

Hyperparameter-Tuned K-Nearest Neighbor for Herbal Leaf Classification Using Shape, Color, and Texture Feature Fusion in Smart Agriculture

Avip Kurniawan¹, Agustinus Eko Setiawan², Vevi Maritha³, Muhammad Abrar Argianata⁴

^{1,4}Department of Informatics Engineering, Faculty of Technology, Universitas Krisnadwipayana, Jakarta, Indonesia

²Department of Informatics Engineering, Faculty of Technology and Business, Universitas Aisyah Pringsewu, Lampung, Indonesia

³Pharmacy Study Program, Faculty of Health and Science, Universitas PGRI Madiun, Madiun, Indonesia

Received:

May 20, 2026

Revised:

September 9, 2026

Accepted:

September 27, 2026

Published:

October 24, 2026

Corresponding Author:

Author Name*:

Avip Kurniawan

Email*:

avip.kurniawan@unkris.ac.id

DOI:

10.63158/journalisi.v8i5.1938

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)



Abstract. This study develops a lightweight K-Nearest Neighbor (K-NN) classifier to identify five Indonesian herbal leaf species—betel, bay, guava, soursop, and moringa—for potential smart-agriculture applications. The initial dataset contained 1,000 leaf images and was expanded to 6,000 images through offline augmentation. After data cleaning, 5,884 images were retained for analysis. Each image was resized to 256 × 256 pixels, converted into grayscale and HSV color spaces, and segmented using Otsu thresholding. A total of 21 features representing shape, color, and Gray-Level Co-occurrence Matrix (GLCM)-based texture characteristics were extracted. K-NN hyperparameters were optimized through Grid Search with Stratified 5-Fold Cross-Validation using 80:20, 70:30, and 60:40 training-testing splits. Under image-level evaluation, the best configuration used Manhattan distance, $k = 1$, and distance weighting, achieving 92.18% accuracy and a 92.29% macro F1-score on the 80:20 split. The 70:30 and 60:40 splits achieved accuracies of 91.79% and 91.21%, respectively. Misclassifications primarily occurred between soursop and guava leaves. Although the framework is compact and interpretable, augmentation before splitting may have produced optimistic results. Future work should apply source-grouped validation and assess real on-device performance.

Keywords: Herbal Leaf Classification, K-Nearest Neighbor, Feature Extraction, Hyperparameter Tuning, Smart Agriculture

1. INTRODUCTION

Herbal plants are an integral part of traditional medicine practices in many countries, including Indonesia. The World Health Organization estimates that approximately 80% of the global population still relies on plant-based medicine as part of primary healthcare services. Indonesia alone is home to more than 30,000 medicinal plant species, approximately 9,600 of which have been shown to possess pharmacological properties [1]. This potential highlights the need for an accurate, rapid, and widely accessible system for identifying medicinal plant species to the wider community [2], [3].

In practice, herbal leaf identification is still largely performed by visual inspection, making the result subjective when species share similar shape, color, or surface texture [4]. Misidentification is not merely taxonomic because an incorrectly selected medicinal leaf can lead to inappropriate herbal preparation and potential safety risks. An objective image-based identification aid is therefore relevant for users who do not have ready access to botanical expertise.

Digital image processing and machine learning provide an alternative for automated identification. Leaf appearance can be represented by complementary shape, color, and texture descriptors [5], [6]. HSV separates chromatic information from brightness, while GLCM characterizes local texture patterns [7], [8]. K-Nearest Neighbor (K-NN) is attractive for field applications because it is non-parametric, computationally lightweight, and traceable to stored training samples [9].

Previous studies have explored different feature-model combinations. Meiriyama et al. [5] obtained 92.67% accuracy using HOG-LBP with K-NN, while Naeem et al. [10] reported 99.01% using multispectral and texture features under laboratory acquisition. Other studies used GLCM with K-NN [7], [8], weighted K-NN [11], or color-GLCM-shape features with conventional classifiers [4]. CNN-based approaches [12], [13] generally achieved higher accuracy but with greater computational requirements, while HSV-K-NN performance can remain sensitive to illumination [1].

A few researchers have combined all three visual feature groups, and fewer still have systematically tuned the K-NN configuration. Two gaps therefore remain: incomplete visual representation when only one or two feature groups are used, and uncertain K-

NN performance when k , weighting, and distance metric are left at default or ad-hoc values. Because distance choice can materially change K-NN performance [14], both feature fusion and hyperparameter tuning need to be evaluated together.

Therefore, this research intends to develop a lightweight K-NN model for five Indonesian herbal leaf species using fused shape, HSV color, and GLCM texture features. The objectives are to (1) construct a 21-dimensional feature vector, (2) identify the optimal K-NN configuration through Grid Search with stratified cross-validation, and (3) evaluate accuracy, precision, recall, F1-score, execution time, and stability across 80:20, 70:30, and 60:40 splits. The study is oriented toward smart-agriculture settings where local processing and limited resources favor efficient models [15], [16]. The selected species—betel, bay, guava, soursop, and moringa—are commonly used medicinal plants in Indonesia [2], [3], [17]. The contributions are the joint use of 21 handcrafted shape-color-texture features, systematic K-NN hyperparameter tuning across three split scenarios, and class-level error analysis to identify the most difficult species pair and guide future feature improvement.

2. METHODS

This study uses an experimental quantitative approach based on the Cross Industry Standard Process for Data Mining (CRISP-DM). The analytical stages cover business and data understanding, data preparation, modeling, and evaluation; direct deployment is outside the present scope [18]. Implementation used Python, OpenCV, scikit-image, and scikit-learn in Google Colaboratory [19].

As summarized in Figure 1, the reported image-level experimental pipeline consists of image acquisition, resizing to 256×256 pixels, grayscale/HSV conversion, threshold segmentation, augmentation and cleaning, extraction and concatenation of 2 shape, 3 HSV, and 16 GLCM features, StandardScaler normalization, Grid Search with Stratified 5-Fold Cross-Validation, and held-out evaluation. This figure describes the protocol that produced the numerical results reported in this manuscript; Accordingly, the reported results reflect the original image-level protocol and do not constitute source-level validation.

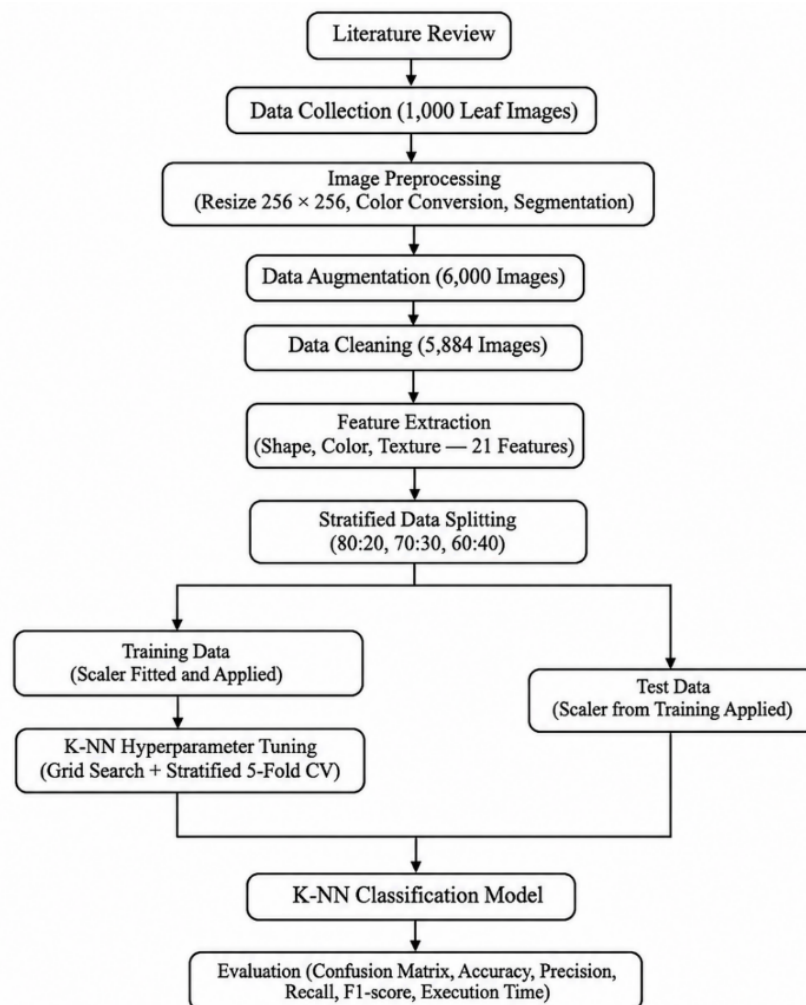


Figure 1. CRISP-DM-based workflow of the reported image-level experiment from image acquisition and preprocessing to 21-feature fusion, K-NN tuning, and evaluation

2.1. Data Collection

The data used in this study consist of digital leaf images from five classes. The primary data were obtained through direct acquisition using a high-resolution digital camera under natural lighting conditions, while the secondary data were obtained from the public Kaggle repository. Each class was represented by 100 primary images and 100 secondary images, resulting in an initial total of 1,000 images with a balanced distribution across classes. The combination of these two sources was used to enrich variations in background, capture angle, and lighting conditions, so that the model would not learn only the characteristics of a single acquisition condition. Sample images from both sources are presented in Figure 2, and the dataset composition is detailed in Table 1.

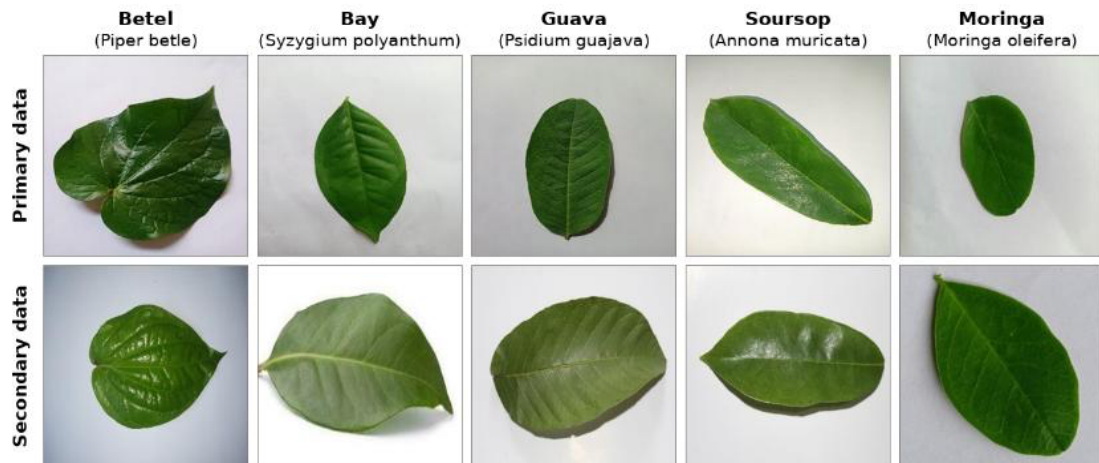


Figure 2. Sample images of the five herbal leaf species from the primary and secondary sources

Figure 2 illustrates the visual challenges addressed in this study. Betel leaves can be clearly distinguished by their heart-shaped blades, while moringa leaves can be identified by their small, rounded leaflets. In contrast, guava and soursop leaves both have elongated elliptical shapes and dark green surfaces, making their differences primarily dependent on subtle details of leaf venation and surface gloss. The figure also shows differences in background and lighting conditions between the two data sources, representing variations that the model must be able to tolerate.

Table 1. Composition of the herbal leaf image dataset

No	Leaf Species (Scientific Name)	Primary Data	Secondary Data	Initial Total	After Augmentation
1	Betel (Piper betle)	100	100	200	1,200
2	Bay (Syzygium polyanthum)	100	100	200	1,200
3	Guava (Psidium guajava)	100	100	200	1,200
4	Soursop (Annona muricata)	100	100	200	1,200
5	Moringa (Moringa oleifera)	100	100	200	1,200
	Total	500	500	1,000	6,000

2.2. Image Preprocessing and Augmentation

The preprocessing stage aims to standardize the quality and format of the images before feature extraction. All images were resized to 256×256 pixels to standardize the input dimensions while reducing the computational load. The RGB images were then converted to grayscale for shape and texture feature extraction and to the HSV color space for color feature extraction, as shown in Figure 3. The leaf objects were separated from the background using thresholding-based segmentation, which produced binary images as shown in Figure 4, because an unremoved background could potentially introduce noise into the calculation of shape features. The threshold was determined automatically using Otsu's method, which selects the value that maximizes the between-class variance of the grayscale histogram, so that no manually tuned threshold was required. Because the polarity of the result depends on whether the leaf is darker or lighter than its background, the binary image was inverted whenever more than half of its pixels were labelled as foreground. Residual speckle and small gaps were then removed by morphological opening followed by morphological closing, both using an elliptical structuring element of 5×5 pixels.



Figure 3. Color space conversion of a leaf image into grayscale and HSV representations

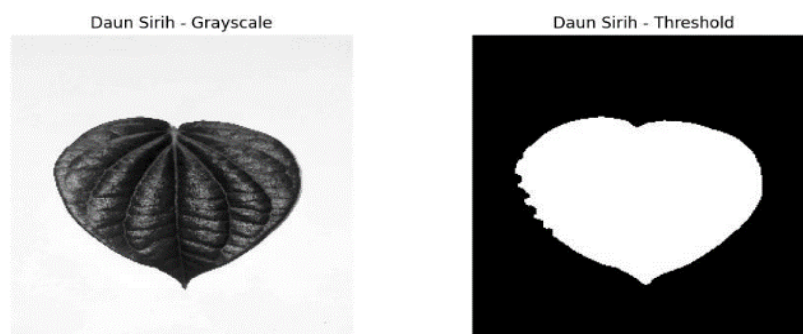


Figure 4. Object segmentation result using thresholding on a grayscale leaf image

The number of original images was considered insufficient to train a model that is robust to variations in acquisition conditions, so data augmentation was performed. Five augmentation techniques were applied: 90-degree rotation, horizontal flipping, brightness adjustment, contrast adjustment, and noise addition, with the results shown in Figure 5. Rotation and flipping improve the model's robustness to variations in orientation, brightness and contrast adjustments simulate differences in lighting conditions, while noise addition mimics camera sensor interference and image compression. The exact parameters were as follows. Rotation applied a 90-degree clockwise turn, and flipping was performed about the vertical axis. Brightness adjustment multiplied the Value channel of the HSV representation by a factor of 1.3 and clipped the result at 255. Contrast adjustment applied the linear transformation $g(x) = \alpha x + \beta$, with $\alpha = 1.5$ and $\beta = 0$. Noise addition used additive Gaussian noise with a mean of 0 and a standard deviation of 10, with the result clipped to the range $[0, 255]$. Each original image produced exactly one variant per technique, so that together with the original image every source photograph contributed six images to the dataset: 1,000 originals therefore yielded 5,000 augmented images and 6,000 images in total. All images, original and augmented alike, were resized to 256×256 pixels after the transformation was applied. Details of the augmentation results are presented in Table 2.



Figure 5. Examples of augmentation results applied to a betel leaf image

Table 2. Number of images produced by each augmentation technique, one variant per technique per source image

No	Treatment	Primary Images	Secondary Images	Total Images
1	Original images	500	500	1,000
2	Rotation of 90 degrees	500	500	1,000

No	Treatment	Primary Images	Secondary Images	Total Images
3	Horizontal flip	500	500	1,000
4	Brightness adjustment	500	500	1,000
5	Contrast adjustment	500	500	1,000
6	Noise addition	500	500	1,000
	Total	3,000	3,000	6,000

The augmentation process produced 6,000 images ready for further processing. The subsequent quality inspection identified several images that could not be properly segmented, particularly those with very low contrast between the object and the background, causing the leaf contour to be poorly formed and the eccentricity value to be undefined. Such images were removed during the data cleaning stage, resulting in a final dataset of 5,884 images used for model development.

2.3. Feature Extraction

Feature extraction transforms the visual information in an image into a numerical representation that can be processed by a classification algorithm. Three groups of features were extracted from each image, namely shape, color, and texture, with the details of the measurements calculated for each group shown in Figure 6.

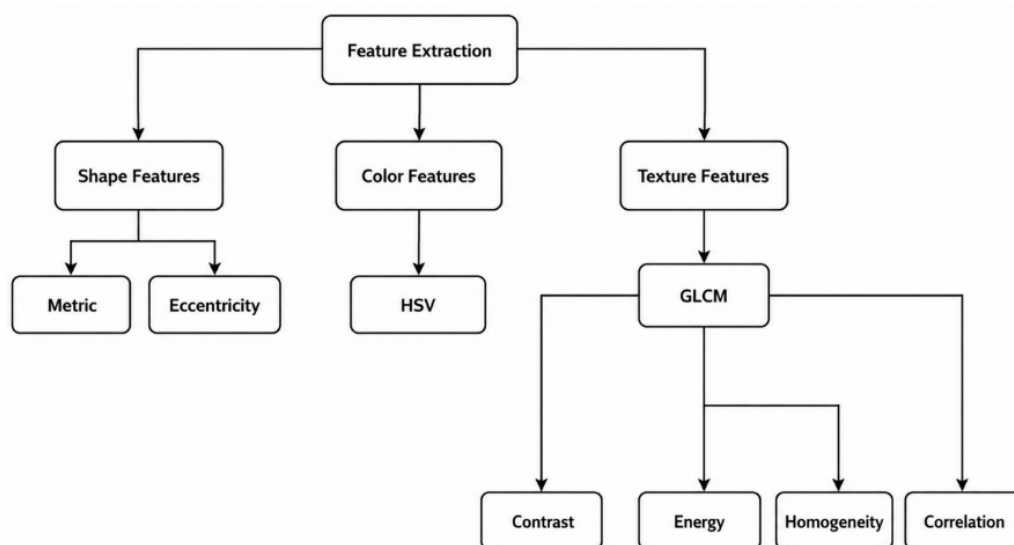


Figure 6. Flowchart of the shape, color, and texture feature extraction stages

Shape features were extracted from the segmented binary images using two measures, namely metric (circularity) and eccentricity. Metric measures the circularity of an object based on its area and contour perimeter, as expressed in Equation (1), where a value approaching 1 indicates a shape that is close to circular. Eccentricity measures the degree of elongation based on the relationship between the major and minor axes of an ellipse fitted to the object contour, as expressed in Equation (2), with values ranging from 0 for a rounded shape to values approaching 1 for a highly elliptical shape.

$$Metric = 4\pi A/P^2 \quad (1)$$

$$Eccentricity = \sqrt{(1 - b^2/a^2)} \quad (2)$$

where A denotes the area of the object, P denotes the contour circumference, a denotes the semi-major axis length, and b denotes the semi-minor axis length of the ellipse.

Color features were extracted in the HSV color space because this color space separates the pure color component (Hue), color intensity (Saturation), and brightness (Value), making it more closely aligned with human color perception and more stable to changes in lighting conditions than RGB. The mean value of each channel was calculated as shown in Equation (3), resulting in three color features: mean_H, mean_S, and mean_V.

$$mean_C = (1/N)\sum_{i=1}^N C_i \quad (3)$$

where C denotes the color channel under consideration, C(i) denotes the intensity value of the i-th pixel in that channel, and N denotes the total number of pixels in the image. Texture features were extracted using GLCM, which analyzes the frequency of occurrence of pixel pairs with specific gray levels at a given distance and direction. This study used a distance of $d = 1$ with four orientation angles, namely 0° , 45° , 90° , and 135° , at 256 gray levels using a symmetric and normalized matrix. Four statistical properties were calculated from the matrix: contrast, which measures intensity differences between neighboring pixels (Equation 4); energy, which measures distribution uniformity (Equation 5); homogeneity, which measures the proximity of the element distribution to the main

diagonal (Equation 6); and correlation, which measures the linear relationship between neighboring pixels (Equation 7).

$$\text{Contrast} = \sum_i \sum_j (i - j)^2 P(i, j) \quad (4)$$

$$\text{Energy} = \sum_i \sum_j P(i, j)^2 \quad (5)$$

$$\text{Homogeneity} = \sum_i \sum_j P(i, j) / (1 + |i - j|) \quad (6)$$

$$\text{Correlation} = \sum_i \sum_j [(i - \mu_i)(j - \mu_j)P(i, j)] / (\sigma_i \sigma_j) \quad (7)$$

where $P(i, j)$ denotes the normalized GLCM element at row i and column j , while μ and σ denote the marginal mean and standard deviation of the GLCM, respectively.

Since each property was calculated at four angles, a total of 16 texture features were obtained. All features from the three groups were then combined based on matching file names and class labels, resulting in a single 21-dimensional feature vector for each image, as detailed in Table 3.

Table 3. Composition of the combined feature vector

Feature Group	Method	Feature Names	Count
Shape	Binary image contour	metric, eccentricity	2
Color	HSV color space	mean_H, mean_S, mean_V	3
Texture	GLCM (d = 1; 0, 45, 90, 135 degrees)	contrast, energy, homogeneity, and correlation at each angle	16
Total dimension of the feature vector			21

2.4. Normalization and Data Splitting

The extracted features have substantially different value ranges; GLCM contrast values can range from tens to hundreds, whereas metric and eccentricity range from 0 to 1. This difference in scale has a substantial effect on K-NN because the algorithm makes decisions based on distances between data points, causing features with larger scales to dominate the calculation while features with smaller scales have almost no influence. Therefore, all features were normalized using standardization (StandardScaler), which transforms each feature to have a mean of 0 and a standard deviation of 1. The

transformation parameters were calculated using only the training data and then applied to the test data to prevent information leakage.

Data splitting was performed using the `train_test_split` function with the `stratify` parameter to preserve the proportion of each class in both the training and test subsets, and `random_state = 42` to ensure reproducibility. Three data-splitting scenarios were evaluated: 80:20, 70:30, and 60:40. In the primary 80:20 scenario, the dataset was divided into 4,707 training images and 1,177 test images.

Because offline augmentation was performed before train-test splitting, augmented variants from one source photograph could occur in both subsets. The reported results therefore represent an image-level evaluation and may overestimate performance on independent source photographs, particularly for $k = 1$ where a very close variant can determine the prediction. The same mechanism can also reduce apparent variance across folds and split scenarios. A leakage-controlled source-level grouped evaluation was not conducted in this study; consequently, the reported results should be interpreted within the original image-level protocol.

2.5. Modeling and Hyperparameter Tuning

The classification algorithm used in this study was K-Nearest Neighbor (K-NN). K-NN assigns the class of a test image based on the majority of labels from its k nearest neighbors in the feature space, without constructing a parametric model at the training stage. The behavior of this algorithm is determined by three main parameters: the number of neighbors (`n_neighbors`), which controls the smoothness of the decision boundary; the weighting scheme (`weights`), which determines whether all neighbors have equal influence or are weighted according to their proximity; and the distance metric (`metric`), which determines how similarity between feature vectors is measured [14]. These three parameters were optimized through Grid Search, which was configured to search the combinations of values within the predefined search space and has been shown to significantly improve model performance compared with the use of default parameters [20].

The validation scheme used was Stratified 5-Fold Cross Validation, with accuracy as the scoring metric. The search space used is presented in Table 4.

Table 4. Hyperparameter search space for the K-NN algorithm

Algorithm	Parameter	Tested Values
K-NN	n_neighbors	1, 3, 5, 7, 9
	weights	uniform, distance
	metric	euclidean, manhattan

2.6. Evaluation Metrics

Model performance was evaluated using a confusion matrix along with four derived metrics: accuracy, precision, recall, and F1-score, as defined in Equations (8)–(11) [9]. Accuracy measures the proportion of correct predictions among all observations, precision measures the correctness of positive predictions, recall measures the model's ability to identify all positive instances, and F1-score is the harmonic mean of precision and recall, which is useful when both aspects are equally important. Because this is a multiclass classification problem with a balanced class distribution, the last three metrics were reported for each class and summarized using the macro average [21], [22].

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (8)$$

$$Precision = TP / (TP + FP) \quad (9)$$

$$Recall = TP / (TP + FN) \quad (10)$$

$$F1 - score = 2 \times Precision \times Recall / (Precision + Recall) \quad (11)$$

Training and prediction time were measured as computational-efficiency indicators. Experiments used Python with OpenCV, scikit-image, and scikit-learn; timing was obtained on an AMD Ryzen 5 7535HS CPU with 16 GB RAM and no GPU. Edge feasibility was assessed only through classifier prediction time and memory footprint because no smartphone or embedded board was benchmarked directly. The reported 0.19 ms per image therefore refers to K-NN prediction after the 21-dimensional feature vector has been extracted, not to the complete image-to-prediction pipeline. The full 5,884-vector

dataset occupies about 0.94 MiB, while the 4,707-vector training database in the 80:20 experiment requires about 0.75 MiB. These values indicate low model-side storage and prediction cost, but image preprocessing and feature extraction would also contribute to latency in a real deployment and should be benchmarked on the target hardware.

3. RESULTS AND DISCUSSION

3.1. Characteristics of Shape, Color, and Texture Features

The distribution of shape features across the entire dataset is presented as a scatter plot of eccentricity and metric in Figure 7. Soursop and guava leaves occupy the high-eccentricity region, reflecting their elongated elliptical shapes, whereas betel and guava leaves account for most of the points with metric values above 0.6 as a consequence of their more rounded leaf blades. Most points from the five classes remain concentrated within the metric band below 0.25, clearly indicating that the two shape features can separate only some of the classes and are not sufficient when used independently.

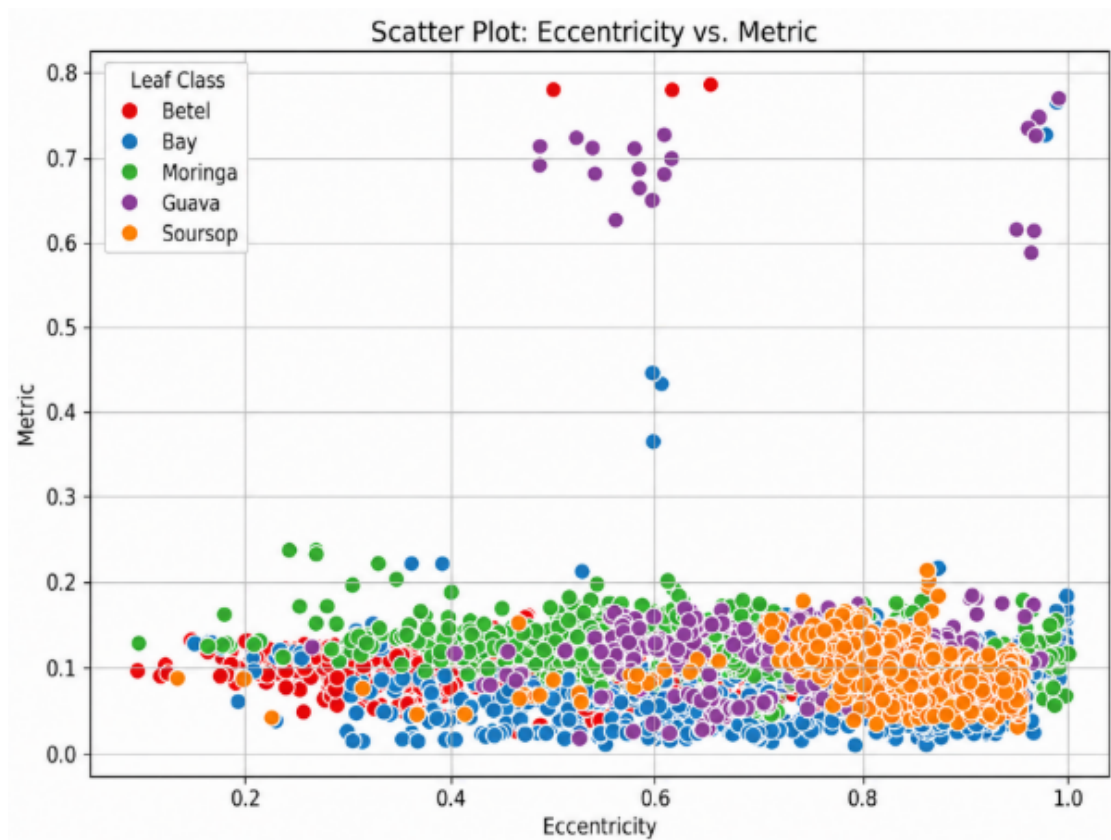


Figure 7. Scatter plot of the eccentricity and metric shape features for the five leaf classes

The distribution of color features in the three-dimensional space of mean Hue, mean Saturation, and mean Value is presented in Figure 8. The five classes form an overlapping data cloud, with the highest density in the Hue range of 40 to 80 and the Value range of 100 to 200, which is reasonable considering that all objects are green leaves. Bay leaves appear to exhibit the widest spread toward higher Value levels, whereas moringa leaves dominate the low-Saturation region. The mean value of the Hue channel remains relatively stable under geometric augmentation, whereas the Value channel shifts in response to brightness and contrast adjustments, consistent with the nature of the HSV color space in separating pure color information from light intensity.

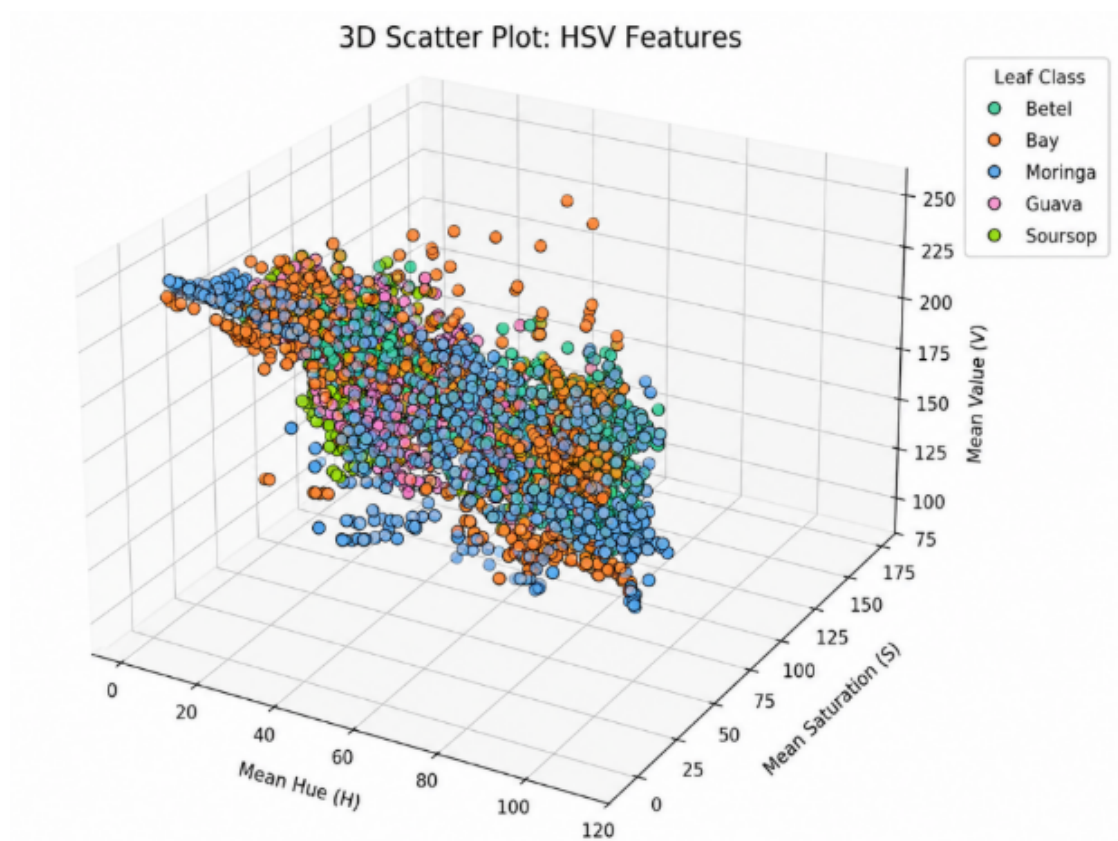


Figure 8. Three-dimensional scatter plot of the HSV color features for the five leaf classes

The relationships among the GLCM texture features are presented as a pairplot in Figure 9. The contrast feature has a right-skewed distribution, with most values below 300, whereas correlation is narrowly concentrated around values close to 1. The energy and homogeneity features exhibit an almost linear positive relationship, while contrast is negatively correlated with correlation, energy, and homogeneity, consistent with the

interpretation that high contrast indicates sharp intensity differences, whereas energy and homogeneity indicate uniformity. Moringa leaves are most clearly separated in the region characterized by low contrast and high energy and homogeneity, which explains why this class is the easiest for the K-NN model to recognize during classification.

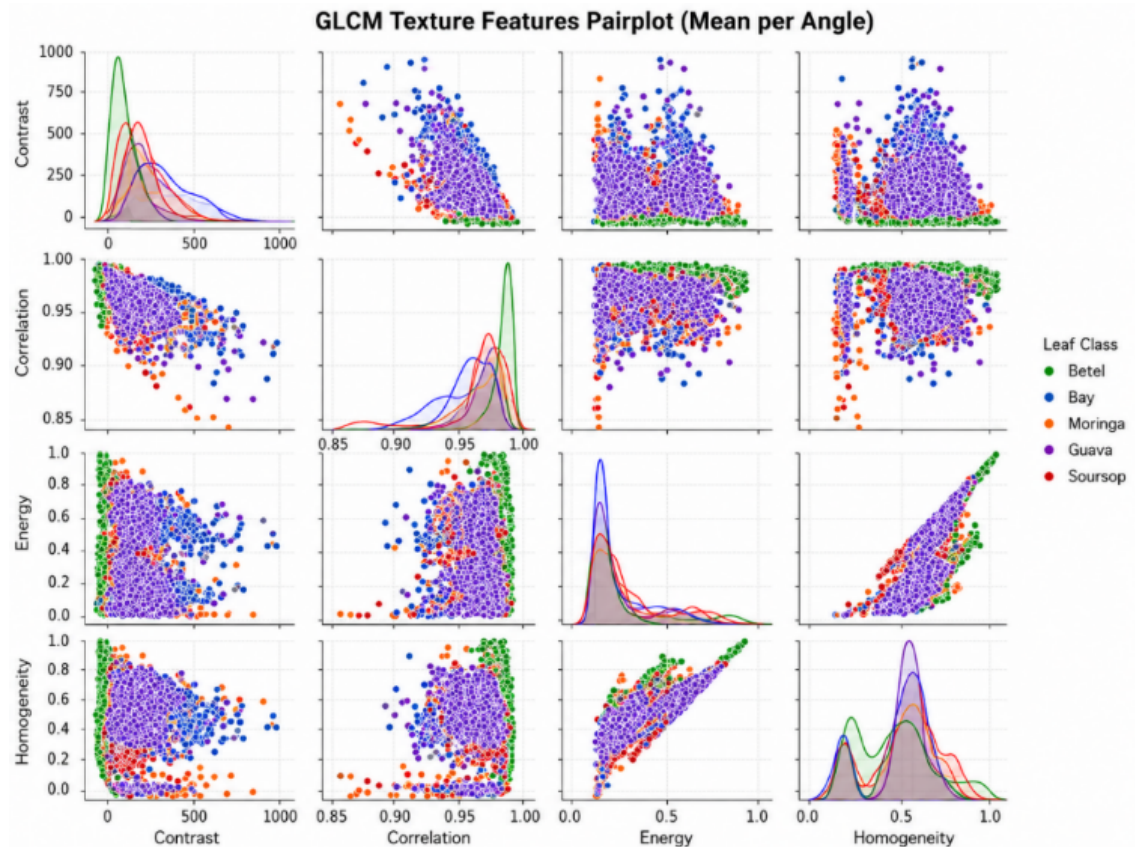


Figure 9. Pairplot of the GLCM texture features contrast, correlation, energy, and homogeneity

In terms of robustness to augmentation, the correlation values of the original images as well as the rotated and flipped images ranged from 0.97 to 0.98 across all orientations, indicating that the texture structure was preserved under geometric transformations. These values decreased slightly in images subjected to contrast adjustment, brightness adjustment, and noise addition, indicating that the GLCM features are relatively robust to mild disturbances but remain sensitive to changes in pixel patterns caused by photometric augmentation.

3.2. Hyperparameter Tuning Results

The Grid Search results in Table 5 consistently selected Manhattan distance and distance weighting, with $k = 1$ for the 80:20 split and $k = 3$ for the 70:30 and 60:40 splits. Manhattan (L1) sums absolute deviations, whereas Euclidean (L2) gives a large deviation on one feature greater influence through squaring. This distinction is important in the heterogeneous 21-dimensional feature space: metric and eccentricity are bounded shape measures, HSV means are intensity statistics, and GLCM contrast is right-skewed. Standardization equalizes scale but does not make these distributions identical. Consequently, an atypical contrast or illumination value can dominate an L2 distance more strongly than an L1 distance, while Manhattan preserves the contribution of the remaining dimensions. The result is especially relevant for leaf images because local glare, shadow, segmentation residue, or mild noise can produce a large standardized deviation in only one or two texture/color coordinates without changing the overall class identity. Under L2, those coordinates receive disproportionate influence; under L1, their contribution remains linear, so agreement across the other descriptors continues to matter.

This provides a plausible technical explanation for Manhattan being selected in all three splits rather than only one favorable partition, and it agrees with evidence that absolute-difference measures can be more robust to noise and outliers [14]. For $k = 1$, uniform and distance weighting yield the same class decision because only one neighbor votes; weighting becomes meaningful in the $k = 3$ scenarios, where closer samples receive greater influence [11]. The selection of $k = 1$ must nevertheless be regarded as provisional. Under the present protocol the training set contains augmented siblings of the test images, and such a sibling is by construction an extremely close neighbor, which systematically favours the smallest possible k . Once all variants of one source photograph are confined to the same partition, those siblings are no longer available, the distance to the first neighbor increases, and a larger k may well be selected. A source-level grouped evaluation would therefore require re-optimizing the complete hyperparameter grid rather than transferring the current configuration to a new split.

Table 5. Hyperparameter tuning results and K-NN performance under the original image-level splitting protocol

Algorithm	Split	Best Parameters	CV Accuracy (Training)	Accuracy (Testing)	Training Time (s)	Prediction Time (s)
K-NN	80:20	metric = manhattan; n_neighbors = 1; weights = distance	0.9191	0.9218	6.18	0.2198
K-NN	70:30	metric = manhattan; n_neighbors = 3; weights = distance	0.9094	0.9179	3.12	0.1753
K-NN	60:40	metric = manhattan; n_neighbors = 3; weights = distance	0.9051	0.9121	3.82	0.2257

Table 5 shows that test accuracy changed by 0.97 percentage points, from 92.18% (80:20) to 91.21% (60:40), and declined as the training set became smaller. This difference is deliberately not interpreted here as evidence of strong stability. Under image-level splitting, augmented variants of the same source photograph are distributed across the training and test subsets in every scenario, which compresses the variance between scenarios by construction; a small spread is therefore an expected property of the protocol rather than an independent finding about the feature representation. Whether the representation is genuinely stable across splitting proportions can only be established under source-level grouped validation. Test accuracy was slightly above cross-validation accuracy in all scenarios, showing a small difference between cross-validation and test performance within this protocol but not proving generalization to independent source images.

K-NN also showed low computational cost. In the 80:20 scenario, parameter search and fitting required 6.18 s and prediction of 1,177 images required 0.2198 s, or about 0.19 ms

per image. As a lazy learner, K-NN shifts cost to inference because each test vector is compared with stored training vectors; consequently, memory and prediction cost scale with the size of the feature database.

3.3. K-NN Model Evaluation

The confusion matrix of the K-NN model with the best configuration under the 80:20 scenario is presented in Figure 10. The majority of predictions lie along the main diagonal, with 1,085 correct predictions out of 1,177 test samples, resulting in an accuracy of 92.18%.

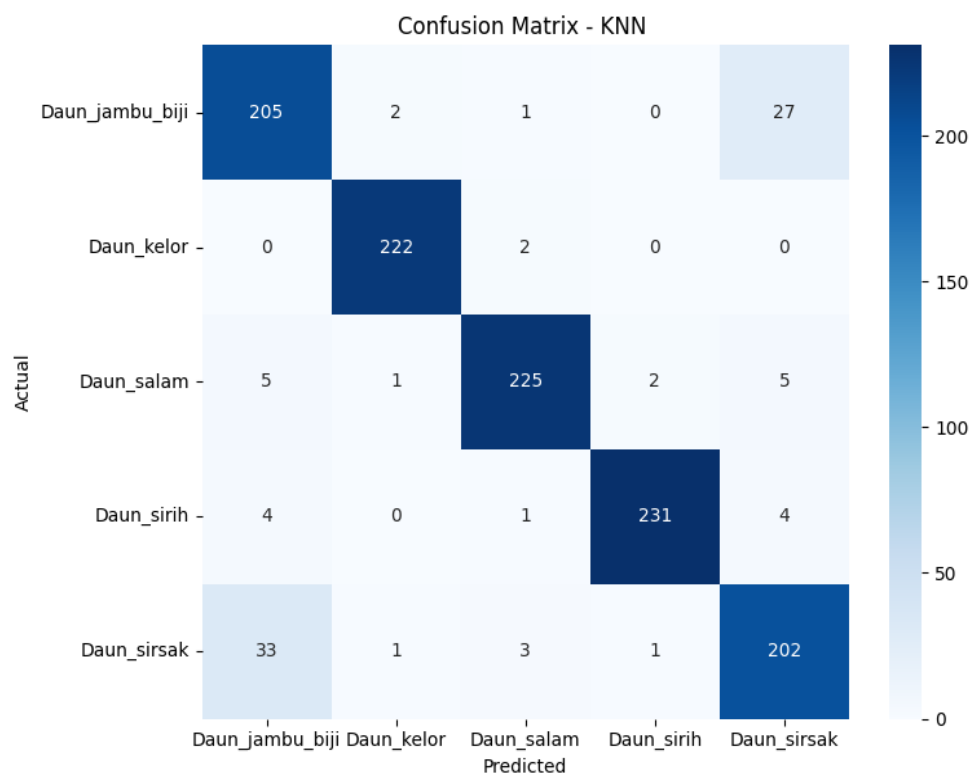


Figure 10. Confusion matrix of the K-NN model under the original image-level 80:20 splitting protocol

Details of the metrics for each class are presented in Table 6. Moringa leaves achieved the best performance with an F1-score of 98.67%, followed by betel leaves at 97.47% and Bay leaves at 95.74%. Conversely, guava and soursop leaves achieved the lowest F1-scores, at 85.06% and 84.52%, respectively.

Table 6. Per-class performance of the K-NN model under the original image-level 80:20 protocol

Leaf Class	TP	FP	FN	Precision (%)	Recall (%)	F1-score (%)
Guava	205	42	30	83.00	87.23	85.06
Moringa	222	4	2	98.23	99.11	98.67
Bay	225	7	13	96.98	94.54	95.74
Betel	231	3	9	98.72	96.25	97.47
Soursop	202	36	38	84.87	84.17	84.52
Macro average	–	–	–	92.36	92.26	92.29

3.4. Discussion

The optimal $k = 1$ configuration in the 80:20 split is consistent with dense local neighborhoods in the standardized feature space, where the nearest sample often already has the correct label. The augmented dataset also creates compact clusters of visually similar variants, so enlarging k can recruit samples from adjacent class boundaries rather than add useful information. This is particularly relevant for the guava-soursop boundary, where the two classes overlap strongly. At $k = 1$, uniform and distance weighting are functionally equivalent; the result therefore reflects locality rather than an independent advantage of weighting. When the training share is reduced to 70% and 60%, the feature space becomes more sparsely sampled and $k = 3$ is selected, suggesting that modest smoothing becomes beneficial when the nearest sample is less reliably representative. The trade-off is that $k = 1$ remains more sensitive to anomalous vectors and to the image-level augmentation protocol discussed in Section 2.4.

The confusion matrix analysis reveals that the largest source of classification errors is concentrated in a single class pair, namely soursop and guava leaves. A total of 33 soursop images were predicted as guava, while 27 guava images were predicted as soursop, resulting in 60 errors, or 65.2% of the total 92 errors, originating from these two classes alone. This highly uneven distribution of errors is important to note because it indicates that the overall accuracy of the model is primarily limited by one specific discrimination problem rather than by weaknesses distributed across all classes.

This dominant error pattern is also visually interpretable. The two species share elongated-elliptical contours, dark-green coloration, and glossy surfaces, which causes their metric, eccentricity, mean HSV, and global GLCM descriptors to overlap. In the sample images, guava also shows more pronounced secondary venation and surface relief, whereas soursop appears smoother. The current representation is not designed to exploit that distinction because the HSV descriptors are global channel means and the GLCM statistics are aggregated over the entire blade. Higher-order HSV statistics such as standard deviation, skewness, and normalized histograms could preserve within-leaf color variation that is lost in the mean, while venation descriptors based on ridge or edge filtering could encode vein density, dominant orientation, and branching structure. These additions directly target the most frequent confusion while retaining the computational advantages of handcrafted features.

For deployment-oriented smart agriculture, K-NN offers a compact stored-feature representation, no conventional iterative retraining, and traceable predictions linked to nearest training samples. New labeled examples can be incorporated by updating the stored feature database rather than retraining a deep network, which is useful when field collections grow incrementally. The measured classifier inference time and sub-1-MiB feature storage indicate potential for offline edge implementation. Traceability is another practical advantage: an operator can inspect the nearest stored examples supporting a prediction and identify problematic reference samples when an error occurs. Nevertheless, these benefits remain deployment-oriented rather than deployment-proven because the reported timing excludes image preprocessing and feature extraction and no smartphone or embedded board was tested directly [15], [16].

The findings of this study in relation to several previous studies in the same domain are summarized in Table 7. It should be emphasized that this comparison is indicative rather than directly equivalent, because each study used different datasets, numbers of classes, and evaluation protocols.

Table 7. Position of this study relative to previous research

Researcher	Features Used	Algorithms	Best Accuracy
Meiriyama et al. [5]	HOG and LBP (shape, texture)	K-NN	92.67%

Researcher	Features Used	Algorithms	Best Accuracy
Noviana and Nugraha [4]	Color, GLCM, shape	Logistic Regression, Decision Tree	78.31%
Naeem et al. [10]	Multispectral and texture (14 selected features)	Multi-Layer Perceptron	99.01%
Basri et al. [12]	Fourier Descriptor (shape)	K-NN, CNN	95.00%
Zahirah et al. [13]	GLCM (texture)	K-NN, CNN	96.00%
This study	Shape, HSV color, and GLCM texture (21 features)	K-NN with hyperparameter tuning	92.18% (image-level)

The 92.18% image-level accuracy is comparable to K-NN with handcrafted features reported by Meiriyama et al. [5] (92.67%) but below multispectral laboratory-based results [10] and CNN-based studies [12], [13], [23]. These values should be interpreted as indicative rather than directly comparable because datasets, class counts, image acquisition conditions, and validation protocols differ. The present study instead emphasizes the integration of shape, HSV, and GLCM features, systematic optimization of k , weighting, and distance metric, evaluation under three train-test proportions, and class-level error analysis [24], [25]. The CNN-based studies achieved roughly 95–96% accuracy, a few percentage points above the present result, but their training and deployment requirements are generally higher. K-NN, by contrast, stores a compact feature database and makes predictions that can be traced to specific neighboring training samples, which may be useful for resource-constrained field applications, although direct edge-device benchmarks are still required. K-NN does not involve iterative parametric model training after feature construction, but the pipeline as a whole is not free of computational preparation: feature extraction, standardization, grid search, cross-validation, and construction of the stored feature database all consume resources, and feature extraction in particular dominates the per-image cost at inference time. The advantage is therefore low model complexity and compact storage rather than zero preparation cost. The practical contribution is therefore a trade-off between predictive performance,

computational simplicity, and interpretability rather than a claim of superiority over deep learning.

Internal validity is limited by augmentation before image-level splitting, which can place variants of the same source photograph in training and test subsets and make the 92.18% estimate potentially optimistic relative to source-level validation. This issue is especially relevant to $k = 1$ because a highly similar variant can become the decisive nearest neighbor, and it may also reduce apparent variance across folds. Therefore, the reported stability across 80:20, 70:30, and 60:40 scenarios should be interpreted as stability under the present image-level protocol rather than proof of source-independent generalization. Construct validity is constrained by illumination sensitivity: HSV Value changes with brightness, Saturation can vary with lighting conditions, and grayscale GLCM can respond to shadows and specular highlights. The current color representation also uses only channel means, so two leaves with similar average color but different within-leaf distributions can appear artificially close in feature space. Severe backlighting, hard midday reflections, deep shade, and mixed artificial illumination were not represented comprehensively and may degrade color and texture descriptors more strongly than the geometric shape descriptors. External validity is limited to five species and relatively controlled backgrounds, while conclusion validity is limited by the absence of same-dataset statistical comparison with competing classifiers. Finally, deployment validity is limited because inference timing was measured on an AMD Ryzen 5 CPU rather than on a smartphone or embedded board, and the reported classifier time excludes image preprocessing and feature extraction. Future work should therefore use grouped source-image splitting, stronger illumination tests, higher-order color moments or histograms, explicit venation descriptors, statistical comparisons, and direct mobile or embedded-device evaluation.

4. CONCLUSION

This study developed a lightweight K-NN classifier for five herbal leaf species using a 21-dimensional fusion of shape, HSV color, and GLCM texture features. Grid Search with Stratified 5-Fold Cross-Validation selected Manhattan distance, $k = 1$, and distance weighting for the 80:20 scenario, yielding 92.18% accuracy and 92.29% macro F1 as an image-level estimate, while the 70:30 and 60:40 splits achieved 91.79% and 91.21%

accuracy under the same protocol. These figures are reported as image-level results throughout and are not claimed as source-independent performance. Prediction required about 0.19 ms per image on the tested CPU. Error analysis showed that 65.2% of misclassifications occurred between soursop and guava, indicating that richer color-distribution and venation descriptors are the most relevant next improvements. Because augmentation preceded image-level splitting, the reported accuracy may be optimistic relative to source-level grouped validation and should not be interpreted as independent-source generalization; the selection of $k = 1$ is the component most exposed to this effect, and the proportion of soursop-guava errors may likewise change once augmented siblings no longer cross the partition boundary. Source-level grouped validation was not conducted in this study; therefore, independent-source performance remains unverified.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Informatics Engineering Study Program, Faculty of Technology, Universitas Krisnadwipayana, and the Informatics Engineering Study Program, Faculty of Technology and Business, Universitas Aisyah Pringsewu, and Pharmacy Study Program, Faculty of Health and Science, Universitas PGRI Madiun, Madiun for the support, facilities, and contributions provided throughout the implementation and completion of this study.

REFERENCES

- [1] A. Riska, Purnawansyah, H. Darwis, and W. Astuti, "Studi perbandingan kombinasi GMI, HSV, KNN, dan CNN pada klasifikasi daun herbal," *The Indonesian Journal of Computer Science*, vol. 12, no. 3, pp. 1201–1215, 2023, doi: 10.33022/ijcs.v12i3.3210.
- [2] S. Ramadhani, J. Iskandar, and T. Husodo, "Studi etnobotani pemanfaatan tumbuhan obat di Desa Cintakarya, Kabupaten Pangandaran, Jawa Barat," in *Prosiding Seminar Nasional Masyarakat Biodiversitas Indonesia*, vol. 6, no. 1, pp. 518–524, 2020, doi: 10.13057/psnmbi/m060107.
- [3] S. F. Fatonah et al., "Pemanfaatan tanaman pekarangan untuk pengobatan herbal," *JPPM (Jurnal Pengabdian dan Pemberdayaan Masyarakat)*, vol. 4, no. 2, pp. 247–256, 2020, doi: 10.30595/jppm.v4i2.6030.

- [4] L. P. R. Noviana and I. N. B. S. Nugraha, "Perbandingan klasifikasi citra daun herbal menggunakan metode logistic regression dan decision tree classifier berdasarkan fitur (warna, GLCM, bentuk)," *JITU: Journal Informatic Technology and Communication*, vol. 7, no. 2, pp. 126–133, 2023, doi: 10.36596/jitu.v7i2.1241.
- [5] M. Meiriyama, S. Devella, and S. M. Adelfi, "Klasifikasi daun herbal berdasarkan fitur bentuk dan tekstur menggunakan KNN," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 3, pp. 2573–2584, 2022, doi: 10.35957/jatisi.v9i3.2974.
- [6] K. Pushpanathan, M. Hanafi, S. Mashohor, and W. F. Fazlil Ilahi, "Machine learning in medicinal plants recognition: A review," *Artificial Intelligence Review*, vol. 54, no. 1, pp. 305–327, 2021, doi: 10.1007/s10462-020-09847-0.
- [7] A. Z. Alfarizi and E. I. Sela, "Klasifikasi rimpang menggunakan metode k-nearest neighbor dan ekstraksi ciri gray level co-occurrence matrix," *Jurnal Fasilkom*, vol. 14, no. 1, pp. 88–94, 2024, doi: 10.37859/jf.v14i1.6832.
- [8] F. M. Fathoni, C. A. Putra, and A. L. Nurlaili, "Klasifikasi penyakit daun anggur menggunakan metode k-nearest neighbor berdasarkan gray level co-occurrence matrix," *Biner: Jurnal Ilmiah Informatika dan Komputer*, vol. 3, no. 1, pp. 8–15, 2024, doi: 10.32699/biner.v3i1.6332.
- [9] M. Vakili, M. Ghamsari, and M. Rezaei, "Performance analysis and comparison of machine and deep learning algorithms for IoT data classification," *arXiv*, arXiv:2001.09636, 2020, doi: 10.48550/arXiv.2001.09636.
- [10] S. Naeem et al., "The classification of medicinal plant leaves based on multispectral and texture feature using machine learning approach," *Agronomy*, vol. 11, no. 2, Art. no. 263, 2021, doi: 10.3390/agronomy11020263.
- [11] S. Patil and M. Sasikala, "Segmentation and identification of medicinal plant through weighted KNN," *Multimedia Tools and Applications*, vol. 82, no. 2, pp. 2805–2819, 2023, doi: 10.1007/s11042-022-13201-7.
- [12] H. Basri, Purnawansyah, H. Darwis, and F. Umar, "Klasifikasi daun herbal menggunakan k-nearest neighbor dan convolutional neural network dengan ekstraksi Fourier descriptor," *Jurnal Teknologi dan Manajemen Informatika*, vol. 9, no. 2, pp. 79–90, 2023, doi: 10.26905/jtmi.v9i2.10350.
- [13] D. Zahirah, Purnawansyah, N. Kurniati, and H. Darwis, "Digital image classification of herbal leaves using KNN and CNN with GLCM features," *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 1, pp. 61–67, 2024, doi: 10.52436/1.jutif.2024.5.1.1162.

- [14] H. A. Abu Alfeilat et al., "Effects of distance measure choice on k-nearest neighbor classifier performance: A review," *Big Data*, vol. 7, no. 4, pp. 221–248, 2019, doi: 10.1089/big.2018.0175.
- [15] M. Pintus, F. Colucci, and F. Maggio, "Emerging developments in real-time edge AIoT for agricultural image classification," *IoT*, vol. 6, no. 1, Art. no. 13, 2025, doi: 10.3390/iot6010013.
- [16] A. T. Khan, S. M. Jensen, A. R. Khan, and S. Li, "Plant disease detection model for edge computing devices," *Frontiers in Plant Science*, vol. 14, Art. no. 1308528, 2023, doi: 10.3389/fpls.2023.1308528.
- [17] F. Azzahra and A. D. Astuti, "Comparative study of the use of natural ingredient in ethnomedical study in Indonesia on the herbal formulary of the RI Ministry of Health," *Stannum: Jurnal Sains dan Terapan Kimia*, vol. 5, no. 2, pp. 58–74, 2023, doi: 10.33019/jstk.v5i2.3895.
- [18] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Computer Science*, vol. 181, pp. 526–534, 2021, doi: 10.1016/j.procs.2021.01.199.
- [19] W. Vallejo, C. Diaz-Urbe, and C. Fajardo, "Google Colab and virtual simulations: Practical e-learning tools to support the teaching of thermodynamics and to introduce coding to students," *ACS Omega*, vol. 7, no. 8, pp. 7421–7429, 2022, doi: 10.1021/acsomega.2c00362.
- [20] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *International Journal of Computers and Applications*, vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.
- [21] P. R. Sihombing and I. F. Yuliati, "Penerapan metode machine learning dalam klasifikasi risiko kejadian berat badan lahir rendah di Indonesia," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 2, pp. 417–426, 2021, doi: 10.30812/matrik.v20i2.1174.
- [22] S. Latif, F. F. Dola, M. M. Afsar, I. J. Esha, and D. Nandi, "Investigation of machine learning algorithms for network intrusion detection," *International Journal of Information Engineering and Electronic Business*, vol. 14, no. 2, pp. 1–22, 2022, doi: 10.5815/ijieeb.2022.02.01.

- [23] M. H. Ahmad, F. M. Hana, T. G. Pratama, and H. Aulida, "Klasifikasi empat jenis daun herbal menggunakan metode convolutional neural network," *Jurnal Ilmu Komputer dan Matematika*, vol. 4, no. 2, pp. 69–76, 2023.
- [24] M. F. Kurnia, S. S. Solihat, G. Windarsih, and D. Usmani, "Identifikasi otomatis lima jenis resak (*Vatica* spp.) berdasarkan beberapa karakter morfologi daun dan algoritma pembelajaran mesin," *Buletin Kebun Raya*, vol. 26, no. 1, pp. 26–37, 2023, doi: 10.55981/bkr.2023.740.
- [25] R. Muliono, J. H. Lubis, and N. Khairina, "Analysis k-nearest neighbor algorithm for improving prediction student graduation time," *Sinkron*, vol. 4, no. 2, pp. 42–46, 2020, doi: 10.33395/sinkron.v4i2.10480.