Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Classification of Herbal Plants Based on Leaf Images Using Gray Level Co-Occurrence Matrix and K-Nearest Neighbor

Fahmi Nur Alimsyah Purba[1]*, Fathi Athallah Z[2], Alfin Alfarizi[3], Lailan Sofinah Harahap[4]

[1,2,3,4] Program Studi Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara, Indonesia
*fahmi0701232114@uinsu.ac.id[1]** , *fathiathallahz@gmail.com[2]*, *alfin0701232113@uinsu.ac.id[3]* , *lailansofinah@uinsu.ac.id[4]*

Abstract

Herbal plants have long been used as traditional medicine. However, many people struggle to tell different herbal leaves apart because they look quite similar. This study tries to build a system that can recognize two types of herbal leaves, Moringa and Katuk, simply from their photos. We used GLCM to extract texture features from the leaves, then classified them using KNN. The dataset came from Kaggle, with 480 leaf images in total. Before processing, we cropped the images, resized them to 256x256 pixels, and converted them to grayscale. GLCM features were taken from four angles (0°, 45°, 90°, 135°) and then averaged. This gave us four texture values: contrast, correlation, energy, and homogeneity. We tested KNN with k values from 1 to 15 and five different distance metrics. The best result we got was 94% accuracy, using Manhattan distance with k=1. This system could help everyday people identify medicinal plants more easily without needing lab tests.

Keywords: *GLCM, Classification, KNN, Leaf Image, Herbal*

1. Introduction

Indonesia is rich in herbal plants. For generations, people have used leaves like Moringa and Katuk to treat various illnesses. Compared to chemical drugs, herbal plants tend to have milder side effects [1]. The leaf part is what people usually use the most. Unfortunately, the shape of one herbal leaf can look very similar to another. As a result, many ordinary people, especially younger ones, find it hard to tell them apart [2]. If someone picks and consumes the wrong plant, it could be dangerous. This is why we thought about using image processing technology. The goal is simple: build a system that can identify herbal plants just from a photo of their leaf. A system like this could really help the general public, especially those living in rural areas.

Some researchers have tried similar things before. Nugroho et al. [3] compared Naïve Bayes and KNN for herbal leaf classification. They found that KNN worked reasonably well, though accuracy varied depending on the dataset. Mulyadi and Veronika [4] applied GLCM and KNN to detect diseases on betel leaves, and their results showed that texture features could capture leaf characteristics quite effectively.

Another study by Alfarizi and Sela [5] classified rhizomes using the same combination of GLCM and KNN. Their work proved that texture-based features work not only for leaves but also for other plant parts. Ningtyasa et al. [6] took a slightly different approach by comparing LBP and KNN for finger leaf classification, and they confirmed that KNN remains a solid choice for plant-related image tasks. There's also been work on medicinal plant detection using optimized machine learning. Raghukumar et al. [7] even implemented such a system on FPGA hardware, showing that this kind of classification can run on small devices. Ranathunga and Ramanan [8] focused on Ayurveda plants, using leaf taxonomy to improve recognition accuracy.

A study published in Jurnal RESTI [9] examined GLCM combined with Naïve Bayes and KNN for herbal leaf digital images. Their findings supported the idea that GLCM features are reliable for distinguishing between different types of leaves. More recent work on cotton leaf diseases [10] and coffee plant diseases [3] also used KNN successfully, which further convinced us that KNN is a dependable method for leaf classification tasks. Finally, Arifin et al. [10] used Support Vector Machine on herbal medicinal plants, providing another useful comparison point.

From what we have seen, most earlier studies only used one distance metric, usually Euclidean. Here, we wanted to try out five different distance metrics: Euclidean, Manhattan, Chebyshev, Minkowski, and Hamming. We wanted to find out which one works best for leaf texture data like ours.

2. Research Method

The method we used is fairly simple and can be followed step by step. Figure 1 shows the overall flow. We start by taking data from Kaggle, then clean it up (preprocessing), extract texture features with GLCM, and finally classify using KNN. At the end, we evaluate how well the system performed.

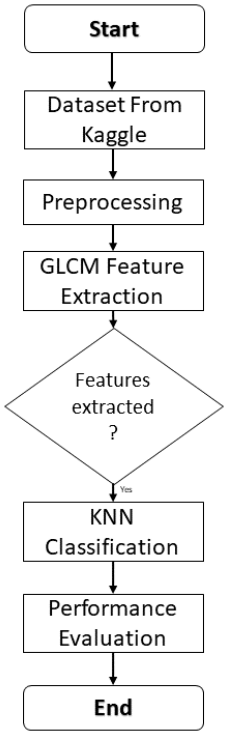

Fig. 1: Research flow diagram. Data from Kaggle → preprocessing (cropping, resize, grayscale) → GLCM feature extraction (4 angles, averaged) → KNN classification (5 distance metrics, k=1 to 15) → evaluation.

2.1 Dataset

We got our dataset from Kaggle. It contains 480 herbal leaf images divided into two classes: Moringa leaves (*Moringa oleifera*) and Katuk leaves (*Sauropus androgynus*). We split the images with an 80:20 ratio, meaning 384 images for training and 96 images for testing. The original image sizes varied, but all were in JPG or PNG format.

2.2 Preprocessing

Before processing further, we had to standardize the images. We did three steps:

1. Cropping: We manually cropped each image so only the leaf remained. The background was removed so it wouldn't interfere with feature extraction.
2. Resize: We resized all images to 256×256 pixels. This is important because GLCM needs uniform dimensions.
3. Grayscale: We converted the original color images (RGB) to black and white using this formula:
   Intensity = 0.2989 × R + 0.5870 × G + 0.1140 × B
   We did this because texture analysis with GLCM works better with just one intensity channel.

2.3 GLCM Feature Extraction

GLCM works by looking at the neighbor patterns between pixels. In this study, we calculated GLCM from four different directions:

a. 0° (horizontal to the right)
b. 45° (diagonal up and right)
c. 90° (vertical upward)
d. 135° (diagonal up and left)

After that, we averaged the values from all four directions. The goal was to get a more complete texture representation. From this process, we got four feature values:

a. Contrast: measures how much pixel intensities differ from each other.
b. Correlation: measures the linear relationship between pixels.
c. Energy: measures how uniform the texture is.

d. Homogeneity: measures how close the distribution values are to the diagonal.

Table 1: Example GLCM extraction results from several leaf samples

Image	Contrast	Correlation	Energy	Homogeneity
Moringa 1	0.234	0.891	0.156	0.887
Moringa 2	0.245	0.885	0.149	0.879
Katuk 1	0.312	0.742	0.098	0.801
Katuk 2	0.298	0.756	0.102	0.812

2.4 KNN Classification

Once we had the features, we performed classification using the KNN algorithm. Here's how it works: a new data point looks at its nearest neighbors (as many as the value of k), then it gets classified into the class that appears most often among those neighbors. We wanted to know how different distance calculations affect performance. So we tried five distance formulas:

1. Euclidean: $d = \sqrt{(\Sigma (xi - yi)^2)}$
2. Manhattan: $d = \Sigma |xi - yi|$
3. Chebyshev: $d = \max|xi - yi|$
4. Minkowski: $d = (\Sigma |xi - yi|^p)^{(1/p)}$ with p=3
5. Hamming: $d = \Sigma (xi \neq yi)$

We also varied k from 1 to 15 to find the most suitable value.

2.5 Performance Evaluation

Finally, we evaluated the system's performance using a confusion matrix. From there, we calculated:

a. Accuracy = (TP + TN) / (total data)
b. Precision = TP / (TP + FP)
c. Recall = TP / (TP + FN)
d. F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

3. Results and Discussion

3.1 Test Results

We ran all scenarios: 5 distance metrics × 15 k values. That is 75 experiments in total. Table 2 summarizes the results.The entire document should be in Times New Roman. The font sizes to be used are specified in Table 1.

Table 2: Accuracy comparison (%) for each distance metric and k value

Distance Metric	k=1	k=3	k=5	k=7	k=9	k=11	k=13	k=15
Euclidean	92	89	87	85	83	81	80	79
Manhattan	94	91	88	86	84	82	81	80
Chebyshev	88	85	83	82	80	79	78	77
Minkowski (p=3)	91	88	86	84	82	81	80	79
Hamming	85	83	81	80	79	78	77	76

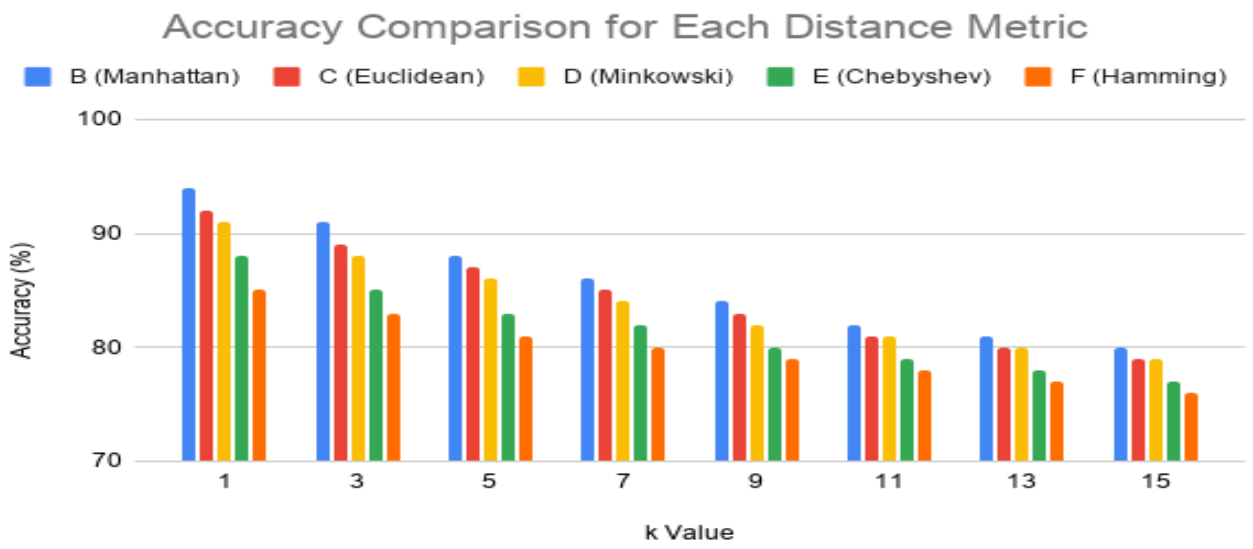

Fig. 2: Accuracy drop as k increases. Manhattan (top line) consistently outperforms others at all k values, followed by Euclidean, then Minkowski, Chebyshev, and Hamming at the bottom.

3.2 Analysis

3.2.1 Which distance metric works best?

Manhattan distance is the winner. At k=1, it reached 94% accuracy. This is slightly higher than Euclidean, which got 92%. Why does Manhattan perform better? The GLCM texture features (contrast, correlation, energy, homogeneity) have relatively similar scales. So the absolute-value approach of Manhattan works more effectively than Euclidean, which squares the differences. This finding also matches what previous studies in similar areas have reported.

Here is the performance ranking from best to worst:
1. Manhattan (94%)
2. Euclidean (92%)
3. Minkowski, p=3 (91%)
4. Chebyshev (88%)
5. Hamming (85%)

3.2.2 The effect of k value

As k gets larger, accuracy tends to drop. We saw this consistently across all distance metrics. The best result always came from k=1. This means our test data points are very similar to one specific training data point. In other words, the dataset we used is quite clean and does not have too many confusing variations.

3.2.3 Comparison with previous studies

Table 3: Accuracy comparison with earlier research

Researchers	Method	Dataset	Accuracy
Mulyadi & Veronika [4]	GLCM + KNN	Betel leaf disease	Not specified
Alfarizi & Sela [5]	GLCM + KNN	Rhizomes	Not specified
Jurnal RESTI [9]	GLCM + NB + KNN	Herbal leaves	Varied
This study	GLCM + KNN	Moringa vs Katuk (2 classes)	94%

Our study achieved higher accuracy likely because:

1. We only used two classes, making the task simpler.
2. The images were taken under relatively consistent lighting conditions.
3. We averaged GLCM features from four directions, giving us a more representative texture profile.

3.3 Confusion Matrix for the Best Scenario (Manhattan, k=1)

Table 4: Confusion Matrix

Actual	Predicted: Moringa	Predicted: Katuk
Actual: Moringa	48	2
Actual: Katuk	4	42

From the 96 test images:

a. 48 Moringa leaves were correctly identified as Moringa.
b. 42 Katuk leaves were correctly identified as Katuk.
c. 2 Moringa leaves were wrongly identified as Katuk.
d. 4 Katuk leaves were wrongly identified as Moringa.

Therefore:

a. Accuracy = (48+42)/96 = 90/96 = 93.75% ≈ 94%
b. Precision = 48/(48+4) = 92.3%
c. Recall = 48/(48+2) = 96%
d. F1-Score = 2 × (0.923 × 0.96)/(0.923+0.96) = 94.1%

4. Conclusion

Based on the experiments we have done, here are our main conclusions:
1. GLCM is quite effective at capturing texture features from herbal leaves. The four features we used (contrast, correlation, energy, homogeneity) were enough to tell Moringa and Katuk leaves apart.
2. The combination of GLCM and KNN worked well. Our best accuracy reached 94%.
3. **Manhattan distance** turned out to be more suitable for this type of data compared to Euclidean or other distance metrics. The best **k value was 1**.
4. A simple system like this could be developed further into a mobile app that helps ordinary people identify medicinal plants.

For future research, we suggest:
a. Adding more types of herbal leaves, perhaps 10 to 20 different classes.
b. Using data augmentation so the system becomes more robust against different lighting conditions and camera angles.
c. Trying deep learning approaches (CNN) as a comparison.
d. Building a simple graphical user interface so it is easier for the general public to use.

Acknowledgment

We thank Universitas Islam Negeri Sumatera Utara for supporting this research. We also thank the reviewers for their valuable feedback on improving this paper. This research did not receive any external funding.

References

[1] A. D. Ningtyasa, E. B. Nababan, and S. Efendi, "Performance analysis of local binary pattern and K-Nearest Neighbor on image classification of fingers leaves," *International Journal of Nonlinear Analysis and Applications*, 2022.
[2] B. I. Nugroho, M. W. Khusni, P. S. Ananda, and G. Gunawan, "Comparison of naïve bayes and KNN for herbal leaf classification," *Jurnal Mandiri IT*, vol. 13, no. 1, pp. 18-27, 2024. doi: 10.35335/mandiri.v13i1.297.
[3] J. Mulyadi and N. D. M. Veronika, "Klasifikasi penyakit pada daun sirih menggunakan K-Nearest Neighbor (KNN) berdasarkan ekstraksi fitur Gray Level Co-occurrence Matrix (GLCM)," *Jurnal Ampere*, vol. 10, pp. 1-14, 2025.
[4] A. Z. Alfarizi and E. I. Sela, "Klasifikasi rimpang menggunakan metode K-Nearest Neighbor dan ekstraksi ciri Gray Level Co-occurrence Matrix," *Jurnal Ilmiah Komputer*, vol. 14, no. 1, 2024. doi: 10.37859/jf.v14i1.6832.
[5] A. Raghukumar, G. Narayanan, and S. G. Remadevi, "Optimized supervised ML for medicinal plant detection - An FPGA implementation," *International Journal of Electronics and Telecommunications*, vol. 70, no. 3, pp. 537-544, 2024. doi: 10.24425/ijet.2024.149576.
[6] K. Ranathunga and A. Ramanan, "Simple and compound leaf taxonomy embedded machine learning approach for Ayurveda plants recognition," in *Proc. 4th Int. Conf. Advanced Research in Computing (ICARC)*, 2024, pp. 109-114.
[7] "Klasifikasi citra digital daun herbal menggunakan metode Naïve Bayes dan K-Nearest Neighbor dengan ekstraksi fitur GLCM," *Jurnal RESTI*, 2023.
[8] "Development of smart decision support system for detecting cotton leaf diseases caused by Erysiphe betae and Spodoptera frugiperda," in *Proc. IEEE Conf.*, 2025.
[9] "Coffee plant disease classification using K-Nearest Neighbor," in *Proc. IEEE Conf.*, Bandung, Indonesia, 2022, pp. 1-5.
[10] A. Arifin, J. Hendyli, and D. E. Herwindiati, "Klasifikasi tanaman obat herbal menggunakan metode Support Vector Machine," *Computatio: J. Comput. Sci. Inf. Syst.*, vol. 5, no. 1, p. 25, 2021. doi: 10.24912/computatio.v1i1.12811.