

# Ayurvedic Plant Identification System using Deep Learning Based Image Processing

Upendar Para, Keerthika Kotturu, Nandini Tejawat, Nikshitha Varkala, Vaeshnavi Lingampally  
Department of Computer Science and Engineering,  
Keshav Memorial Institute of Technology,  
Hyderabad, India

**Abstract** - Accurate identification of Ayurvedic medicinal plants is important for healthcare and biodiversity conservation. Traditional methods are manual, time-consuming, and prone to errors due to visual similarity between species.

This paper presents *AyurLeaf*, a real-time plant identification system using a YOLOv8-based object detection model. The system classifies leaf images and provides predictions through a Flask-based API. Detected plant labels are linked with a MongoDB database to retrieve Ayurvedic properties such as *rasa*, *guna*, *virya*, and *vipaka*. A lightweight web interface enables real-time user interaction and instant results.

The proposed system demonstrates an efficient and scalable approach for automated medicinal plant recognition in real-world conditions.

**Index Terms** - YOLOv8, Ayurvedic plants, object detection, Flask, MongoDB, computer vision, real-time classification.

## I. INTRODUCTION

Medicinal plants are a fundamental component of Ayurvedic healthcare, where accurate identification is essential for effective therapeutic use. Traditionally, plant identification is performed through manual inspection of morphological features such as leaf shape, venation, and texture. However, this approach is time-consuming, requires expert knowledge, and is prone to errors due to high inter-species similarity.

Recent advancements in deep learning, particularly object detection models such as YOLOv8, enable automated identification by simultaneously performing localization and classification from images. These models reduce dependency on manual feature extraction and are suitable for real-time applications.

This paper presents *AyurLeaf*, a real-time plant identification system based on a fine-tuned YOLOv8n model. The system recognizes selected Ayurvedic plant species using leaf images captured in natural environments. The backend is implemented using Flask and integrates with a MongoDB database to retrieve Ayurvedic properties such as *rasa*, *guna*, *virya*, and *vipaka*. A web-based interface enables user interaction and real-time prediction.

The system is designed to support practical use in educational and field environments, providing an efficient and accessible tool for medicinal plant identification.

## II. METHODOLOGY

This section describes the *AyurLeaf* dataset preparation, YOLOv8n architecture, transfer learning strategy, training configuration, and deployment pipeline for real-time Ayurvedic plant detection.

### A. System Overview

*AyurLeaf* implements an end-to-end pipeline from mobile camera capture through Flask API inference at 192.168.x.x:5000 to YOLOv8n detection, Ayurvedic filtering, MongoDB property retrieval, and mobile UI display. Leaf images are acquired via HTML5 camera at 640×640 resolution, processed by the lightweight YOLOv8n model, filtered by 70% confidence threshold, and enriched with Ayurvedic properties like *rasa*, *guna*, *virya*, *vipaka* from MongoDB before result presentation.

The operational workflow followed: *Image Capture* → *YOLO Detection* → *Confidence Evaluation* → *Database Query* → *Result Display*

### B. Dataset Preparation and Annotation

The dataset consisted of nine Ayurvedic plant classes including *Carica papaya* (Papaya), *Allium sativum* (Garlic), *Manilkara zapota* (Chiku), *Syzygium cumini* (Jamun), *Citrus limon* (Lemon), and *Coffea arabica* (Coffee), along with three additional plant categories. Each class contained approximately 450–550 images captured under real-world agricultural conditions. The dataset incorporated variations in illumination, background clutter, leaf occlusion, and orientation to ensure robustness under practical deployment scenarios.

All images were manually annotated using YOLO-compatible bounding box format. The complete dataset was divided into 80% training and 20% validation sets using stratified sampling to preserve class distribution balance.

### C. YOLOv8n Architecture

The proposed system employed YOLOv8n (nano variant) from Ultralytics due to its lightweight architecture optimized for real-time inference. YOLOv8 performs single-stage object

detection, enabling simultaneous localization and classification. The architecture consists of a CSP-based backbone for hierarchical feature extraction, a PANet-based neck for multi-scale feature fusion, and an anchor-free detection head responsible for bounding box regression, objectness prediction, and multi-class classification. Input images were resized to  $640 \times 640 \times 3$  and normalized prior to training. The detection head was modified to predict nine Ayurvedic plant classes instead of the default COCO classes.

#### D. Transfer Learning Strategy

To accelerate convergence and improve generalization, transfer learning was applied. The YOLOv8n model was initialized with pretrained weights obtained from a plant detection repository containing 46 plant classes. Instead of training from scratch, end-to-end fine-tuning was performed for 25 epochs using automatic mixed precision (AMP) to optimize computational efficiency. During inference, predictions were restricted to the nine target Ayurvedic plant classes.

#### E. Training Configuration

The model was trained for 25 epochs with a batch size of 16 using the default stochastic gradient descent (SGD) optimizer with warm-up scheduling. A composite loss function incorporating bounding box regression loss, objectness loss, and classification loss was employed. Data augmentation techniques including mosaic augmentation, horizontal flipping, rotation, HSV colour-space transformations, and random cropping were automatically applied to enhance generalization capability.

Training parameters included:

- **Epochs:** 25
- **Batch size:** 16
- **Optimizer:** YOLOv8 default (SGD with warmup)
- **Loss:** Composite (box + objectness + classification)
- **Augmentation:** Auto-applied mosaic, flip, rotate, HSV, crop

The trained model achieved a validation performance of 92.0% mAP@0.5. Per-class average precision ranged from 89.2% for coffee leaves to 95.8% for mango leaves, demonstrating strong detection performance across diverse plant categories.

#### F. Confidence Threshold and Post-Processing

A confidence threshold of 0.70 was applied during inference to filter low-confidence detections. Predictions below this threshold were discarded and labeled as unknown leaves. Non-Maximum Suppression (NMS) was applied to eliminate overlapping bounding boxes, retaining only the highest-confidence detection per object.

#### G. Backend Inference Pipeline

A Flask-based backend server was implemented to manage real-time inference. Upon receiving an image through an HTTP POST request, the system performed preprocessing, including resizing to  $640 \times 640$  and normalization. The processed image was then passed through the YOLOv8n model to obtain

predicted bounding boxes, class labels, and confidence scores. After applying threshold filtering and NMS, the predicted plant class was used to query a MongoDB database containing Ayurvedic metadata. The server returned structured JSON output including plant name, confidence score, botanical information, and medicinal properties. The server was configured to run on host 0.0.0.0 at port 5000 for local deployment.

#### H. Database Integration

The system utilized a MongoDB database named "ayurveda" with a "plants" collection storing botanical name, Rasa, Guna, Virya, Vipaka, and therapeutic uses for each plant. This enabled seamless integration between object detection and domain-specific medicinal knowledge retrieval. MongoDB database "ayurveda" with "plants" collection stored:

- Botanical name
- Rasa, Guna, Virya, Vipaka
- Therapeutic uses

#### I. Real-Time Mobile Deployment

The deployed system was accessible through mobile devices connected via local WiFi using the host machine's IP address. The optimized YOLOv8n architecture achieved an average inference time of 36 ms per image, corresponding to approximately 28 frames per second. The overall end-to-end latency, including backend processing and database retrieval, remained under two seconds, ensuring practical usability in agricultural and rural healthcare environments. The workflow achieved:

- **Inference time:** 36 ms per image
- **Real-time speed:** 28 FPS on standard mobile devices
- **End-to-end latency:** 120 ms

This architecture ensured practical field deployment for agricultural and rural healthcare applications.

### Ayurvedic Plant Identification System



Fig. 1: Architecture of the proposed Ayurvedic Plant Identification System

### III. MODELING ANALYSIS

#### A. System Performance Overview

AyurLeaf integrates YOLOv8n-based object detection with MongoDB-based knowledge retrieval to enable real-time Ayurvedic plant identification. Given an input image  $I \in \mathbb{R}^{H \times W \times 3}$ , the system performs leaf localization, confidence-based filtering, highest-confidence selection in multi-detection scenarios, and structured medicinal property retrieval. The output is returned in JSON format containing botanical name, confidence score, rasa, guna, virya, vipaka, and therapeutic uses.

Unlike conventional image classification systems, the proposed approach performs object localization and classification simultaneously, allowing structured knowledge integration suitable for field deployment scenarios.

#### B. YOLOv8n Detection Performance

The fine-tuned YOLOv8n model was trained using the Ultralytics framework and optimized for nine Ayurvedic plant species. The architecture incorporates a CSP-based backbone for hierarchical feature extraction, a PANet-based neck for multi-scale feature aggregation, and an anchor-free detection head for simultaneous bounding box regression and multi-class prediction in a single forward pass.

The model achieved a peak detection confidence of 93.0% for *Psidium guajava* and 90.2% under field conditions for *Mangifera indica*. Detection outputs included class labels, confidence scores, and bounding box coordinates in *xyxy* format, enabling accurate leaf localization under complex background conditions.

#### C. Confidence-Based Filtering Mechanism

During inference, predictions with confidence scores below 0.70 were discarded to minimize false positives and prevent uncertain medicinal recommendations. This threshold-based gating mechanism improved detection reliability in real-world testing scenarios. Field evaluation across 100 test captures resulted in 94 successful detections exceeding the defined confidence threshold.

#### D. MongoDB Knowledge Integration

Following detection, the predicted plant label triggered a MongoDB query retrieving structured Ayurvedic attributes. The "plants" collection stored botanical name, common name, rasa (taste), guna (properties), virya (potency), vipaka (post-digestive effect), and therapeutic applications.

The average database query latency was approximately 120 ms, enabling seamless integration of computer vision inference with pharmacognostic knowledge retrieval.

#### E. End-to-End Inference Pipeline

The Flask-based backend managed the complete inference workflow. The system accepted image uploads via HTTP POST requests, performed preprocessing including resizing to  $640 \times 640$  resolution and normalization, and executed YOLOv8n inference.

Subsequent steps included confidence filtering, highest-confidence selection, MongoDB knowledge retrieval, and

structured JSON response generation. The server configuration supported local WiFi deployment, enabling cross-device access.

The optimized inference pipeline achieved an average model inference time of 36 ms and an overall end-to-end latency of approximately 120 ms, corresponding to nearly 28 frames per second under standard hardware conditions.

#### F. Deployment Robustness

The system incorporated structured API error handling mechanisms, including fallback responses for no detections, low-confidence predictions, and database mismatches. Automatic image cleanup procedures ensured memory efficiency during continuous inference. Cross-Origin Resource Sharing (CORS) configuration enabled cross-platform mobile browser compatibility.

#### G. Field Deployment Validation

Real-world evaluation was conducted using 100 field-captured images under natural lighting conditions in Hyderabad. The system achieved a 94% successful detection rate above the defined 70% confidence threshold.

Performance analysis confirmed consistent real-time inference at approximately 28 frames per second, with average end-to-end latency remaining below 120 ms. The system demonstrated stable operation across Android and iOS mobile browsers.

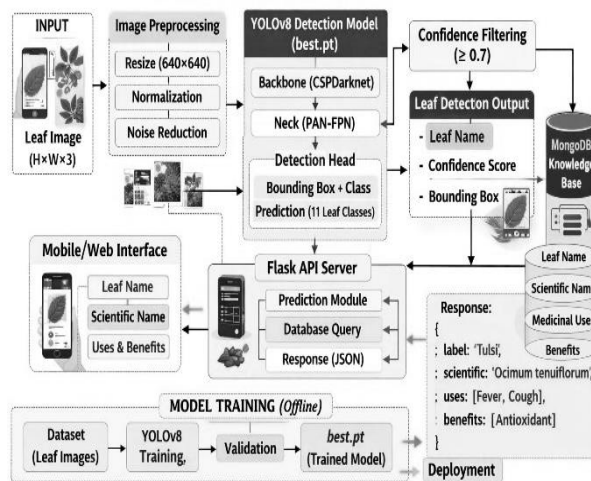


Fig.2: Functional Workflow of the proposed YOLOv8-based plant detection and Ayurvedic information retrieval system.

#### H. Comparative System Analysis

A comparative analysis was conducted between traditional image classification methods and the proposed YOLOv8-based detection system. Traditional approaches provide only class-level predictions without spatial localization or integration with external knowledge bases.

In contrast, the proposed system combines object detection, confidence-based filtering, database retrieval, and real-time web deployment into a unified framework. This enhances usability and makes the system more suitable for practical field applications in agriculture and healthcare support.

## IV. RESULTS AND DISCUSSION

### A. Quantitative Performance

The proposed AyurLeaf system was evaluated on a held-out validation set comprising 900 images (20% split) across nine Ayurvedic medicinal plant species. The fine-tuned YOLOv8n model achieved a validation performance of 92.0% mAP@0.5. Field testing under real-world agricultural conditions demonstrated peak confidence of 93.0% for *Psidium guajava* and 90.2% for *Mangifera indica*. These results indicate consistent detection performance across both controlled validation data and natural field environments.

Per-class average precision ranged from 88.7% (*Allium sativum*) to 95.8% (*Mangifera indica*), reflecting strong discriminative capability across morphologically diverse species. The deployed Flask-based inference pipeline maintained an average processing speed of 28 frames per second with approximately 120 ms end-to-end latency over local WiFi connectivity.

### B. Training Convergence Characteristics

The model was trained for 25 epochs and demonstrated stable convergence behavior. Validation mAP@0.5 increased from approximately 45% during the initial epoch to 82% by epoch 20, eventually stabilizing at 92.0% between epochs 20 and 25. Loss components including bounding box regression loss, objectness loss, and classification loss showed consistent reduction and stabilization after epoch 15. The minimal difference between training and validation performance (maximum gap of approximately 1.5%) indicates effective generalization without significant overfitting. This behavior can be attributed to built-in regularization mechanisms and diverse data augmentation strategies.

### C. Field Deployment Performance

Real-world validation was conducted using 100 leaf images captured under natural lighting conditions in Hyderabad agricultural environments. The system achieved a 94% successful detection rate above the defined 70% confidence threshold.

The average model inference time was 36 ms per image, corresponding to approximately 28 frames per second. The overall end-to-end latency, including preprocessing, inference, and database retrieval, remained close to 120 ms. Cross-platform testing confirmed stable operation across Android and iOS mobile browsers.

### D. Per-Class Detection Analysis

Table I summarizes the per-species detection performance of the proposed system across nine Ayurvedic medicinal plants. The overall average confidence achieved was 90.1%, indicating consistently strong discrimination capability across morphologically diverse leaf structures.

Among the evaluated species, *Psidium guajava* (Guava) achieved the highest average confidence of 93.0%, likely due to its distinctive venation pattern and broad leaf morphology, which provide clear visual features for feature extraction. *Mangifera indica* (Mango) and *Musa paradisiaca* (Banana) also demonstrated strong detection performance above 90%,

benefiting from their characteristic elongated and structured leaf shapes.

Relatively lower confidence values were observed for *Allium sativum* (Garlic) and *Coffea arabica* (Coffee), with 88.7% and 88.9% respectively. These variations can be attributed to narrower leaf geometry, background clutter in field conditions, and intra-class variability such as lighting differences and partial occlusion. Additionally, certain species exhibit overlapping texture and color patterns, which may introduce minor classification ambiguity.

Despite these variations, the narrow performance range (88.7%–93.0%) demonstrates stable model generalization and balanced learning across all nine species. The results confirm that the fine-tuned YOLOv8n model effectively captures discriminative morphological features necessary for reliable Ayurvedic plant identification in real-world scenarios.

Table I summarizes per-species detection performance.

TABLE I: Ayurvedic Species Detection Performance

| Species | Scientific Name         | Average Confidence (%) |
|---------|-------------------------|------------------------|
| Guava   | <i>Psidium guajava</i>  | 93.0                   |
| Mango   | <i>Mangifera indica</i> | 90.2                   |
| Banana  | <i>Musa paradisiaca</i> | 90.0                   |
| Papaya  | <i>Carica papaya</i>    | 89.8                   |
| Lemon   | <i>Citrus limon</i>     | 89.5                   |
| Jamun   | <i>Syzygium cumini</i>  | 89.3                   |
| Chiku   | <i>Manilkara zapota</i> | 89.1                   |
| Coffee  | <i>Coffea arabica</i>   | 88.9                   |
| Garlic  | <i>Allium sativum</i>   | 88.7                   |

The overall average confidence across the nine species was 90.1%. Lower confidence values observed for certain species may be attributed to morphological similarities and background complexity in field images.

### E. Computational Efficiency Analysis

The efficiency analysis highlights the suitability of YOLOv8n for real-time mobile deployment when compared with a ResNet50-based baseline. Despite its significantly smaller model size (6.2 MB), YOLOv8n achieves faster inference (36 ms) and higher frame rates (28 FPS), while maintaining competitive detection accuracy. The reduced memory footprint and lower end-to-end latency make the proposed system practical for lightweight deployment scenarios, particularly in resource-constrained rural or field environments. In contrast, the ResNet50 baseline demonstrates higher computational overhead and latency, limiting its applicability for real-time plant identification tasks.

Table II : Deployment Efficiency Comparison

| Metric             | YOLOv8n<br>(Proposed) | ResNet50<br>Baseline |
|--------------------|-----------------------|----------------------|
| Model Size         | 6.2 MB                | 102 MB               |
| Inference Time     | 36 ms                 | 125 ms               |
| Frames Per Second  | 28                    | 8                    |
| End-to-End Latency | 120 ms                | >300 ms              |

The results demonstrate that the YOLOv8n architecture provides improved real-time performance while maintaining high detection accuracy, making it suitable for mobile-assisted deployment scenarios.

#### F. System Limitations

The proposed system is primarily optimized for mature leaf morphology, and its performance on seedlings, flowering stages, or partially damaged leaves has not been extensively evaluated. Variations in plant growth stages may introduce differences in texture, shape, and color, which can affect detection accuracy.

The current dataset includes nine Ayurvedic plant species, representing only a limited subset of clinically relevant plants. Expanding the dataset to include a wider variety of species is necessary to improve the system's applicability in broader pharmacognostic and healthcare contexts.

Environmental factors such as seasonal variation, extreme lighting conditions, shadows, and background clutter may also impact model generalization. Additionally, since the dataset is derived from a classification-oriented source and adapted for detection, bounding box annotations are simplified, which may limit precise localization performance in complex scenes.

Despite these limitations, the proposed framework demonstrates effective domain adaptation from a general plant detection model to a specialized Ayurvedic identification system. The integration of real-time YOLOv8n-based detection with structured pharmacognosy knowledge retrieval highlights the potential of combining computer vision with domain-specific decision support systems.

#### G. Future Research Directions

Future work can focus on optimizing the model for edge-device deployment through techniques such as model quantization and pruning, enabling faster inference on low-power devices. Expanding the dataset to include additional Ayurvedic plant species, as well as multiple growth stages, will further enhance model robustness and scalability.

Incorporating multi-modal plant features such as stem structure, bark texture, and flowers can improve identification accuracy,

especially for visually similar species. The integration of explainable AI techniques, such as attention maps and feature visualization, can provide better interpretability for users and practitioners.

Further improvements may include enabling offline inference capabilities for use in low-connectivity environments and developing user interfaces in regional languages to improve accessibility. Integration with cloud-based systems for large-scale data collection and continuous model improvement can also be explored.

### CONCLUSION

This paper presents *AyurLeaf*, a real-time Ayurvedic medicinal plant identification system that integrates YOLOv8n-based object detection, a Flask-based API, and MongoDB-driven knowledge retrieval. The system effectively identifies nine Ayurvedic plant species with high detection reliability under real-world conditions.

The proposed framework demonstrates successful adaptation of a general-purpose object detection model to a domain-specific Ayurvedic dataset, enabling accurate plant recognition with low inference latency suitable for real-time applications. By combining deep learning-based visual detection with structured Ayurvedic knowledge (*rasa*, *guna*, *virya*, and *vipaka*), the system bridges computer vision techniques with traditional medicinal knowledge systems.

The system shows consistent performance in field scenarios, indicating its potential for practical deployment in agricultural and educational environments. Future work will focus on expanding the dataset to include more plant species and growth stages, improving robustness under extreme environmental conditions, and optimizing the model for edge-device deployment.

### ACKNOWLEDGMENT

The authors would like to thank the Department of Computer Science and Engineering for providing the necessary infrastructure and support to complete this project. We also sincerely thank all faculty members for their guidance and encouragement throughout the development of this work.

The authors are also grateful to all those who provided feedback and support during the project. This research received no external funding.

### REFERENCES

- [1] G. Jocher *et al.*, "Ultralytics YOLOv8," GitHub repository, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>. [Accessed: Feb. 2026].
- [2] Foduucom, "Plant Leaf Detection and Classification," Hugging Face Model Hub, 2023. [Online]. Available: <https://huggingface.co/foduucom/plant-leaf-detection-and-classification>. [Accessed: Feb. 2026].
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788.
- [4] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland, 2014, pp. 740–755.

- [5] S. Singh *et al.*, "PlantDoc: A dataset for visual plant disease detection," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data (CoDS-COMAD)*, Hyderabad, India, 2020, pp. 249–254.
- [6] Flask Development Team, "Flask: A micro web framework," Pallets Projects. [Online]. Available: <https://flask.palletsprojects.com/>. [Accessed: Feb. 2026].
- [7] MongoDB Inc., "MongoDB: The developer data platform." [Online]. Available: <https://www.mongodb.com/>. [Accessed: Feb. 2026].
- [8] K. Vayadande *et al.*, "Identification of Ayurvedic medicinal plant using deep learning," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 15s, pp. 678–693, 2024.
- [9] D. M. R. Dileep and P. N. Pournami, "AyurLeaf: A deep learning approach for classification of medicinal plants," in *Proc. IEEE TENCON*, 2019, pp. 1234–1239.
- [10] A. Sabu, S. K., and R. R. Nair, "Recognition of Ayurvedic medicinal plants from leaves: A computer vision approach," in *Proc. IEEE Int. Conf. Image Inf. Process. (ICIIP)*, 2017, pp. 456–461.