



RESEARCH ARTICLE

OPEN ACCESS

FEATURE-BASED CLASSIFICATION OF MEDICINAL PLANT: PIGWEED (URAY) AND BIRCH FLOWER (HIMBABAO) USING K-NEAREST NEIGHBORS

Abegail Ronio Rareza¹

¹Cavite State University, Philippines.

¹<https://orcid.org/0009-0005-1794-8038>

E-mail: abie.rareza@cvsu.edu.ph

ARTICLE INFO

Article History

Received: February 03, 2026

Reviewed: March 05, 2026

Accepted: April 01, 2026

Published: May 29, 2026

Keywords:

Traditional Medicine,
Classifier,
Pigweed,
Birch Flower,
Feature Extraction,
k-Nearest Neighbors
Amaranth,

ABSTRACT

This study focuses on the feature extraction of an individual image samples for both Pigweed and Birch Flower. The features are categorized with three Main types: Morphology, Color and Texture features. The extracted features will then be trained using the MATLAB Machine Learning using the Classification Learner App. Morphological features describes the shape and structural properties of the plant region in the image. This includes the area, perimeter, eccentricity, solidity, extent, major and minor axis length. The color features are also extracted from both RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value) color spaces to capture the visual appearance of the plant region wherein the algorithm calculated the mean and standard deviation which describes the average and the variability of the color. The texture features characterized the spatial variation in pixel intensity, which is important for distinguishing surface patterns. This includes contrast, correlation, energy, and homogeneity. The result shows that the validation and testing accuracy level using the k-Nearest Neighbor ranges from 88.57-99.29% and 83.33-98.33% respectively. The study's outcome demonstrates the potential of feature-based image analysis in the accurate identification of medicinal and botanical species, encouraging future developments in automated plant recognition systems.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Traditional medicine used to depend primarily on medicinal plants to provide natural treatments for a wide range of illnesses. Pigweed (*Uray*) with scientific name of *Amaranthus Spinosus* and Birch Flower (*Himbabao*) scientifically named as *Broussonetia Luzonica* have been known for their potential therapeutic properties and the flower of both plant has the similarity in color and surface texture which may resemble each other and could lead to confusion. Leafy vegetables promote reparation of energy loss due to oxidative stress, and they have the potential to alleviate hunger and malnutrition as well as other forms of metabolic imbalance ravaging the world. However, these vegetables are underutilized, despite the fact that they harbor essential minerals needed for critical cellular activities [1]. The *Uray* plant grows annually as an erect, monoecious herb, up to 100–130 centimeters tall. Its stem is either round or slightly angled in shape. It is smooth or slightly hairy, and usually green in color. The flowers of *Uray* grow in dense groups. The lower groups appear at the leaf joints (called axils), while the upper one's form both at the leaf joints and at the tips of the branches.

These upper flower groups often form spike-like structures and may have small branches near the base. The ones at the leaf joints usually have sharp spines that can grow up to 2 centimeters long [2]. The *Himbabao* tree, also called Birch Flower, is a medium-sized tree with a trunk that is roughly 30 cm broad and can reach a height of 15 meters. It has smooth bark. The leaves are hairy on the underside, have rounded bases and pointy points, and grow alternately on the branches. The flowers are tiny and grow on long, thin, spike-like stems [3]. However exact classification and identification of these plants continues to be challenging as they have common features particularly elongated leaves and green foliage, that can make it challenging for researchers to distinguish between the two species and leading to incorrect categorization. Both are widely found in the Philippines and are utilized as vegetables in local cuisine; they also function as traditional herbal medicine.

The focus of this research is the classification of the flower of both plants since it has similarities in their flowering structures though they belong to different plant families. Many modern methods for prediction leverage nearest neighbor search to find past training examples most similar to a test example, an idea that dates back in text to at least the 11th century and has stood the test of time [4]. This study aims to extract and classify the a set of features from the individual sample of flower images of *Uray* Plant and *Himbabao* Tree using an online free software version of MATLAB to support the identification and classification. The extracted features include Morphology, Color and Texture features. In machine learning, pattern recognition, and image processing, feature extraction begins with an initial set of measurement data and establishes derived values (features) aimed at providing information and non-redundancy, thus facilitating the subsequent learning and generalization steps and bringing better interpretability in some cases.

Image feature extraction is a process that uses certain methods to extract the local structure, content features, and global features of the image. Through feature extraction, the essential features with high information content, distinguishability, and high repeatability can be extracted from complex images [5]. The K-Nearest Neighbors (KNN) algorithm is a popular but efficient feature-based classification method for plant identification. KNN is a machine learning technology that can be used to differentiate between different kinds of medicinal plants since it classifies objects based on feature similarity. The feature extraction of the image samples are categorized with three Main types: Morphology, Color and Texture features. The extracted features will then be trained using the MATLAB Machine Learning using the Classification Learner App. Morphological features describes the shape and structural properties of the plant region in the image. This includes the area, perimeter, eccentricity, solidity, extent, major and minor axis length. The color features are also extracted from both RGB (Red, Green, Blue) and HSV (Hue, Saturation, Value) color spaces to capture the visual appearance of the plant region wherein the algorithm calculated the mean and standard deviation which describes the average and the variability of the color. The texture features characterized the spatial variation in pixel intensity, which is important for distinguishing surface patterns. This includes contrast, correlation, energy, and homogeneity.

II. THEORETICAL REFERENCE

II.1 EXISITING STUDIES USING K-NEAREST NEIGHBOR ALGORITHM

Among the 23 machine-learning algorithms, the best performance was achieved by medium K-nearest neighbor which has 97% accuracy and 0.14574 seconds in speed. When compared with previous studies that used RGB and HSV color features to differentiate mature and immature coffee beans, it can be concluded that texture features are far superior in distinguishing the two coffee bean groups [6].

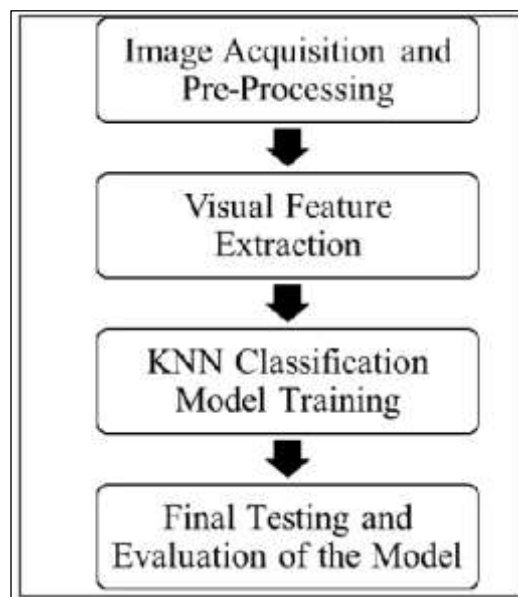


Figure 1: Procedural framework for the study.
Source: [7].

Figure 1 shows the overall procedural framework for the study. It consists of four major steps that were followed methodically to ensure proper data acquisition. Specifically, the steps include: (1) image acquisition and pre-processing, (2) visual feature extraction, (3) KNN classification model training, and (4) final testing and evaluation of the model [5]. The numerical representations of the texture extracted from the images enhanced the classification capabilities of the machine learning algorithm by providing more parameters to be fed into the system. Specifically, the MATLAB's Classification Learner App was used to classify the diseases using three KNN variants, namely: fine KNN, cosine KNN, and weighted KNN. These three variants were used since each offers different metrics and weighting strategies that optimize the classification of the dataset. Lastly, hold-out validation was implemented as the validation process to ensure the model's accuracy and generalization capabilities [7].

III. MATERIALS AND METHODS

This section discussed the methodology employed in the study to classify the difference between the two medicinal plant, Pigweed (*Uray*) and Birch Flower (*Himbabao*) based on the extracted features from an individual image sample using machine learning.

III.1 MATERIALS

The data used for this research consists a total of 200 flower samples of both plants; 100 samples of Pigweed (*Uray*) and 100 samples of Birch Flower (*Himbabao*). Following sample collection, the camera was positioned 200 centimeters away from the sample object to ensure consistent image capture, which is essential for the accurate extraction of morphological features. Shown in Figure 2. To ensure an unbiased model evaluation, the dataset was split into a simple train-test split: 70% training set and 30% test set.



Figure 2: Camera and plant sample position for image capturing.
Source: Author, (2026).



Figure 3: Sample raw *Uray* Plant.
Source: Author, (2026).



Figure 4: Sample raw *Himbabao* Plant.
Source: Author, (2026).

III.2 METHODS

An online free software was used to develop the Machine Learning based classifier. MATLAB Online was used to develop the machine learning-based classifier. The algorithm was created to process image samples organized in a designated computer folder, enabling feature-based extraction for classification purposes. The feature is categorized into three types: Morphological feature, Color feature, and a Texture feature. Figure 5 shows the procedural framework for the feature extraction before the training process. The algorithm aimed to extract relevant features to support the medicinal plant identification. The main steps include (1) Image Acquisition, (2) Pre-Processing, (3) Feature Extraction, and (4) Data Storage and Compilation.

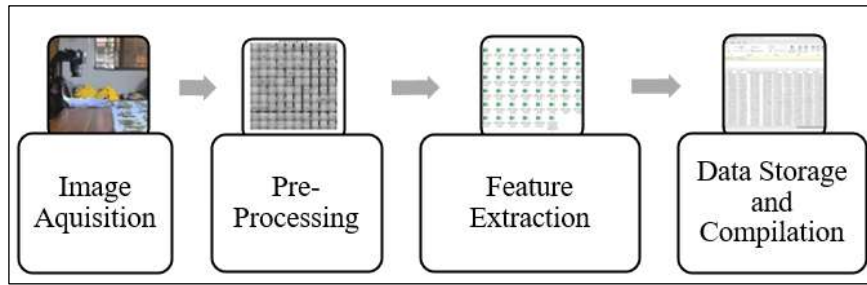


Figure 5: Steps in the Feature Extraction Algorithm used in training the dataset.
Source: Author, (2026).

Moreover, Figure 6 shows the overall procedural framework for the training, validation and testing process. Three major steps were followed to ensure the proper data classification. The steps include: (5) KNN Classification Model Training, (6) Training Result comparison, and (7) Testing and Evaluation of the Model.

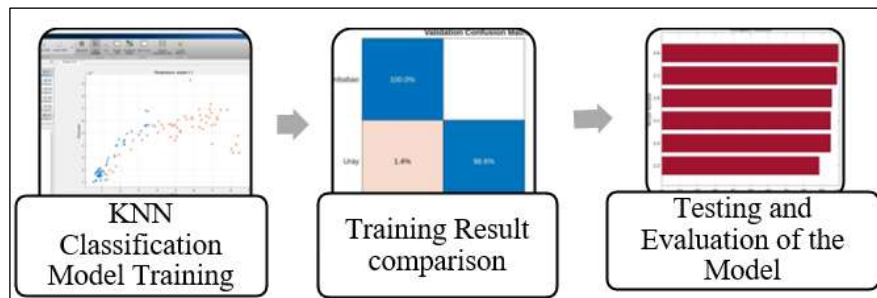


Figure 6: Steps in the Model Training and Testing.
Source: Author, (2026).

III.2.1 Image acquisition

Sample collection was the first step to create a dataset. Took individual images for a total of 200 samples, 100 flower samples of *Uray* and 100 flower samples of *Himbabao*. After sample collection, the camera was positioned 200 centimeters away from the sample to ensure that all images were captured at a consistent level, which is essential for accurate extraction of morphological features. This includes the (a) Area for the total number of pixels in the sample, (b) Perimeter for the length of the sample boundary, (c) Eccentricity for the degree of elongation based on ellipse fitting, (d) Solidity for the compactness, (e) Extent which is the ratio of the area to bounding box area, and (f) Major and Minor Axis Length of the best fit ellipse.

III.2.2 Pre-Processing

The captured image is saved to a designated folder, which will be accessed in the subsequent feature extraction step. The pre-processing procedure involves developing an algorithm specifically designed to extract the key features necessary for accurately classifying the sample objects. Shown in Figures 7, 8 and 9.

```

1 -   clc; clear; close all;
2
3   % Set your image folder path
4   folderPath = 'C:\Folderpath\Image_Samples';
5   imageFiles = dir(fullfile(folderPath, '*.jpg'));
6
7   compiledFeatures = {}; % To store all features
8
9   for i = 1:length(imageFiles)
10      filename = imageFiles(i).name;
11      filepath = fullfile(folderPath, filename);
12      img = imread(filepath);
13      gray = rgb2gray(img);
14
15      % Preprocessing
16      bin = imbinarize(imcomplement(gray));
17      bin = imfill(bin, 'holes');
18      bin = bwareaopen(bin, 300);
  
```

Figure 7: Snippets of code used for the pre-processing of the images of both.
Source: Author, (2026).

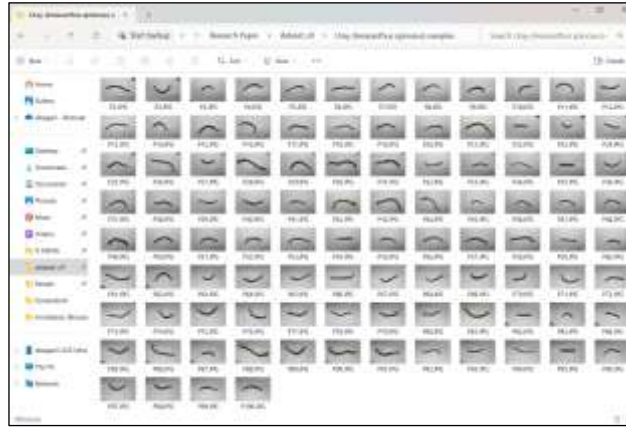


Figure 8: Individual samples of Pigweed (*Uray*).
Source: Author, (2026).

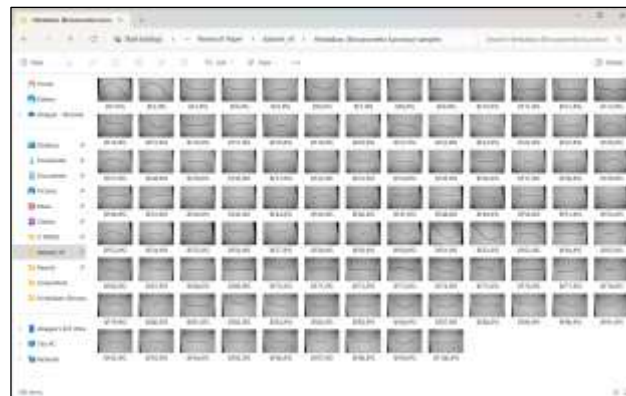


Figure 9: Individual samples of Birch Flower (*Himbabao*).
Source: Author, (2026).

III.2.3 Feature extraction

MATLAB performs the feature extraction from an image of the plant using segmentation mask. (1) Morphological, (2) Color, and (3) Texture features are calculated to be used for classification and analysis.

III.2.3.1 Morphological features

The *regionprops* function in MATLAB was used to derive the morphological characteristics of the image sample, which showed the shape forms like the Area, Eccentricity, Solidity, Extent, Major and Minor axis lengths. Shown in Figure 10.

```

26 % Morphological features
27 - props = regionprops(mask, 'Area', 'Perimeter', 'Eccentricity', ...
28   'Solidity', 'Extent', 'MajorAxisLength', 'MinorAxisLength');
```

Figure 10: Snippets of code used for extracting the morphological features of the plant.
Source: Author, (2026).

III.2.3.2 Color features

The color features were obtained by masking the RGB image and the variation for each RGB and HSV channel were calculated to describe the color characteristics of the plant. As shown on Figure 11.

```

32 % Color features (RGB & HSV mean/std)
33 - maskedRGB = bsxfun(@times, img, cast(mask, 'like', img));
34 - r = maskedRGB(:, :, 1); g = maskedRGB(:, :, 2); b = maskedRGB(:, :, 3);
35 - r = r(mask); g = g(mask); b = b(mask);
36
37 - hsv = rgb2hsv(img);
38 - h = hsv(:, :, 1); s = hsv(:, :, 2); v = hsv(:, :, 3);
39 - h = h(mask); s = s(mask); v = v(mask);
```

Figure 11: Snippets of code used for extracting the color features of the plant.
Source: Author, (2026).

III.2.3.3 Texture Features

Figure 12 shows texture feature was extracted from the grayscale version of the original image, which measures the contrast, correlation, energy, homogeneity.

```

41 % Texture features
42 - glcms = graycomatrix(gray, 'Offset', [0 1]);
43 - statsGLCM = graycoprops(glcms, {'Contrast', 'Correlation', 'Energy', 'Homogeneity'});
44 - entropyVal = entropy(gray(mask));
45 - stdVal = std(double(gray(mask)));

```

Figure 12: Snippets of code used for extracting the texture features of the plant.

Source: Author, (2026).

III.2.4 Data Storage And Compilation

After extracting the morphological, color, and texture features from the segmented plant image, the numerical values were thoroughly compiled into a structured format for further study. To ensure the clarity and consistency in interpretation, the corresponding features were combined and saved as an individual Excel file (.xlsx format).

Moreover, the individual file was then compiled and used for training the dataset. As shown on Figure 13.

```

47 % Store features
48 - features = {
49     props.Area, props.Perimeter, props.Eccentricity, props.Solidity, ...
50     props.Extent, props.MajorAxisLength, props.MinorAxisLength, ...
51     Roundness, mean(r), std(double(r)), ...
52     mean(g), std(double(g)), mean(b), std(double(b)), ...
53     mean(h), std(h), mean(s), std(s), mean(v), std(v), ...
54     statsGLCM.Contrast, statsGLCM.Correlation, ...
55     statsGLCM.Energy, statsGLCM.Homogeneity, ...
56     entropyVal, stdVal
57 }
58
59 - featureLabels = {
60     'Area', 'Perimeter', 'Eccentricity', 'Solidity', 'Extent', 'MajorAxisLength', ...
61     'MinorAxisLength', 'Roundness', ...
62     'Mean_R', 'Std_R', 'Mean_G', 'Std_G', 'Mean_B', 'Std_B', ...
63     'Mean_H', 'Std_H', 'Mean_S', 'Std_S', 'Mean_V', 'Std_V', ...
64     'GLCM_Contrast', 'GLCM_Correlation', 'GLCM_Energy', 'GLCM_Homogeneity', ...
65     'Entropy', 'Std_Gray'
66 }
67
68 % Save to individual file
69 - T = cell2table(features, 'VariableNames', featureLabels);
70 - writetable(T, fullfile(folderPath, [filename(1:end-4) 'features.xlsx']));
71
72 % Append to compiled data
73 - compiledFeatures = [compiledFeatures; [filename, features]];
74 - end

```

Figure 13: Snippets of code used for extracting the dataset.

Source: Author, (2026).

III.2.5 KNN Classification Model Training

Following the feature extraction, data storage and compilation, the k-Nearest Neighbors machine learning algorithm was used to classify two different plant species. The MATLAB Classification Learners App was used specifically using six different KNN variants; this includes Fine KNN, Medium KNN, Coarse KNN, Cosine KNN, Cubic KNN, and Weighted KNN. These variants were then used and produced different accuracy level. Before training the dataset, a 5-fold cross-validation method was applied, which means the training set was split into five parts, and the model was trained and validated five times using different combinations. This method allowed the validation of the model during training. From the total number of 200 dataset, 70% were used for training and cross-validation while the remaining 30% were set aside for testing the final model.

III.2.6 Training Result Comparison

The training results of all applied classifiers were compared in terms of accuracy, error rate, weighted precision, weighted recall, and training time to assess the performance of all the six KNN variants.

The model variant with the highest level of accuracy after the training will then be selected as the most appropriate classifier to be used for the testing and evaluation of the remaining 30% of the total dataset.

III.2.7 Testing and Evaluation of The Model

Following the training result, the summary of the classification results was compared and evaluated for further analysis.

Testing results will then be presented and compared to the result of the training and validation.

Among the evaluated models, the variant having the highest validation accuracy and the shortest training time was identified as the most effective classifier for medicinal plant recognition. When evaluating the effectiveness of the classification model, classification metrics is considered, this includes, Accuracy, Precision, Recall and F1 score. Each metrics provides different considerations into the model performance, especially when dealing with imbalanced dataset. Accuracy is defined as the the ratio of correctly predicted observations to the total number of observations in Equation (1):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision also called Positive Predictive Value is the ratio of correctly predicted positive observations to the total predicted positives in Equation (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall also known as True Positive Rate is the ratio of correctly predicted positive observations to all actual positives in Equation (3):

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where: TP – True Positive Prediction

TN – True Negative Prediction

FP – False Positive Prediction

FN – False Negative Prediction

While, F1 Score is the harmonic mean of Precision and Recall. It balances the trade-off between Precision and Recall, especially useful in imbalanced classes in Equation (4).

$$F1\ Score = 2 \times \frac{Precision + Recall}{Precision * Recall} \quad (4)$$

These values are directly generated from MATLAB's classification training output and do not require manual computation. They provide a comprehensive evaluation of the model's classification performance.

IV. RESULTS AND DISCUSSIONS

This section discussed the datasets, KNN classification, and testing and evaluation of the study to classify the difference between the two medicinal plant, Pigweed (*Uray*) and Birch Flower (*Himbabao*). The data set consisted of labeled image samples that are pre-processed and extracted to quantify the features and then compiled and organized for further analysis. The results include individual feature values for each image sample and summarized providing the overview of the variations within the dataset and serves as the basis for classification and recognition. A total of 25 predictors were obtained and used for the training, validation, and testing the dataset. The outcomes are presented separately based on the Morphological features, Color features, and Texture features. A total of 200 images were extracted and combined for the study. During training and cross-validation, 140 samples were used while the remaining 60 samples were reserved for testing to evaluate the model's performance.

IV.1 VISUAL FEATURE EXTRACTION

Table 1 shows the summary of the extracted feature values of the images of Pigweed (*Uray*) and Birch Flower (*Himbabao*) respectively based on their morphological characteristics, such as: area, perimeter, eccentricity, solidity, extent, major and minor axis which describes the shape and structure of both plants.

Table 1: Summary of the features extracted from the image samples of both *Uray* and *Himbabao*.

FEATURES:	MEDICINAL PLANT	
	URAY	HIMBABAO
	MORPHOLOGICAL FEATURES	
Area	1214290 - 9261229	538937 - 3876222
Perimeter	15941.148 - 92944.441	10119.919 - 67623.449
Eccentricity	0.89758923 - 0.99119054	0.85380014 - 0.98491387
Solidity	0.35531400 - 0.99119054	0.10661320 - 0.90169724
Extent	0.20764849 - 0.82555569	0.09764961 - 0.51583186
MajorAxisLength	2647.675 - 6608.268	1297.595 - 9243.233
MinorAxisLength	665.513 - 2608.902	4812.036 - 615.817
	COLOR FEATURES	
	URAY	HIMBABAO
	MORPHOLOGICAL FEATURES	
Mean_R	72.08598908 - 154.47394420	13.643010 - 114.92179318
Std_R	6.68634335 - 35.33909788	8.83791317 - 50.21029554
Mean_G	61.05145929 - 153.38188293	7.62580519 - 114.83187779
Std_G	6.98001929 - 37.97471723	6.74846272 - 54.05286411
Mean_B	40.47432843 - 153.69556171	6.58110151 - 110.16784694
Std_B	7.18305238 - 45.70276490	7.12412247 - 54.651938
Mean_H	0.12004872 - 0.67451770	0.15927012 - 0.41702444
Std_H	0.08027518 - 0.42889131	0.21668031 - 0.37894838
Mean_S	0.01373574 - 0.44550961	0.02770973 - 0.50918531
Std_S	0.01149347 - 0.23494281	0.02222985 - 0.36800096
Mean_V	0.28410940 - 0.60758294	0.05366450 - 0.45505230
Std_V	0.02667715 - 0.13911420	0.03464041 - 0.19862919
	TEXTURE FEATURES	
	URAY	HIMBABAO
	MORPHOLOGICAL FEATURES	
GLCM_Contrast	0.03599871 - 0.06994371	0.03949806 - 0.06835838
GLCM_Correlation	0.96993885 - 0.98710983	0.96749678 - 0.99228737
GLCM_Energy	0.23627418 - 0.34274281	0.23544176 - 0.40335679
GLCM_Homogeneity	0.96507612 - 0.98200132	0.96610505 - 0.98025127
Entropy	4.7674035 - 6.82775261	3.17047740 - 6.31095655
Std_Gray	6.83075647 - 37.56896767	7.04568311 - 52.90176494

Source: Author, (2026).

IV.2 KNN CLASSIFICATION MODEL TRAINING

After the extraction of the numerical parameters present in both plants, the KNN machine learning was trained using MATLAB online, Classification Learners App. The classification was determined by the 25 predictors obtained from the preceding process. Table 2 presented the classification results of the six KNN model variants used for training using 5-folds cross-validation, from the total of 200 samples, 140 samples (70%) were used for training the algorithm. Furthermore, a hold-out validation structure was used to set aside 60 samples (30%) from the original sample for the testing of the KNN algorithm.

Table 2: Comparison of the trained KNN model.

Model	KNN Variant	Accuracy (%)	Training Time (sec)
2.1	Fine	98.57	6.33
2.2	Medium	95	12.94
2.3	Coarse	88.57	12.47
2.4	Cosine	95	11.91
2.5	Cubic	95.71	10.29
2.6	Weighted	99.29	9.41

Source: Author, (2026).

From the table, it shows that the Fine KNN model has the fastest training time of 6.33 seconds with 98.57% level of accuracy. Coarse KNN has the lowest level of accuracy of 88.57% with 12.47 seconds training time, while Medium KNN has the slowest training time of 12.94 seconds. Weighted KNN model has the highest accuracy level of 99.29% with a total training time of 9.41 seconds. This result provides insight of which model classifies the plant species more accurately.

Table 3: Validation Results of the KNN classification using 5-folds cross-validation method.

KNN Variant	Accuracy (%)	Error Rate (%)	Weighted Precision (%)	Weighted Recall (%)
Fine	98.57	1.43	98.61	98.57
Medium	95	5	95.45	95
Coarse	88.57	11.43	88.86	88.57
Cosine	95	5	95.45	95
Cubic	95.71	4.29	96.05	95.71
Weighted	99.29	0.71	99.30	99.29

Source: Author, (2026)

Table 3 presented the validation results of all KNN classification model evaluated using weighted precision and weighted recall. Based on the results, it shows that weighted KNN among all other models performed best in classifying the plant species. It showed accuracy, error rate, precision and recall of 99.29%, 0.71%, 99.30%, and 99.29% respectively. Second is the Fine KNN having the accuracy, error rate, precision and recall of 98.57%, 1.43%, 98.61%, and 98.57% respectively. To further elaborate, the validation confusion matrix for the Weighted KNN was shown in Figure 14. From the total 140 samples, 70 were classified correctly as *Himbabao*, while for *Uray*, 69 were classified correctly and 1 misclassified as *Himbabao*.

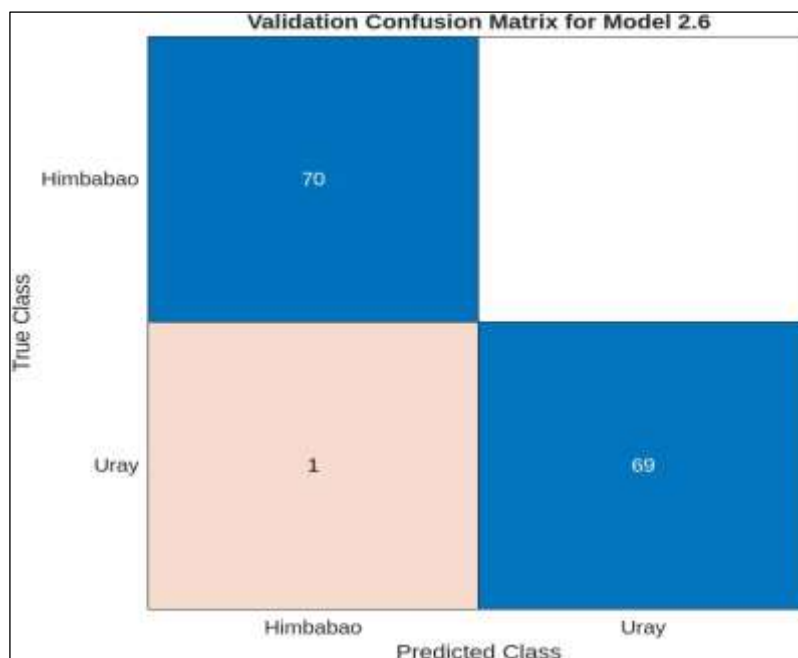


Figure 14: Validation confusion matrix for weighted KNN model generated from MATLAB online.

Source: Author, (2026).

The Parallel Coordinates Plot visually shows the how does the different features contribute to the classification outcome and how distinct the feature distributions are between the classes. Since the study has morphological, color, and texture features, with a total of 25 different predictors, each line on the plot represents each samples. The lines are color coded based on their respective class label, allowing visual interpretation of feature patterns and distinction. This visualization supports the interpretation of the feature combination will affect the result of the KNN classifier. The plot results is shown on Figure 15 for visual interpretation of the dataset.

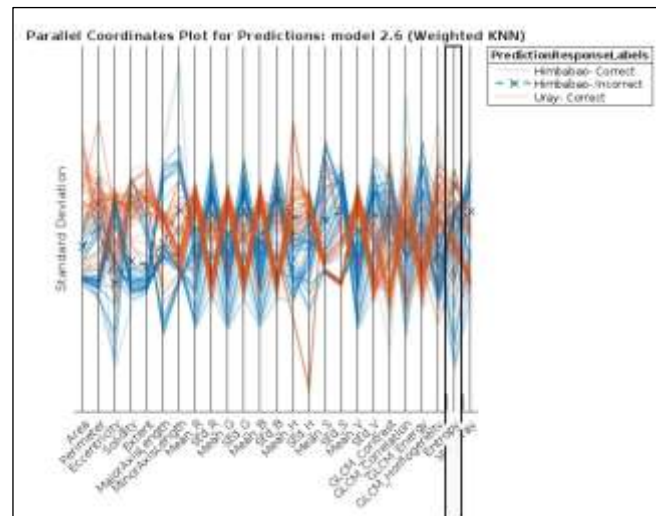


Figure 15: Parallel Coordinates Plot generated from MATLAB online.
Source: Author, (2026).

IV.3 TESTING AND EVALUATION OF THE PROPOSED CLASSIFICATION MODEL

After the process of initial evaluation of the KNN machine learning algorithm, the hold out validation (5-fold validation was used in this study) helped the optimization of the algorithm to adjust to a new data samples. Upon choosing the most suitable model, the remaining 60 samples (30%) are subjected for testing and evaluation to assess the final performance evaluation of the KNN model classification. Table 4 shows the classification results of the six KNN model variants after the testing process.

Table 4: Testing Results of the KNN classification using hold-out validation structure.

KNN Variant	Accuracy (%)	Error Rate (%)	Weighted Precision (%)	Weighted Recall (%)
Fine	100	0	100	100
Medium	93.33	6.67	94.12	93.33
Coarse	83.33	16.67	83.48	83.33
Cosine	93.33	6.67	94.12	93.33
Cubic	93.33	6.67	94.12	93.33
Weighted	98.33	1.67	98.39	98.33

Source: Author, (2026).

The Table shows the testing results of all KNN classification model evaluated using weighted precision and weighted recall. Based on the results, the Fine KNN model classification has obtained the highest level of accuracy of 100% after being tested for classification. The weighted KNN having the highest level of accuracy on the previous training and validation has now the accuracy level, error rate, precision and recall of 98.33%, 1.67%, 98.39%, and 98.33% respectively. Having the results of both Fine and Weighted KNN, the close performance implies that both models are reliable classifiers for the given dataset. Text Confusion matrix for both model is shown in Figure 16.

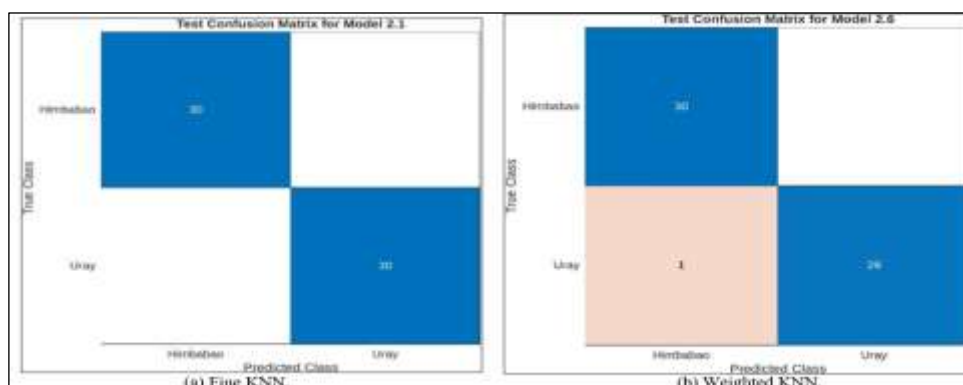


Figure 16: Test Confusion Matrix for Models (a) Fine KNN and (b) Weighted KNN
Source: Author, (2026).

These results indicate that the Fine KNN is highly effective in distinguishing between the classes having the perfect accuracy after being tested. However, the Weighted KNN model did not reach 100% test accuracy but its performance remains notable with only a slight error of 1.67% in accuracy. These values suggest that misclassification were minimal and the model has maintained a strong consistency in predicting both positive and negative classes. On the other hand, the Fine KNN's test accuracy may indicate a slight overfitting, especially that the dataset is small and lack of variation may occur. In contrary, the Weighted KNN's along with its balanced performance across all metrics demonstrate a good adaptability across all different datasets shown in Figure 17.

Per-Class Metrics (Validation)			
	Precision	Recall	F1 Score
Himbabao	98.6%	100.0%	99.3%
Uray	100.0%	98.6%	99.3%

Per-Class Metrics (Test)			
	Precision	Recall	F1 Score
Himbabao	96.8%	100.0%	98.4%
Uray	100.0%	96.7%	98.3%

Figure 17: Per-class metrics of Weighted KNN model generated from MATLAB online.
Source: Author, (2026).

IV.4 VERIFICATION OF DATA LABELING

To ensure the reliability and accuracy of the model, verification process was conducted. In plant identification studies, especially for medicinal and agricultural purposes this step shall not be neglected. While automated tools like image classification models help identifying plants, it can still make mistakes especially when distinguishing between similar species and physical features similarity. The involvement of a farmer regarding classification, helped confirm the correct labeling of plant species and ensuring that each sample in the dataset is accurately identified. In addition, the expertise provided by the farmer helped avoid misclassification due to look-a-like plants, which is a common issue in plant identification. *Uray* and *Himbabao* share similar leaf shape and texture, growth habit, color and flower structure. In this study, flower samples were obtained and used for this feature-based classification. All plant samples collected were sourced and verified by a local famer with extensive knowledge of these medicinal plants to ensure the accuracy of the dataset labels. The correct labeling of the data samples, as verified by the farmer is shown on Figure 18 wherein samples 1, 3, 5, 7 and 9 are called *Himbabao* and samples 2, 4, 6, and 8 are called *Uray*.



Figure 18: Correct labeling of the samples sourced and verified by a local farmer.
Source: Author, (2026).

According to the farmer, both plants are edible and are commonly used in traditional Filipino cuisine. The young leaves and sprouts of *Uray* are typically cooked and consumed as leafy vegetables. While the flower clusters of *Himbabao* are commonly used in regional dishes, often prepared in traditional meals such as *dinengdeng*, *pinakbet*, and *bulanglang*. Additionally, the plants are traditionally known for their medicinal properties. *Uray* is used for the treatment of wounds, digestive problems, and their use as a natural diuretic while *Himbabao* traditionally believed to help digestion and inflammation.

V. CONCLUSIONS

In summary, this study aimed to accurately identify and classify medicinal plants using a verified sample data. This study explored how machine learning specifically the k-Nearest Neighbor (KNN) algorithm, can be used to distinguish between the two medicinal plants – Pigweed (*Uray*) and Birch flower (*Himbabao*) that are often led to confusion in identification due to its similarity. By extracting the key features from the plant images such as morphological, color and texture features, and training several KNN models using MATLAB online, the study evaluated which model established the best performance for classification. Among the models tested, the Fine KNN achieved perfect accuracy (100%) during testing, while the Weighted KNN showed the most consistent and reliable performance across all stages of training, validation, and testing.

These results highlight how machine learning can be accurately used in plant identification especially when the species possess similarities in their physical features. The critical part of the study was the verification of the data. This step is essential to ensure that each sample was correctly labeled before being used in the model. The farmers expertise helped avoid misidentification, which is a common issue in plant recognition tasks. The collected samples were sourced and verified by a local farmer which added the accuracy of the dataset used. Overall, combining machine learning with human expertise proved to be an effective way to improve plant classification. This approach can be helpful for future research and practical use in agriculture, herbal medicine, and conservation especially in communities that rely on accurate plant familiarity for food and traditional healing practices.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Abegail Ronio Rareza.

Methodology: Abegail Ronio Rareza.

Investigation: Abegail Ronio Rareza.

Discussion of results: Abegail Ronio Rareza.

Writing – Original Draft: Abegail Ronio Rareza.

Writing – Review and Editing: Abegail Ronio Rareza.

Resources: Abegail Ronio Rareza.

Supervision: Abegail Ronio Rareza.

Approval of the final text: Abegail Ronio Rareza.

VII. ACKNOWLEDGMENTS

The authors want to express their sincerest gratitude and appreciation to the professors, the institution, and other relevant individuals who made the conduct of the study possible.

VIII. REFERENCES

- [1] M. O. Jimoh, K. Okaiyeto, O. O. Oguntibeju, and C. P. Laubscher, "A Systematic Review on Amaranthus-Related Research," *Horticulturae*, vol. 8, no. 3, p. 239, Mar. 2022, doi: 10.3390/horticulturae8030239.
- [2] M. Baral, A. Datta, S. Chakraborty, and P. Chakraborty, "Pharmacognostic studies on stem and leaves of *Amaranthus spinosus* Linn," *East Asia*.
- [3] "Potential Properties of *Himbabao* (*Broussonetia Luzonica* B.) As Anticancer, Antifungal and Antimicrobial," *Int. J. Acad. Multidiscip. Res. IJAMR*, vol. Vol. 6, no. Issue 4, pp. 37–51, Apr. 2022.
- [4] G. H. Chen and D. Shah, "Explaining the Success of Nearest Neighbor Methods in Prediction," 2025, arXiv. doi: 10.48550/ARXIV.2502.15900.
- [5] Z. Xu, S. Ahmad, Z. Liao, X. Xu, and Z. Xiang, "Image feature extraction algorithm based on visual information," *J. Intell. Syst.*, vol. 32, no. 1, p. 20230111, Dec. 2023, doi: 10.1515/jisys-2023-0111.
- [6] E. Arboleda, "Classification of Immature and Mature Coffee Beans Using Texture Features and Medium K Nearest Neighbor," *J. Artif. Intell. Technol.*, May 2023, doi: 10.37965/jait.2023.0203.
- [7] E. Arboleda, S. G. Rodrin, and E. C. San Diego, "Classification of Prominent Cacao Pod Diseases Using Multi-Feature Visual Analysis and k-Nearest Neighbors Algorithm," *ITEGAM- J. Eng. Technol. Ind. Appl. ITEGAM-JETIA*, vol. 11, no. 51, 2025, doi: 10.5935/jetia.v11i51.1367