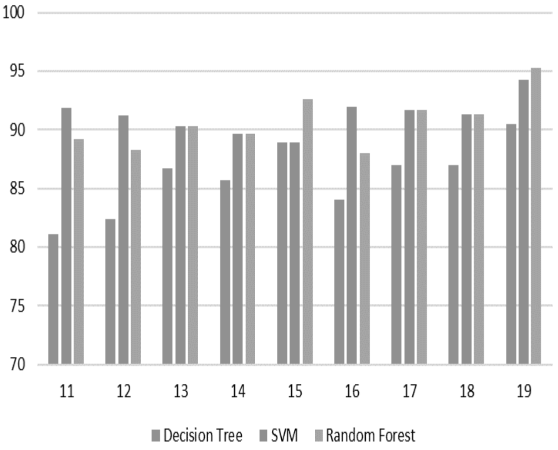

Figure 3. Detailed graph of accuracy comparison, using multiple machine learning technique in proposed method.

Figure 3 is a line graph comparing the accuracy of three classification methods: Support Vector Machine (SVM), Random Forest, and Decision Tree. The x-axis represents the number of splits, and the y-axis represents accuracy. The red line with circles represents the SVM method. It starts with the highest accuracy and increases steadily until around 14 splits. After that, the accuracy starts to decrease and fluctuate more. The blue line with triangles represents the Random Forest method. It starts with a lower accuracy than SVM but increases steadily throughout the measured range. By around 18 splits, it surpasses the accuracy of SVM. The black line with squares represents the Decision Tree method. It starts with the lowest accuracy but increases the most rapidly at first. However, it also shows the most significant fluctuations throughout the measured range.

Overall, the graph suggests that Random Forest offers the most stable and accurate performance across a higher number of splits. SVM performs well up to a certain number of splits but suffers from accuracy drops for more complex scenarios. Decision Tree offers the fastest initial gains in accuracy but can be unstable. The data plotted in the graph reaches a peak accuracy of 0.95248, achieved with 19 splits.

4.3. Comparison with other works

We compared our model's apple disease accuracy to previous studies. Table 3 and Figure 4 shows the comparison result.

The proposed method in this paper achieves a classification accuracy of 95.24%, which is the highest accuracy among the methods shown.

Table 3. Accuracy Comparison: Proposed Approach vs. Existing Methods.

Ref#	Method	Accuracy
0	**Proposed work**	**95.24%**
1	Behera, S. K. et al.[10] (multiclass SVM & fuzzy logic)	90%
2	Kamala, K. L. et al.[40] (SVC)	94.11%
3	Tiwari, R. et al.[36] (K-Means Clustering)	93.3%
4	Doh, B. et al.[37] (SVM)	93.12%
5	Habib, M. T. et al.[38] (SVM)	90.15%
6	Hossein A, et al.[29] (ANN)	73.0%
7	Sharif, M. et al.[39] (M-SVM)	90.4%
8	Dubey, S. R. et al.[41] (SVM)	93%

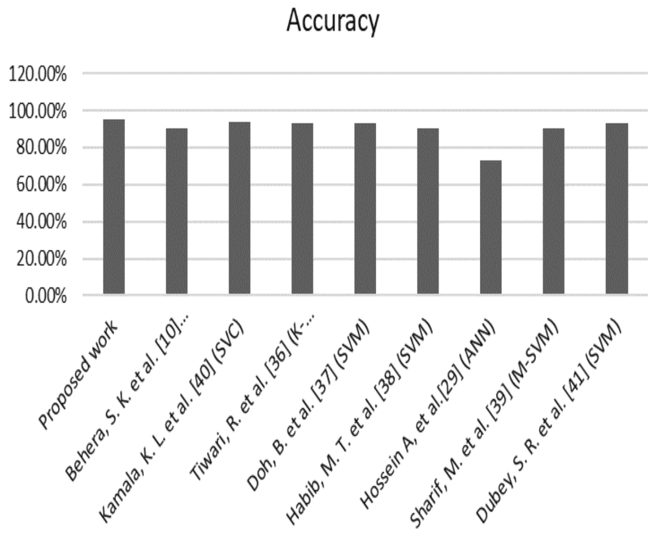

Figure 4. The graph illustrated accuracy results of our algorithm with highest accuracy comparing with existing methods proposed by various researcher.

5. Conclusion and Future Work

In conclusion, this research presented a novel and highly accurate model for classifying apple diseases using real-world images containing multiple disease regions. The proposed method incorporates several key steps: image pre-processing to enhance contrast and remove noise, segmentation using K-means clustering with fuzzy C-means, feature extraction with Grey Level Co-occurrence Matrix (GLCM), and feature preprocessing with techniques like LDA. Finally, a combination of K-Fold cross validation and a random forest machine learning method achieved high accuracy in disease classification compared to existing techniques. This approach offers a promising solution for early and precise detection of apple diseases, potentially improving crop management and reducing agricultural losses.

Therefore, with results exceeding 95%, we recognize the potential for further refinement. To this end, we'll be transitioning to deep learning techniques and leveraging the power of neural networks in future iterations, aiming to push the boundaries of performance even further.

References

[1] Patocka, J., Bhardwaj, K., Klimova, B., Nepovimova, E., Wu, Q., Landi, M., ... & Wu, W. (2020). Malus domestica: A review on nutritional features, chemical composition, traditional and medicinal value. Plants, 9(11), 1408.

[2] Boyer, J., & Liu, R. H. (2004). Apple phytochemicals and their health benefits. Nutrition journal, 3, 1-15.

[3] Song, X., & Mariano, V. Y. (2023, January). Image-based apple disease detection based on residual neural network and transfer learning. In 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA) (pp. 365-369). IEEE.

[4] Alharbi, A. G., & Arif, M. (2020, October). Detection and classification of apple diseases using convolutional neural networks. In 2020 2nd international conference on computer and information sciences (ICCIS) (pp. 1-6). IEEE.

[5] Ma, B., Hua, Z., Wen, Y., Deng, H., Zhao, Y., Pu, L., & Song, H. (2024). Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments. Artificial Intelligence in Agriculture, 11, 70-82.

[6] Awate, A., Deshmankar, D., Amrutkar, G., Bagul, U., & Sonavane, S. (2015, October). Fruit disease detection using color, texture analysis and ANN. In 2015 international conference on green computing and internet of things (ICGCIoT) (pp. 970-975). IEEE.

[7] Samajpati, B. J., & Degadwala, S. D. (2016, April). Hybrid approach for apple fruit diseases detection and classification using random forest classifier. In 2016 International conference on communication and signal processing (ICCSP) (pp. 1015-1019). IEEE.

[8] Sugiarti, Y., Supriyatna, A., Carolina, I., Amin, R., & Yani, A. (2021, September). Model Naive Bayes classifiers for detection apple diseases. In 2021 9th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-4). IEEE.

[9] Sujatha, P. K., Sandhya, J., Chaitanya, J. S., & Subashini, R. (2018, December). Enhancement of segmentation and feature fusion for apple disease classification. In 2018 Tenth International Conference on Advanced Computing (ICoAC) (pp. 175-181). IEEE.

[10] Behera, S. K., Jena, L., Rath, A. K., & Sethy, P. K. (2018, April). Disease classification and grading of orange using

machine learning and fuzzy logic. In 2018 International Conference on Communication and Signal Processing (ICCSP) (pp. 0678-0682). IEEE.

[11] Xuan, G., Gao, C., Shao, Y., Zhang, M., Wang, Y., Zhong, J., ... & Peng, H. (2020). Apple detection in natural environment using deep learning algorithms. IEEE Access, 8, 216772-216780.

[12] Nagesh, A. S., & Balaji, G. N. (2022, December). Deep learning approach for recognition and classification of tomato fruit diseases. In 2022 International Conference on Data Science, Agents & Artificial Intelligence (ICD-SAAI) (Vol. 1, pp. 1-6). IEEE.

[13] Sharma, R., Kukreja, V., Sood, P., & Bhattacharjee, A. (2023, May). Classifying the Severity of Apple Black Rot Disease with Deep Learning: A Dual CNN and LSTM Approach. In 2023 3rd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS) (pp. 173-177). IEEE.

[14] Agrawal, P., Singh, N., Bansal, N., & Goel, A. (2023, May). Accurate Disease Detection in Apple Fruit Images using Transfer Learning: A Novel Approach for Agricultural Sustainability. In 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC) (pp. 482-488). IEEE.

[15] Kaur, A., & Chadha, R. (2023, March). An Optimized Ant Gardient Convolutional Neural Network for Disease Detection in Apple Leaves. In 2023 2nd International Conference for Innovation in Technology (INOCON) (pp. 1-8). IEEE.

[16] Nagaraju, Y., Swetha, S., & Stalin, S. (2020, December). Apple and grape leaf diseases classification using transfer learning via fine-tuned classifier. In 2020 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT) (pp. 1-6). IEEE.

[17] Thwaites, A., Wingfield, C., Wieser, E., Soltan, A., Marslen-Wilson, W. D., & Nimmo-Smith, I. (2018). Entrainment to the CIECAM02 and CIELAB colour appearance models in the human cortex. Vision research, 145, 1-10.

[18] Lin, P., Xiaoli, L., Li, D., Jiang, S., Zou, Z., Lu, Q., & Chen, Y. (2019). Rapidly and exactly determining postharvest dry soybean seed quality based on machine vision technology. Scientific Reports, 9(1), 17143.

[19] Ng, K., & Song, T. (2024). Hybrid Quantum-Classical Neural Network for LAB Color Space Image Classification. arXiv preprint arXiv:2406.02229.

[20] Zhang, H., & Liu, J. (2022). Fuzzy c-means clustering algorithm with deformable spatial information for image segmentation. Multimedia Tools and Applications, 81(8), 11239-11258.

[21] Coates, A., & Ng, A. Y. (2012). Learning feature representations with k-means. In Neural Networks: Tricks of the Trade: Second Edition (pp. 561-580). Berlin, Heidelberg: Springer Berlin Heidelberg.

[22] Xu, X., Xu, S., Jin, L., & Song, E. (2011). Characteristic analysis of Otsu threshold and its applications. Pattern recognition letters, 32(7), 956-961.

[23] Garousi, V., Joy, N., & Keleş, A. B. (2024). AI-powered test automation tools: A sys-tematic review and empirical evaluation. arXiv preprint arXiv:2409.00411.

[24] Yousefi, J. (2011). Image binarization using Otsu thresholding algorithm. Ontario, Canada: University of Guelph, 10.

[25] Ranjitha, K. V., & Pushphavathi, T. P. (2024). Analysis on Improved Gaussian-Wiener filtering technique and GLCM based Feature Extraction for Breast Cancer Diagnosis. Procedia Computer Science, 235, 2857-2866.

[26] Zubair, A. R., & Alo, O. A. (2024). Grey level co-occurrence matrix (GLCM) based second order statistics for image texture analysis. arXiv preprint arXiv:2403.04038.

[27] García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: methods and prospects. Big data analytics, 1, 1-22.

[28] Li, C. N., Shao, Y. H., Wang, Z., Deng, N. Y., & Yang, Z. M. (2019). Robust Bhattacharyya bound linear discriminant analysis through an adaptive algorithm. Knowledge-Based Systems, 183, 104858.

[29] Hossein Azgomi, Fatemeh Roshannia Haredasht and Mohammad Reza Safari Motlagh. (2023). Diagnosis of some apple fruit diseases by using image processing and artificial neural network .https://doi.org/10.1016/ j.foodcont.2022.109484.

[30] Gorriz, J. M., Segovia, F., Ramirez, J., Ortiz, A., & Suckling, J. (2024). Is K-fold cross validation the best model selection method for Machine Learning?. arXiv preprint arXiv:2401.16407.

[31] Sara Alqethami,Badriah Almtanni ,Walla Alzhrani ,Manal Alghamdi (2022, April). Disease Detection in Apple Leaves Using Image Processing Techniques(Vol. 12, No. 8335-8341). ETASR.

[32] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[33] Vural, V., Fung, G., Rosales, R., & Dy, J. G. (2009, June). Multi-Class Classifiers and their Underlying Shared Structure. In IJCAI (pp. 1267-1272).

[34] Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. The Journal of Machine Learning Research, 5, 101-141.

[35] Kaggle. Apple Diseases Image Dataset Dataset.https://www.kaggle.com/datasets/davidhilton/apple-diseases-image-dataset

[36] Tiwari R, Chahande M. Apple Fruit Disease Detection and Classification Using K-Means Clustering Method. In: Das S, Mohanty MN, editors. Advances in Intelligent Computing and Communication. Singapore: Springer Singapore; 2021. pp. 71–84.

[37] Doh B, Zhang D, Shen Y, Hussain F, Doh RF, et al. Automatic Citrus Fruit Disease Detection by Phenotyping Using Machine Learning. In: 2019 25th International Conference on Automation and Computing (ICAC). Lancaster, United Kingdom: IEEE; 2019. pp. 1–5.

[38] Habib MT, Majumder A, Jakaria AZM, Akter M, Uddin MS, et al.Machine vision based papaya disease recognition.Journal of King Saud University - Computer and Information Sciences 2020 Mar;32:300–309.

[39] Sharif M, Khan MA, Iqbal Z, Azam MF, Lali MIU, et al. Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection.Computers and Electronics in Agriculture 2018 Jul;150:220–34.

[40] Lisha Kamala K, Anna Alex S. Apple Fruit Disease Detection for Hydroponic plants using Leading edge Technology Machine Learning and Image Processing. In: 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC). Trichy, India: IEEE; 2021. pp. 820–25.

[41] Dubey SR, Jalal AS. Detection and Classification of Apple Fruit Diseases Using Complete Local Binary Patterns. In: 2012 Third International Conference on Computer and Communication Technology. Allahabad, Uttar Pradesh, India: IEEE; 2012. pp. 346–51