Feature Engineering Approaches to Plant Leaf Disease Classification Utilising Computer Vision and Machine Learning

Ho An Loc, Nguyen Bao Khang, Tu Nguyen Duc Cuong
FPT University, Da Nang, Viet Nam

July 1 2024

Abstract

Accurately identifying and managing leaf diseases in plants is crucial for optimizing crop yield and sustainable agriculture. This study proposes a comprehensive approach employing machine learning, computer vision, and image processing techniques to classify plant leaf diseases. The methodology focuses on extracting discriminative features—color, texture, and shape—from leaf images using advanced algorithms to capture intricate disease indicators.

The Support Vector Machine (SVM) model proposed achieved exceptional results: an accuracy of 98.23%, precision of 98.23%, recall of 98.23%, and an F1 score of 98.22% through a thorough repeat hold-out validation process. These metrics underscore SVM's effectiveness in precisely classifying plant leaf diseases based on extracted image features. The research enhances disease detection efficiency and reliability, promoting sustainable agricultural practices through technological innovation.

1 Introduction

The identification and classification of plant leaf diseases have become critical tasks in modern agriculture, driven by the need to optimize crop yields and ensure sustainable farming practices. The emergence of this problem is rooted in the necessity to detect and manage diseases promptly, preventing their spread and mitigating adverse impacts on crops [KMJ+23]. Early disease detection enables timely intervention, safeguarding plant health and enhancing agricultural productivity. This proactive approach not only boosts crop yields but also minimizes negative environmental and health impacts associated with excessive pesticide use.

Traditional manual classification methods are often labor-intensive, costly, and less efficient. They are prone to human error and can result in significant financial losses due to delayed disease detection, leading to crop failure and increased pesticide costs. In contrast, automated classification systems provide a cost-effective and reliable alternative, significantly improving detection speed and accuracy.

Technological advancements have revolutionized traditional agricultural practices [JHS+22], paving the way for precision farming and the early detection of plant diseases. Leveraging machine learning, computer vision, and image processing technologies allows for the accurate and efficient classification of leaf diseases. This technological integration ensures rapid and precise identification of disease symptoms, facilitating immediate remedial actions.

Despite numerous advancements, achieving high accuracy in plant leaf disease classification still poses challenges due to subtle variations in image characteristics and noise. This has spurred significant research efforts, aiming to overcome these challenges and enhance the reliability of automated classification systems.

Plant leaf diseases are frequently identifiable and analyzable through direct observation. This study thus concentrates on leveraging characteristic symptoms of plant leaf diseases, encompassing alterations in shape, color, structure, or internal features of the leaves [PHL+15] [JBR+24]. The application of feature extraction methods in computer vision [Ric10] has proven effective in the classification of these diseases. The study employs straightforward machine learning algorithms like Support Vector

Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), Logistic Regression (LR), Extra Trees (ET), and CatBoost (CB) to enhance classification accuracy.

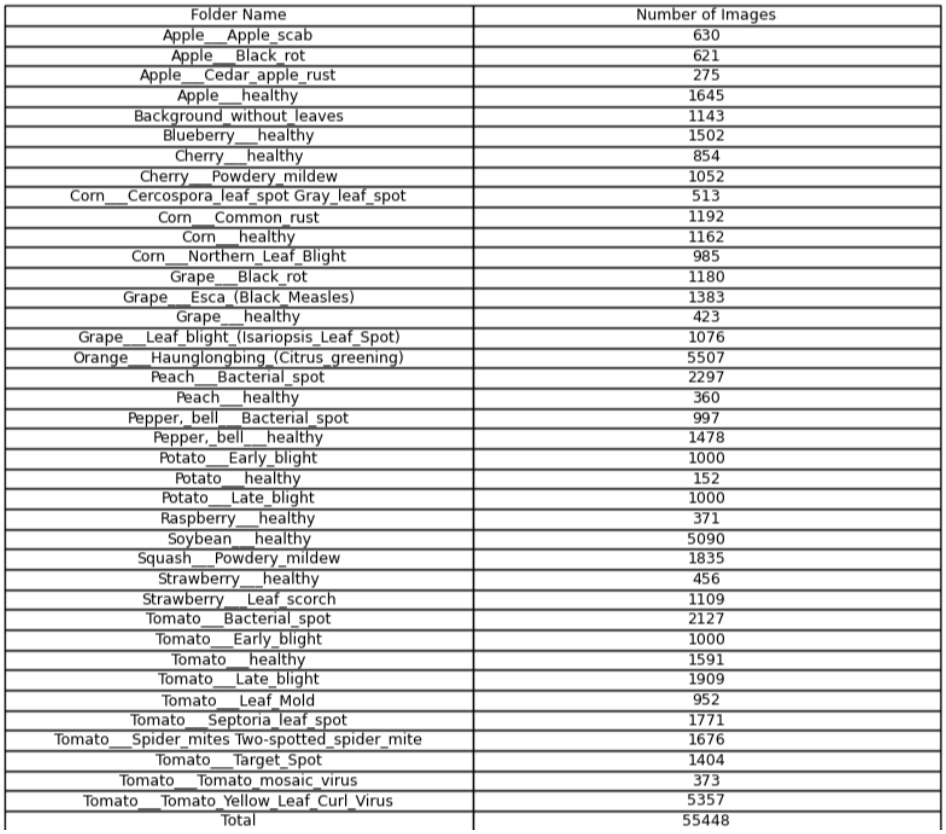
To enhance the efficiency of the classification process, this research applies feature selection techniques using random forest[Mah24] to reduce the number of features and computation time. This optimization process aims not only to maintain but also to enhance the accurate classification capability of the system.

To evaluate and ensure the reliability of the obtained results, the study employed techniques such as repeat hold-out and k-fold cross-validation. These methods assess the model's effectiveness on independent datasets, ensuring consistent and applicable results in agricultural practice.

2 Materials and Methodologies

2.1 Dataset

The diseased and healthy plant leaf images were sourced from the PlantVillage dataset available at https://data.mendeley.com/datasets/tywbtsjrjv/1. This dataset comprises a total of 55,448 images (256 x 256), categorized into 38 distinct classes of diseased and healthy plant leaves. Additionally, there is one class containing 1,143 background images. Each class is labeled to indicate whether the plant leaf is healthy or infected with a specific disease. For more detailed information:



Folder Name	Number of Images
Apple___Apple_scab	630
Apple___Black_rot	621
Apple___Cedar_apple_rust	275
Apple___healthy	1645
Background_without_leaves	1143
Blueberry___healthy	1502
Cherry___healthy	854
Cherry___Powdery_mildew	1052
Corn___Cercospora_leaf_spot Gray_leaf_spot	513
Corn___Common_rust	1192
Corn___healthy	1162
Corn___Northern_Leaf_Blight	985
Grape___Black_rot	1180
Grape___Esca_(Black_Measles)	1383
Grape___healthy	423
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1076
Orange___Haunglongbing_(Citrus_greening)	5507
Peach___Bacterial_spot	2297
Peach___healthy	360
Pepper,_bell___Bacterial_spot	997
Pepper,_bell___healthy	1478
Potato___Early_blight	1000
Potato___healthy	152
Potato___Late_blight	1000
Raspberry___healthy	371
Soybean___healthy	5090
Squash___Powdery_mildew	1835
Strawberry___healthy	456
Strawberry___Leaf_scorch	1109
Tomato___Bacterial_spot	2127
Tomato___Early_blight	1000
Tomato___healthy	1591
Tomato___Late_blight	1909
Tomato___Leaf_Mold	952
Tomato___Septoria_leaf_spot	1771
Tomato___Spider_mites Two-spotted_spider_mite	1676
Tomato___Target_Spot	1404
Tomato___Tomato_mosaic_virus	373
Tomato___Tomato_Yellow_Leaf_Curl_Virus	5357
Total	55448

Figure 1: Classes of leaf disease dataset

2.2 Experiment Setup

The experiments were conducted on a CPU core i7 equipped with 32 GB of RAM running Windows 11. The implementation utilized essential data science tools such as NumPy, Pandas, Scikit-Learn, and Jupyter Notebook for coding. Python 3.0 served as the primary programming language for development and execution.

2.3 Methodologies

2.3.1 Basic features

a) **Color descriptors**
In this section of the study, color histograms[HSS+21] are a statistical method used for representing the distribution of colors in an image. By quantizing the color space into discrete bins and counting the number of pixels that fall into each bin, histograms provide a concise summary of the color content across the image. This method facilitates efficient color-based image retrieval, segmentation, and classification tasks by capturing the frequency of occurrence of different color intensities or combinations within specified color spaces such as HSV. By examining empirical data, the selection of 64 bins has been identified as optimal. This choice strikes a balance between capturing detailed color information and maintaining computational efficiency within the image analysis framework.

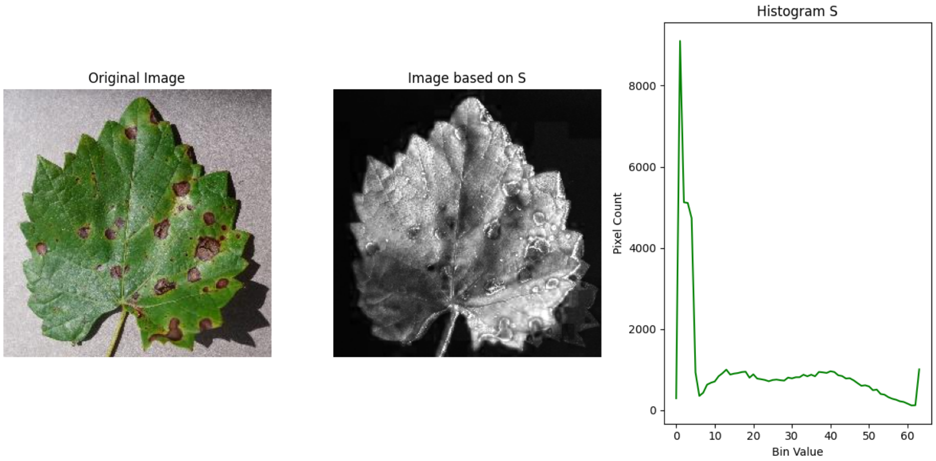


Figure 2: Example of Histogram Color utilizing HSV space

Besides, color spaces like HSV, RGB, Lab, and grayscale are also used to compute mean and standard deviation values for assessing color distribution in images. For each color space (excluding grayscale images), color distribution is evaluated by normalizing each color channel value with respect to the total average color value of the image. This method measures the percentage contribution of each color channel to the overall image color, offering detailed insights into color distribution and characteristics across different color spaces.

b) **Shape descriptors**
Detecting plant diseases based on image shape features involves utilizing the shape and patterns of plant leaves for disease identification and analysis. A prominent example of a shape feature commonly applied is the leaf contour [WFY+23]. The leaf contour represents the distribution of pixels along the leaf boundary, providing detailed information about the size, shape, and structure of the leaf. Below, edge detection algorithms will be employed to automatically extract the leaf contour [zLKG18].

Measuring leaf area is utilized to detect the presence and severity of leaf diseases, such as rust and mildew, which can cause leaf shriveling and size reduction. However, shape features like leaf area often vary with scale and can be influenced by the distance between the camera and the leaf during image acquisition [KPG18].

Additionally, other shape features such as leaf perimeter, aspect ratio, and compactness have also been employed for plant disease detection [MAA+22]. Leaf perimeter denotes the total length of the leaf boundary, reflecting the overall shape of the leaf. Aspect ratio is the ratio of leaf width to height, providing information about leaf shape and orientation. Leaf compactness measures the uniformity of leaf shape and can be used to detect the presence of disease or damage [zLKG18].

Parameter	Description
Area	Number of pixels inside the contour
Perimeter	Arc length of the contour
Length	Major axis length of the minimum bounding rectangle
Width	Minor axis length of the minimum bounding rectangle
Length to Width Ratio	Aspect ratio of Length to Width
Major Axis Length	Length of the major axis of the fitted ellipse
Minor Axis Length	Length of the minor axis of the fitted ellipse
Convex Hull Area	Area of the convex hull around the contour
Convex Hull Perimeter	Perimeter of the convex hull around the contour
Solidity	Aspect ratio of Area to Convex Hull Area
Aspect Ratio	Aspect ratio of Width to Length
Extent	Ratio of the area of an object's bounding box to its actual area

c) **Texture descriptors**

Parameter	Formula
Mean (m)	$m = \sum_{i=1}^{L-1} z_i p(z_i)$
Standard deviation (σ)	$\sigma = \sqrt{\sum_{i=1}^{L-1} (z_i - m)^2 p(z_i)}$
Uniformity	Uniformity $= \sum_{t=0}^{L-1} p^2(z_i)$
Third moment	Third moment $= \sum_{i=1}^{L-1} (z_i - m)^3 p(z_i)$
Energy	Energy $= \sum_i \sum_j P(i,j)^2$
Contrast	Contrast $= \sum_i \sum_j (i-j)^2 \cdot P(i,j)$
Homogeneity	Homogeneity $= \sum_i \sum_j \frac{P(i,j)}{1+(i-j)^2}$
Entropy	Entropy $= -\sum_i \sum_j P(i,j) \cdot \log(P(i,j))$
Correlation	Correlation $= \frac{\sum_i \sum_j [ij \cdot P(i,j)] - \mu_x \cdot \mu_y}{\sigma_x \cdot \sigma_y}$
Skewness	Skewness $= \frac{\sum_{i=1}^{L-1} (z_i - m)^3 p(z_i)}{\sigma^3}$

Where z_i represents the gray-scale intensity, $p(z_i)$ denotes the probability density function indicating the ratio of pixels with intensity z_i to the total number of pixels in the image, and $P(i,j)$ stands for the gray-level co-occurrence matrix (GLCM) [MAM+22] representing the joint probability of two pixels having specific intensities i and j being spatially adjacent in the image. The texture feature set comprises 11 components: mean (m), standard deviation (σ), uniformity, third moment, energy, contrast, homogeneity, entropy, correlation and skewness.

Local binary patterns (LBP) are regarded as one of the most powerful texture descriptors. It was first introduced by Ojala et al. [OPH96] and is used to represent the local features of an image, which are the important points of an image [HEHS+23]. The classic LBP operator is defined as a 3×3 pixel window. The center pixel of this window is used as a threshold; if the value of the neighboring pixel is less than the threshold value, the pixel value is labeled 0. Otherwise, it is labeled 1. This approach will generate an 8-bit binary number that will be converted to a decimal value, as shown below:

$$LBP_{I,J}(g_c) = \sum_{j=0}^{J-1} G(g_j - g_c) \cdot 2^j \tag{1}$$

where:

$$G(m) = \begin{cases} 1 & \text{if } m < 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In equation (4), $g_c = g(m,n)$ is the center pixel at position (m,n), and $g_j = g(m_j, n_j)$ is a neighboring pixel of the central pixel g_c. The LBP feature extraction of an image sample is indicated in Figure 3.

Additionally, Texture features are commonly extracted from plant leaf images using various image processing techniques, such as Gabor filters [JF91] and fast Fourier transforms (FFT) [Bra56]. These methods analyze the spatial frequency and local patterns on the leaf surface, offering valuable insights into detecting plant diseases and assessing their severity. One effective non-parametric descriptor is

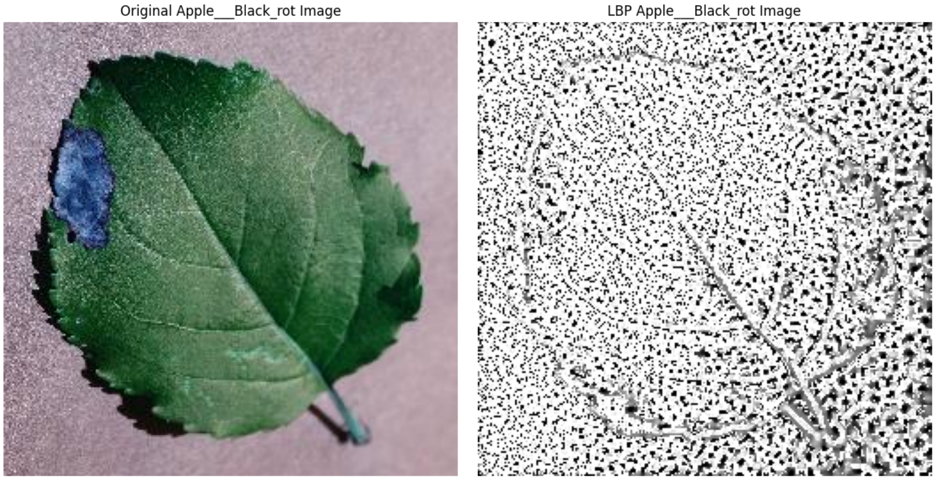


Figure 3: Visualization of Local Binary Pattern (LBP)

the texture spectrum, which utilizes the frequency spectrum of the leaf texture to distinguish between different patterns [ZIL12]. It captures spatial frequency, energy distribution, and provides a detailed characterization of textural features present in the leaf image.

2.3.2 Computer Vision Based Techniques

a) **Canny Edge Detection**

The Canny edge detection algorithm [Can86] is a widely used method for detecting edges in images. It aims to identify areas of the image with rapid intensity changes, which often correspond to edges. The algorithm involves several steps to ensure that the detected edges are accurate and meaningful. The main steps of the Canny edge detection algorithm are:

1. Gaussian Smoothing: Gaussian smoothing is the first step in the Canny edge detection algorithm, used to reduce noise and detail in the image. The smoothed image $I'(x, y)$ is obtained by convolving the original image $I(x, y)$ with a 2D Gaussian kernel $G(x, y)$:

$$I'(x, y) = I(x, y) * G(x, y)$$

where $G(x, y)$ is defined as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Here, σ is the standard deviation of the Gaussian distribution, controlling the extent of smoothing.

2. Gradient Calculation: After smoothing, the gradient of the image is calculated to detect edges. The gradients in the x and y directions, G_x and G_y, are obtained by applying derivative filters (such as Sobel operators) to the smoothed image $I'(x, y)$:

$$G_x = \frac{\partial I'}{\partial x}, \quad G_y = \frac{\partial I'}{\partial y}$$

The magnitude and direction of the gradient are then calculated as follows:

$$G = \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Where: - G is the gradient magnitude, indicating the strength of the edge at each pixel. - θ is the gradient direction, indicating the orientation of the edge.

These gradient values are used in subsequent steps to identify and refine the edges in the image.

3. Non-Maximum Suppression: After computing the gradient magnitude and direction, non-maximum suppression thins out the edges by retaining only local maxima. This involves:

- Comparing each pixel's gradient magnitude to its neighbors along the gradient direction.
- Setting the pixel value to zero if it is not greater than its neighbors' magnitudes.

The result is a binary image with thin edges, suitable for subsequent edge tracking and thresholding.

4. Double Thresholding: Double thresholding distinguishes strong edges from weak ones and reduces noise. This involves:

- **Defining high and low thresholds:** Two thresholds classify pixels based on gradient magnitudes.
- **Classifying pixels:**
  - **Strong edges:** Pixels with gradient magnitudes above the high threshold.
  - **Weak edges:** Pixels with gradient magnitudes between the thresholds.
  - **Non-edges:** Pixels with gradient magnitudes below the low threshold.

The result is an image with clearly marked strong edges and identified weak edges for further processing.

5. Hysteresis Thresholding: Hysteresis thresholding ensures weak edges are correctly classified based on their connectivity to strong edges. This involves:

- Tracing weak edges to check their connectivity to strong edges.
- Retaining connected weak edges and suppressing those that are not connected.

The result is a binary image with well-defined, continuous edges, highlighting the boundaries of objects in the image.

In addition to the aforementioned Canny method, utilizing Sobel [Sob68] and Robert [Rob63] operators to assist in plant leaf disease recognition is a crucial step in the image processing and analysis pipeline. Sobel and Robert operators detect edges and boundaries within images, thereby highlighting essential features of plant leaf diseases. This significantly enhances the accuracy and efficiency of classification and recognition models, contributing to timely diagnosis and treatment.

b) **Harris Corner Detection**

The Harris corner detection algorithm [HS88] is utilized for identifying key points in an image that correspond to corners, which are significant for various computer vision tasks. This algorithm involves the following key steps:

1. Image Derivatives Calculation:

$$I_x = I * \frac{\partial}{\partial x} G, \quad I_y = I * \frac{\partial}{\partial y} G$$

where I is the image and G is the Gaussian kernel.

2. Structure Tensor Formation:

$$M = \sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

3. Corner Response Function:

$$R = \det(M) - k(\text{trace}(M))^2$$

where k is an empirical constant typically set between 0.04 and 0.06.

4. Non-Maximum Suppression: - Suppress non-maximum points based on R values to select prominent corners.

5. Thresholding: - Apply a threshold to retain only corners with sufficiently high R values.

6. Corner Localization: - Refine corner locations to sub-pixel accuracy using interpolation.

7. Result Visualization: - Mark detected corners on the image to visualize the detected keypoints.

c) **SIFT Descriptor**

Lowe [Low04] introduced the scale-invariant feature transform (SIFT), designed to withstand image scaling, translation, rotation, and partly illumination changes. Computation of SIFT features involves several key steps:

(i) **Detecting scale-space extrema using the Laplacian of Gaussian (LoG):** Initially, local extrema of the Laplacian in scale space are extracted. This is achieved by constructing a Gaussian pyramid, where the image is progressively smoothed with Gaussian filters of increasing standard deviations. Differences of Gaussians (DoG) are then calculated at different scales to approximate the Laplacian of Gaussian. Keypoints are identified as local maxima and minima in the DoG images.

(ii) **Localizing keypoints:** Once potential keypoints are detected, they are refined to improve their localization. This involves fitting a quadratic function to the local sample points to determine the exact location and scale of each keypoint. Keypoints with low contrast or those located along edges are discarded to enhance stability and robustness.

(iii) **Assigning canonical orientation:** To ensure rotational invariance, an orientation is assigned to each keypoint. This is done by computing the gradient magnitude and orientation around the keypoint within a local region. A histogram of gradient orientations is constructed, and the highest peak in the histogram indicates the dominant orientation, which is then assigned to the keypoint.

(iv) **Describing keypoints:** Keypoints are described using histograms of gradient orientations within a local neighborhood around the keypoint. The region is divided into smaller sub-regions (typically a 4x4 grid), and an orientation histogram is created for each sub-region. The concatenated histograms form a 128-dimensional SIFT descriptor, which is highly discriminative and compact compared to raw image data.

For image description with SIFT, modern approaches often use the Bag of Words (BoW) [Liu13] technique. This technique involves clustering the SIFT descriptors from a training set to form a visual vocabulary. Each image is then represented as a histogram of visual words, where the descriptor size is influenced by the BoW model's vocabulary size. In our experiments, we typically set the vocabulary size to 400 using elbow k-means clustering.

2.3.3 Feature Selection And Machine Learning

a) **Feature Selection**

Finally, after the feature selection process, the number of features has been reduced from over 250 down to 64. This reduction has significantly improved both the runtime and accuracy of the model. According to the Pareto Principle, approximately 20% of important factors influence 80% of the results. Therefore, the method of selecting important features using Random Forest [Mah24] has played a crucial role in optimizing these outcomes before proceeding with training using other machine learning algorithms.

b) **Classification**

In this section, various machine learning algorithms were employed to train and evaluate the model:

- **k-Nearest Neighbors (KNN)**[Alt92]: A non-parametric method used for classification and regression. It assigns new data points to the most common class among its k nearest neighbors in the feature space.

- **Logistic Regression (LR)**[Cox58]: A linear model for binary classification that uses the logistic function to model the probability of a certain class. It can be extended to handle multi-class problems using techniques like one-vs-rest or multinomial logistic regression.

- **Support Vector Machine (SVM)**[Vap95]: A powerful supervised learning algorithm used for classification, regression, and outlier detection. It finds the optimal hyperplane that best separates classes in the feature space. For non-linear data structures of the dataset, the Radial Basis Function (RBF) kernel is often suitable and promising based on numerous experimental studies [MSG+23].

- **Random Forest (RF)**[Bre01]: An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or the average prediction (regression) of the individual trees.
- **CatBoost (CB)**[OGV+17]: A gradient boosting library that uses gradient boosting on decision trees. It is designed to handle categorical features automatically and can achieve state-of-the-art results with less parameter tuning.
- **Extra Trees (ET)**[GEW06]: Another ensemble learning method similar to Random Forest, but with a higher degree of randomization. It builds multiple decision trees using random splits and averages the predictions of the trees.

These algorithms were selected based on their suitability for the task and their performance in handling the dataset.

2.3.4 Results and Discussions

The reliability of classification models was assessed based on their performance metrics, including accuracy, precision, recall, and F1 score. These metrics are defined as follows:

- **Accuracy**:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

  Accuracy measures the overall correctness of the model by calculating the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances.

- **Precision**:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

  Precision indicates the proportion of true positive predictions among all positive predictions made by the model. It reflects the model's ability to avoid false positives.

- **Recall** (also known as Sensitivity or True Positive Rate):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

  Recall measures the model's ability to identify all relevant positive instances, calculating the proportion of true positive predictions among all actual positives.

- **F1 Score**:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

  The F1 Score is the harmonic mean of precision and recall, providing a single metric that balances both concerns.

Finally, the classification results of these methods using different types of features are shown in Table 1, Table 2, and Table 3 below:

Table 1: hold-out 7:3

Model	Accuracy	Precision	Recall	F1 Score
KNN	0.9332	0.9337	0.9332	0.9321
LR	0.9603	0.9604	0.9603	0.9603
SVM	0.9832	0.9833	0.9832	0.9832
RF	0.9575	0.9579	0.9575	0.9566
CB	0.9733	0.9734	0.9733	0.9733
ET	0.9350	0.9356	0.8836	0.8977

Again, SVM consistently emerged as the top-performing model across all evaluation methods, demonstrating superior accuracy, precision, recall, and F1 score. CatBoost and Random Forest followed closely, showcasing strong performance and robustness, particularly in handling complex decision boundaries and ensemble learning benefits.

Table 2: 10-Repeated-Hold-out (7:3)

Model	Accuracy	Precision	Recall	F1 Score
KNN	0.9445 ± 0.0017	0.9446 ± 0.0017	0.9445 ± 0.0017	0.9440 ± 0.0017
LR	0.9539 ± 0.0011	0.9538 ± 0.0010	0.9539 ± 0.0011	0.9537 ± 0.0011
SVM	0.9823 ± 0.0005	0.9823 ± 0.0005	0.9823 ± 0.0005	0.9822 ± 0.0005
RF	0.9568 ± 0.0013	0.9567 ± 0.0013	0.9568 ± 0.0013	0.9561 ± 0.0013
CB	0.9791 ± 0.0009	0.9791 ± 0.0009	0.9791 ± 0.0009	0.9791 ± 0.0009
ET	0.9540 ± 0.0011	0.9542 ± 0.0011	0.9540 ± 0.0011	0.9533 ± 0.0011

Table 3: 10 k-fold

Model	Accuracy	Precision	Recall	F1 Score
KNN	0.9501 ± 0.0023	0.9496 ± 0.0023	0.9501 ± 0.0023	0.9504 ± 0.0023
LR	0.9564 ± 0.0026	0.9562 ± 0.0026	0.9564 ± 0.0026	0.9564 ± 0.0027
SVM	0.9851 ± 0.0015	0.9851 ± 0.0015	0.9851 ± 0.0015	0.9852 ± 0.0015
RF	0.9616 ± 0.0019	0.9611 ± 0.0019	0.9616 ± 0.0019	0.9618 ± 0.0019
CB	0.9810 ± 0.0017	0.9809 ± 0.0017	0.9810 ± 0.0017	0.9809 ± 0.0017
ET	0.9575 ± 0.0023	0.9568 ± 0.0024	0.9575 ± 0.0023	0.9577 ± 0.0022

2.3.5 Conclusion And Future Works

Throughout our evaluation, the importance of feature engineering was evident. Effective feature engineering involves selecting and transforming raw data into meaningful features, including basic features and various techniques from computer vision and image processing. These features not only enabled accurate classification between classes but also improved the models' ability to generalize.

Reducing the feature set to 64 using random forest significantly sped up computations while increasing accuracy. The best performance was achieved by SVM, with accuracy, precision, recall, and F1 score achieving 98.23%, 98.23%, 98.23%, and 98.22%, respectively.

In future research, our focus will be on enhancing the quality of features and designing applications tailored to specific agricultural challenges, particularly in production plants in Vietnam aimed at assessing plant leaf quality. We aim to advance feature engineering techniques by exploring deep learning-based feature extraction methods and integrating domain-specific knowledge to extract meaningful features related to agricultural variables unique to Vietnamese production environments. Furthermore, we will develop user-centric applications that integrate predictive models for real-time decision support in crop management, disease detection, and resource allocation. By enhancing feature quality and application design, we aim to contribute towards more effective and sustainable agricultural practices through the application of machine learning technologies.

References

[Alt92] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, August 1992.

[Bra56] Ronald N. Bracewell. Strip integration in radio astronomy. *Australian Journal of Physics*, 9:198–217, 1956.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[Cox58] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, April 2006.

[HEHS^{+}23] Khalid M. Hosny, Walaa M. El-Hady, Farid M. Samy, Eleni Vrochidou, and George A. Papakostas. Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern. *IEEE Access*, 11:62307–62317, 2023.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[HSS^{+}21] Hamdani Hamdani, Anindita Septiarini, Andi Sunyoto, Suyanto Suyanto, and Fitri Utaminingrum. Detection of oil palm leaf disease based on color histogram and supervised classifier. *Optik*, 245:167753, 2021.

[JBR^{+}24] Seyed Mohamad Javidan, Ahmad Banakar, Kamran Rahnama, Keyvan Asefpour Vakilian, and Yiannis Ampatzidis. Feature engineering to identify plant diseases using image processing and artificial intelligence: A comprehensive review. *Smart Agricultural Technology*, 8:100480, 2024.

[JF91] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[JHS^{+}22] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Shanay Rab, and Rajiv Suman. Exploring impact and features of machine vision for progressive industry 4.0 culture. *Sensors International*, 3:100132, 2022.

[KMJ^{+}23] Bahman Khoshru, Debasis Mitra, Kuldeep Joshi, Priyanka Adhikari, Md Shafiul Islam Rion, Ayomide Emmanuel Fadiji, Mehrdad Alizadeh, Ankita Priyadarshini, Ansuman Senapati, Mohammad Reza Sarikhani, Periyasamy Panneerselvam, Pradeep Kumar Das Mohapatra, Svetlana Sushkova, Tatiana Minkina, and Chetan Keswani. Decrypting the multi-functional biological activators and inducers of defense responses against biotic stresses in plants. *Heliyon*, 9(3):e13825, 2023.

[KPG18] S. Kaur, S. Pandey, and S. Goel. Semi-automatic leaf disease detection and classification system for soybean culture. *IET Image Processing*, 12(6):1038–1048, 2018.

[Liu13] Jialu Liu. Image retrieval based on bag-of-words model. *ArXiv*, abs/1304.5168, 2013.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[MAA^{+}22] Mohammed Maray, Amani Abdulrahman Albraikan, Saud S. Alotaibi, Rana Alabdan, Mesfer Al Duhayyim, Waleed Khaild Al-Azzawi, and Ahmed alkhayyat. Artificial intelligence-enabled coconut tree disease detection and classification model for smart agriculture. *Computers and Electrical Engineering*, 104:108399, 2022.

[Mah24] Ahsanullah Yunas Mahmoud. Efficient feature selection for classification of immunotherapy and medical treatments utilising random forest and decision trees. *Intelligence-Based Medicine*, page 100151, 2024.

[MAM^{+}22] Amal Mathew, Anson Antony, Yash Mahadeshwar, Tanisha Khan, and Apeksha Kulkarni. Plant disease detection using glcm feature extractor and voting classification approach. *Materials Today: Proceedings*, 58:407–415, 2022. International Conference on Artificial Intelligence Energy Systems.

[MSG^{+}23] Anant Mehta, Prajit Sengupta, Divisha Garg, Harpreet Singh, and Yosi Shacham Diamand. Benchmarking the effectiveness of classification algorithms and svm kernels for dry beans. *arXiv preprint arXiv:2307.07863*, 2023.

[OGV^{+}17] Liudmila Ostroumova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Neural Information Processing Systems*, 2017.

[OPH96] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[PHL+15] Thi-Thu-Hong Phan, Tran Thi Thanh Hai, Thi-Lan Le, Vo Ta Hoang, Hai Vu, and Thuy Thi Nguyen. Comparative study on vision based rice seed varieties identification. *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*, pages 377–382, 2015.

[Ric10] Szeliski Richard. *"Computer Vision: Algorithms and Applications"*. Springer, 2010.

[Rob63] Lawrence G Roberts. Machine perception of three-dimensional solids. *Optical and Electro-optical Information Processing*, pages 159–197, 1963.

[Sob68] Irwin Sobel. A 3x3 isotropic gradient operator for image processing. *Presentation at Stanford Artificial Intelligence Project (SAIL)*, 1968.

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

[WFY+23] Hao Wu, Lincong Fang, Qian Yu, Jingrong Yuan, and Chengzhuan Yang. Plant leaf identification based on shape and convolutional features. *Expert Systems with Applications*, 219:119626, 2023.

[ZIL12] Dengsheng Zhang, Md. Monirul Islam, and Guojun Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, 2012.

[zLKG18] Wei zhen Liang, Kendall R. Kirk, and Jeremy K. Greene. Estimation of soybean leaf area, edge, and defoliation using color image analysis. *Computers and Electronics in Agriculture*, 150:41–51, 2018.