MACHINE LEARNING

Image-Based Medicinal Plant Identification Using Machine Learning with Integrated Medicinal Value and Safety Guidance

T. Chethan Kumar, B. S. Chandana, M. L. Chinmayi, S. Darshini, P. Ranjitha

Department of Information Science and Engineering, Kalpataru Institute of Technology, Tiptur, Karnataka, India

Published November 2026

Abstract

Correct identification of medicinal plants is important because visually similar leaves may be confused, and incorrect identification can lead to unsafe decisions. This paper presents a web-based educational prototype that classifies five medicinal plants from leaf photographs and displays associated traditional-use information and safety precautions. The implemented dataset contains 446 labelled images belonging to Aloe vera, Amla, Amruthaballi, Arali and Ashoka. Each image is represented by a 2,852-dimensional descriptor combining colour histograms and RGB statistics, Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and a thumbnail representation. A feature-scaled Support Vector Machine (SVM) with a radial basis function (RBF) kernel performs classification and produces probability-based confidence estimates. The classifier is integrated into a Flask web application that validates image uploads, displays the predicted class and three closest matches, and retrieves plant information from a structured JSON file. Medicinal information is withheld when the maximum predicted probability is below 60%. The reported held-out test accuracy is 93.3%, shuffled cross-validation accuracy is 94.8%, and blocked cross-validation accuracy is 79.6%. Average prediction time is approximately 13 ms, and the saved model occupies approximately 9 MB. Nine reported functional tests passed. The difference between shuffled and blocked evaluation indicates that generalization to separated image groups remains a concern. The prototype is intended for education and awareness; it is not a diagnostic tool and does not provide treatment or dosage advice.

Keywords: Medicinal plant identification; leaf image classification; machine learning; Support Vector Machine; HOG; LBP; Flask; medicinal information; safety guidance

1 Introduction

Medicinal plants have an established place in traditional healthcare systems, including Ayurveda, Siddha and Unani. Different plant parts, such as leaves, roots, bark, flowers and seeds, are described in traditional preparations. Plant-derived compounds have also contributed to modern pharmaceutical research. India has considerable plant diversity, and many species are familiar to local communities. Reliable identification is essential because plants with similar visual characteristics may be mistaken for one another.

Traditional identification uses botanical characteristics such as leaf shape, margin, apex, base and venation, together with flowers, fruits and other plant structures. This process can require specialist knowledge and may be difficult when only a leaf photograph is available. Computer vision provides a supporting approach by converting an image into numerical descriptors and comparing those descriptors with labelled examples.

The present work develops a local web application that accepts a photograph of a leaf and predicts one of five trained classes. It then displays the prediction confidence, alternative matches and a structured information record containing reported traditional uses and precautions. The implementation uses Python, Flask and a scikit-learn SVM model. It is designed as an educational aid and must not replace expert botanical identification or qualified medical advice.

1.1 Objectives

1. Organise 446 labelled leaf photographs belonging to five medicinal-plant classes.
2. Represent each image using a 2,852-dimensional descriptor containing colour, HOG, LBP and thumbnail features.
3. Train a feature-scaled RBF-kernel SVM with balanced class weights and probability estimation.
4. Evaluate the classifier using a held-out test split, shuffled cross-validation, blocked cross-validation and class-wise measures.
5. Develop a Flask interface for image upload, preview, prediction and result display.
6. Display medicinal information and precautions only when the prediction meets the configured 60% confidence threshold.

1.2 Scope and applicability

The implemented system processes one leaf photograph at a time and classifies it into one of five known classes: Aloe vera, Amla, Amruthaballi, Arali and Ashoka. It displays the predicted class, confidence estimate, top-three matches and the corresponding JSON information. The application validates supported image uploads and provides error or uncertainty messages.

The current implementation does not analyse flowers, fruits, bark, stems or roots. It is not designed to identify every plant species, and an unknown plant may be assigned to one of the five known classes. Its dataset may not represent all seasons, cameras, backgrounds or lighting conditions. The medicinal content is general educational information and must not be interpreted as a diagnosis, prescription, dosage recommendation or confirmation of safety.

2 Literature survey

2.1 Traditional and image-based identification

Traditional botanical identification relies on a combination of morphological traits and expert knowledge. Automated leaf identification narrows the task to image-based analysis, allowing a system to compare measurable visual patterns. However, leaf appearance changes with age, environment, illumination, orientation and damage. A leaf-only classifier therefore provides supporting evidence rather than definitive taxonomic identification.

2.2 Handcrafted features and classical machine learning

Conventional image-processing pipelines commonly use resizing, colour-space statistics, edge descriptors and texture measurements. HOG represents local gradient-orientation patterns, while LBP describes local texture transitions. Colour histograms and channel statistics capture broad colour distributions. These descriptors can be combined into a feature vector and classified using algorithms such as k-nearest neighbours, decision trees, random forests or SVMs. Classical methods

are computationally efficient and can be suitable for relatively small datasets, although performance depends on feature design and image consistency.

2.3 Deep learning and recent research

Recent work has increasingly investigated convolutional neural networks, transfer learning, feature fusion and vision transformers for plant recognition. Gautam et al. (2025) proposed a lightweight deep-learning approach for medicinal-leaf classification using feature fusion [1]. Matin et al. (2025) released a labelled medicinal-leaf image dataset covering six species, highlighting the importance of data diversity and dataset documentation [2]. Sharma et al. (2025) described an Xception-based transfer-learning pipeline for medicinal-plant leaf classification [3]. Sachar and Kumar (2025) investigated an optimised machine-learning pipeline involving segmentation, feature selection and classification [4].

Research published in 2026 further illustrates the range of current approaches. Aruna et al. compared SVM, CNN, YOLOv8, ViT, ResNet and ANN approaches for medicinal-leaf classification [5]. Algemayel et al. reviewed machine-learning models for plant-species identification and discussed generalisation challenges in field images [6]. Khuat and Van investigated a unified EfficientSwinB-based framework for medicinal-plant recognition and related plant classification [7]. Priya et al. presented FloraMediX, an AI-enabled platform combining visual recognition with structured medicinal-plant information [8]. These studies demonstrate the value of both image classification and information retrieval, while also showing that larger and more diverse datasets are important.

2.4 Research gap and motivation

A number of recent systems focus mainly on improving image-classification accuracy. However, a practical educational interface also needs to communicate uncertainty and present relevant contextual information responsibly. The present prototype addresses this application-level need by connecting a lightweight feature-based classifier to a structured plant-information file and suppressing the medicinal-information panel when the predicted probability is below a configured threshold. The work is deliberately limited to five classes and does not claim to provide a comprehensive medicinal-plant database.

3 System requirements and implementation environment

3.1 Hardware requirements

Table 1. Hardware requirements.

Component	Requirement / purpose
Processor	A Core i3-class processor or equivalent can run the application; a Core i5-class processor or higher is preferable for development and training.
RAM	4 GB minimum; 8 GB or more recommended for development and concurrent tools.
Storage	At least 10 GB of free space is recommended for images, dependencies and model files.
GPU	Not required for the implemented SVM pipeline; optional GPU resources may support future CNN experiments.
Camera / display	A phone camera or webcam and a standard display are sufficient to capture and review leaf images.

3.2 Software requirements

The evaluated implementation uses Python, Flask, scikit-learn, scikit-image, Pillow and NumPy. The model-training and prediction scripts share the same image-feature extraction procedure. The browser provides the user interface, while Flask handles upload validation, model inference and retrieval of the plant-information record. CNN frameworks and Streamlit are not part of the evaluated implementation and are therefore not presented as implemented components.

3.3 Dataset description

The dataset contains 446 labelled leaf photographs distributed across five classes: Aloe vera, Amla, Amruthaballi, Arali and Ashoka. The source manuscript reports 118 Aloe vera images and 67 Amla images; it does not provide the exact counts for the remaining three classes in the available summary. The dataset is organised by class so that the training script can read images and their labels from the corresponding folders. The limited dataset size and unequal class distribution are considered when interpreting the results.

4 Methodology and system design

The methodology consists of two connected processes: (i) an offline training and evaluation process that creates the saved classifier, and (ii) an online prediction process that uses the saved model to classify a newly uploaded leaf photograph. The same feature-extraction function is used in both processes so that the training and prediction vectors have a consistent representation.

4.1 Overall system architecture

Figure 1 presents the implemented architecture. The user selects an image through the browser. Flask receives the file and checks its extension, size and readability. The validated image is passed to the feature-extraction module, which generates the 2,852-dimensional descriptor. The saved scaler and SVM model produce a class prediction and probability estimates. The application applies the confidence rule before retrieving the associated plant-information record from the JSON file.

Figure 1. Architecture of the implemented Flask–SVM system

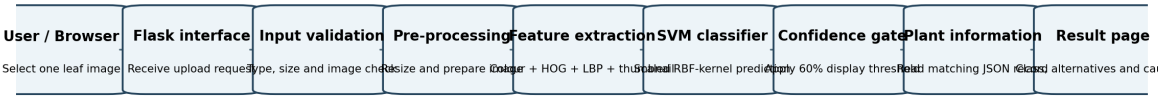

Figure 1. Architecture of the implemented Flask–SVM system.

4.2 Dataset preparation

The labelled photographs are arranged in class-specific folders. During training, the data-loading routine reads the images and associates each image with the class represented by its folder. The training script checks that the expected classes and usable images are available before model fitting. The report does not claim that image augmentation was applied to the evaluated SVM model.

4.3 Image pre-processing

Before feature extraction, images are read and reduced to a consistent image representation. The source implementation reports that images are reduced to 128 × 128 pixels. This provides a common input size for the descriptor-generation process and reduces computational requirements. During prediction, the uploaded image is checked for validity and processed through the same image-feature function used during training.

4.4 Feature extraction

Each image is represented by a 2,852-dimensional feature vector formed by combining four groups of descriptors:

- **Colour descriptors:** colour histograms and RGB channel statistics represent the overall colour distribution of the leaf image.
- **Histogram of Oriented Gradients (HOG):** captures local gradient-orientation patterns associated with edges, contours and structural appearance.
- **Local Binary Patterns (LBP):** represents local texture by encoding neighbourhood intensity patterns.
- **Thumbnail representation:** a reduced image representation retains coarse spatial and appearance information.

The feature groups are combined into one numerical vector. The exact histogram-bin configuration and individual descriptor dimensions should be stated in a reproducibility appendix if the implementation code or configuration file is available.

4.5 Feature scaling and SVM classification

The training pipeline applies StandardScaler to the extracted features and then fits a Support Vector Classifier with an RBF kernel. Feature scaling is important because the descriptor combines values with different numerical ranges. The RBF kernel enables a nonlinear decision boundary in the feature space. Balanced class weights are used to reduce the influence of unequal class frequencies during training. Probability estimation is enabled so that the application can display a confidence estimate and rank the three closest class predictions.

The fitted scaler and classifier are saved for later use. At prediction time, the uploaded image is transformed using the same feature-extraction function and the saved scaler before the SVM generates the class probabilities. The predicted class is the class with the highest estimated probability.

4.6 Training and evaluation procedure

The training script reads the labelled image folders, extracts descriptors, validates the data, fits the scaler and SVM, and evaluates the resulting model. The source report records an 80:20 held-out test evaluation, shuffled cross-validation and blocked cross-validation. It also reports class-wise measures and a confusion matrix. A fixed random seed is used to support repeatability. The final saved model is reported as retrained on all 446 images after evaluation.

Figure 2. Model training and evaluation workflow



Figure 2. Model training and evaluation workflow.

4.7 Prediction and confidence-based information display

When a user uploads a photograph, Flask validates the file and passes it to the feature extractor. The saved classifier returns probability estimates for the five known classes. The application compares the highest probability with a configured threshold of 60%. If the highest probability is below 60%, the result is labelled uncertain and medicinal information is withheld. If the probability is at least 60%, the application retrieves the matching record from plant_info.json and displays the plant name, reported uses, parts used, key compounds where available, and precautions. This threshold is an application rule; it has not been independently calibrated and does not guarantee that a prediction is correct.



Figure 3. Prediction-time workflow.

4.8 Medicinal information and safety guidance

The plant-information module retrieves a structured JSON record associated with the predicted class. The record may include common and scientific names, family, a short description, plant parts used, reported traditional uses, key compounds, evidence notes and cautions. The information is presented as educational context, not as an instruction to consume or prepare a plant. The application displays a disclaimer and does not provide dosage, treatment or food-safety recommendations. The source manuscript also notes that common names may refer to different species and that some traditional-use claims have limited evidence.

4.9 Upload validation and application flow

The application accepts JPG, JPEG, PNG and WEBP images up to 8 MB. It verifies that the uploaded content can be opened as an image, assigns a random filename and limits retained uploads to the latest 50 files. Large images are reduced, and a pixel limit is used to protect against unusually large image payloads. Invalid, missing or unreadable files produce an error message rather than being passed to the classifier.

5 Results and performance analysis

The following results are transcribed from the original project report. They describe the reported experiment on the 446-image, five-class dataset.

Table 2. Reported evaluation results.

Evaluation measure	Reported result
Held-out test accuracy (80:20 split)	93.3%
Shuffled cross-validation accuracy	94.8%
Blocked cross-validation accuracy	79.6%
Prediction time	Approximately 13 ms per image
Saved model size	Approximately 9 MB
Functional tests	9 of 9 passed

5.1 Class-wise results

Table 3. Reported class-wise accuracy.

Plant class	Reported class accuracy
Arali	96.6%
Ashoka	91.4%
Aloe vera	65.3%
Amla and Amruthaballi	Separate class-specific values not stated in the source report.

5.2 Discussion

The held-out test accuracy of 93.3% and shuffled cross-validation accuracy of 94.8% indicate strong performance under the reported evaluation settings. However, blocked cross-validation produced a lower score of 79.6%. This difference is important: shuffled splitting can place visually related images in both training and validation partitions, while blocked evaluation separates image groups more strictly. The blocked result therefore provides a more cautious indication of performance on separated image groups.

The 60% confidence rule withheld six of the 90 held-out test images from medicinal-information display. The reported accuracy among answered images increased from 94.4% to 97.6%. This is a selective-prediction result and should not be interpreted as proof that the application can reliably recognise every unknown plant or reject every non-leaf image. The source report records a random-noise input that still received a high-confidence class, demonstrating a limitation of the current confidence rule.

The reported class-wise results also indicate variation between classes. Aloe vera has a lower reported class accuracy than Arali and Ashoka. The limited number of images, class imbalance, background variation and handcrafted feature representation may contribute to these differences. A larger independently collected test set is needed before making claims about field performance.

5.3 Functional testing

The original project report records nine successful functional tests. These cover expected application behaviour, including file validation and rejection of unsafe result URLs. The tests demonstrate that the implemented application follows its intended software workflow; they do not establish clinical safety, botanical certainty or universal classification accuracy.

6 Advantages, limitations and threats to validity

6.1 Advantages

- The application provides a quick, browser-based prediction from a single leaf photograph.
- The feature-extraction function is shared by training and prediction, reducing inconsistencies between the two stages.
- The result page displays a predicted class, confidence estimate and top-three alternatives.
- Medicinal information is withheld when the prediction does not meet the configured confidence threshold.
- The application uses open-source Python libraries and can run locally without a GPU or external inference service.
- Input validation and temporary upload management provide basic application-level safeguards.

6.2 Limitations

- The classifier recognises only five known classes; unknown species may be incorrectly assigned to one of them.
- The dataset contains 446 images and has unequal class counts. Its variation may not represent field conditions.
- Blocked cross-validation is lower than shuffled cross-validation, indicating a generalisation challenge.
- Handcrafted descriptors may be affected by background, illumination, rotation, position, blur and image quality.
- The 60% probability threshold has not been calibrated on a separate validation set and does not reliably reject non-leaf or unknown inputs.
- The plant-information file is manually maintained, is available only in English in the reported implementation, and does not automatically update.
- The application is not a medical tool and must not be used to make treatment, dosage or consumption decisions.
- The Flask development server is not a production deployment configuration; production use would require appropriate hosting and security controls.

6.3 Threats to validity

The reported performance may be affected by the limited dataset size, image similarity across partitions, class imbalance and the choice of evaluation split. The difference between shuffled and blocked cross-validation suggests that a random split alone may overestimate performance on separated image groups. The confidence threshold also affects which cases receive medicinal-information output. Future evaluations should use an independently collected test set with images captured across different devices, locations, backgrounds and collection dates.

7 Conclusion and future work

7.1 Conclusion

This paper presents a lightweight web-based prototype for identifying five medicinal plants from leaf photographs and displaying associated educational information and safety precautions. The implementation uses a 2,852-dimensional feature descriptor combining colour statistics, HOG, LBP and thumbnail information, followed by a scaled RBF-kernel SVM classifier. The classifier is integrated into a Flask application with upload validation, top-three predictions and a confidence-based information display rule. The original project evaluation reports 93.3% held-out test accuracy, 94.8% shuffled cross-validation accuracy and 79.6% blocked cross-validation accuracy. The lower blocked score highlights the need for stronger evaluation on independent field images. The system is a limited educational prototype and cannot replace expert plant identification or professional medical guidance.

7.2 Future enhancements

1. Expand the dataset with more species and photographs captured using different phones, backgrounds, lighting conditions and collection dates.
2. Create an independent test set and use group-based or location-based splitting to reduce similarity between training and test images.
3. Add an explicit unknown-species and non-leaf rejection mechanism and calibrate confidence estimates using a separate validation set.
4. Compare the current SVM pipeline with transfer-learning models such as MobileNetV2 under the same independent evaluation protocol. These are proposed experiments, not results of the current system.
5. Obtain expert review of scientific names, plant descriptions, reported uses and cautions; attach traceable sources to individual information records.
6. Add multilingual interface support and improve mobile-camera usability.
7. Use a production web server, HTTPS, secure configuration, request limits and model versioning before any wider deployment.

References

[1] V. Gautam, G. Kaur, G. S. P. Ghantasala, P. Vidyullatha, S. Allabun, M. Othman, and A. Zheleznyak, "A lightweight deep learning method for medicinal leaf image classification using feature fusion," *Scientific Reports*, vol. 15, art. 34417, 2025. doi: 10.1038/s41598-025-17436-w.

[2] M. M. H. Matin, M. Sefatullah, M. S. Kabir, and H. B. Habib, "An extensive visual data for reliable identification of medicinal plant leaves," *Data in Brief*, vol. 63, art. 112129, 2025. doi: 10.1016/j.dib.2025.112129.

[3] R. Sharma, K. Yadav, M. Siddiqui, and D. Singh, "App-Based Solutions to Detect Medicinal Plants/Crops using Machine Learning," *Journal of Information Systems Engineering and Management*, vol. 10, no. 41s, 2025. doi: 10.52783/jisem.v10i41s.7992.

[4] S. Sachar and A. Kumar, "Levy's flight-based dragonfly optimized machine learning framework to identify medicinal plants through leaf images," Pattern Recognition, vol. 168, art. 111879, 2025. doi: 10.1016/j.patcog.2025.111879.

[5] A. S. K., P. P., G. K., M. K. S., K. Jaganathan, and K. Karthick, "Classification of Medicinal Plant Leaves using Deep Learning Algorithms," *International Journal of Intelligent Systems and Applications*, vol. 18, no. 1, pp. 132–149, 2026. doi: 10.5815/ijisa.2026.01.10.

[6] C. Algemayel, D. Abou Jaoude, S. Talhouk, I. Issa, and C. Ghassibe, "Advances in machine learning models for plant species identification: A scoping review," *Ecological Informatics*, 2026, art. 103464. doi: 10.1016/j.ecoinf.2025.103464.

[7] P. T. Khuat and H. T. Van, "A unified EfficientSwinB-based framework for medicinal plant recognition and safflower germplasm classification," *Scientific Reports*, vol. 16, art. 11791, 2026. doi: 10.1038/s41598-026-42449-4.

[8] R. L. Priya, S. Poriwade, A. Parthasarathy, S. H. Movva, I. Shinde, A. K. Redhu, P. Chandrani, and V. Gota, "FloraMediX: AI-enabled recognition and therapeutic profiling analysis of Indian medicinal flora," *Frontiers in Bioinformatics*, vol. 6, 2026. doi: 10.3389/fbinf.2026.1864201.

[9] "Deep Neural Network for Multi-Class Classification of Medicinal Plant Leaves," *Expert Systems*, 2022. doi: 10.1111/exsy.13041.

[10] "Deep Ensemble Learning for Automatic Medicinal Leaf Identification," *International Journal of Information Technology*, 2022. doi: 10.1007/s41870-022-01055-z.

[11] "Classification of Medicinal Plants Leaves Using Deep Learning Technique: A Review," 2022. Available: https://ijisae.org/index.php/IJISAE/article/view/2199.

[12] "A Novel Computer Vision Model for Medicinal Plant Identification Using Log-Gabor Filters and Deep Learning Algorithms," 2022. doi: 10.1155/2022/1189509.

[13] "Plant Leaf Classification Using Fine-Tuned Transfer Learning," 2022. doi: 10.17762/msea.v71i4.1204.

[14] "Deep Learning for Medicinal Plant Species Classification and Recognition: A Systematic Review," *Frontiers in Plant Science*, 2023. doi: 10.3389/fpls.2023.1286088.

[15] "Ayur-PlantNet: An Unbiased Lightweight Deep Convolutional Neural Network for Indian Ayurvedic Plant Species Classification," *Journal of Ayurveda and Integrative Medicine*, 2023. doi: 10.1016/j.jaim.2023.100459.

[16] "Assessing Deep Convolutional Neural Network Models and Their Comparative Performance for Automated Medicinal Plant Identification from Leaf Images," *Heliyon*, 2023. doi: 10.1016/j.heliyon.2023.e23655.

[17] "Leaf Classification on Flavia Dataset: A Detailed Review," *Sustainable Computing: Informatics and Systems*, 2023. doi: 10.1016/j.suscom.2023.100907.

[18] "Medicinal Plant Leaf Classification Using Deep Learning and Vision Transformers," *Baghdad Science Journal*, 2024. doi: 10.21123/bsj.2024.10844.

[19] "Improved Medicinal Plant Leaf Classification Using Transfer Learning and Model Averaging Ensemble of Deep Convolutional Neural Networks," *IEEE*, 2024. doi: 10.1109/ICITRI62858.2024.10698832.

[20] "PSR-LeafNet: A Deep Learning Framework for Identifying Medicinal Plant Leaves Using Support Vector Machines," *Big Data and Cognitive Computing*, 2024. doi: 10.3390/bdcc8120176.

[21] "Misidentifications in Ayurvedic Medicinal Plants: Convolutional Neural Network (CNN) to Overcome Identification Confusions," *Computers in Biology and Medicine*, 2024. doi: 10.1016/j.compbiomed.2024.109349.

[22] "HybNet: A Hybrid Deep Models for Medicinal Plant Species Identification," *MethodsX*, 2024. doi: 10.1016/j.mex.2024.103126.

[23] "Deep Learning for Plant Identification and Disease Classification from Leaf Images: Multi-Prediction Approaches," *ACM*, 2024. doi: 10.1145/3639816.