

Machine Learning Semester Project

Group Members:

Syeda Rabbiya Bukhari-bsdsf22a044

Alisha Shahid-bsdsf22a005

Hajra Amir-bsdsf22a027

Plantly: A Smart Plant Care Companion

1. Problem Statement:

Many plant owners struggle to identify the plants they own or understand their care requirements. Misinformation or neglect often leads to poor plant health. There is a clear need for a simple, intelligent, and accessible plant care assistant that can identify a plant from an image and provide personalized care instructions all in real time.

2. Objectives:

- Train an ML model that can classify common household plants from user-uploaded images.
- Develop a backend API that handles image uploads and returns predictions with care data.
- Build a React-based frontend that enables users to interact with the model through a friendly UI.
- Ensure the system is fast, reliable, and easy to use.

3. Proposed Methodology:

Frontend (React):

- Upload plant image via drag/drop or file picker
- Display identified plant and its basic care tips (light, water, soil)
- Responsive, minimal UI for mobile/desktop

Backend (Flask or FastAPI):

- Expose REST API for:
 - Receiving image
 - Returning predicted plant type
 - Serving associated care information
- Handle inference requests via Python (model loaded on server)

ML Model:

- Use **transfer learning** with a lightweight CNN like **MobileNetV2** or **ResNet18**
- Train on a **limited dataset (10–20 classes)** for feasibility
- Output top prediction + confidence

Care Info Mapping:

Manually create a **plant-care JSON/CSV** linking each class to its needs

```
{  
  "Aloe Vera": {  
    "Light": "Bright, indirect sunlight",  
    "Water": "Every 2–3 weeks",  
    "Soil": "Well-draining cactus mix"  
  }  
}
```

4. Dataset Description:

- **Dataset:** A subset from PlantCLEF, PlantNet, or Kaggle’s Plant Seedlings dataset.
- **Classes:** Limit to ~10–20 most common houseplants.
- **Image count:** 100–200 images per class.
- Care information collected from reputable gardening sources (The Spruce, RHS, etc.).

5. Expected Outcomes:

- Accurate plant classification from images ($\geq 85\%$ on limited classes).
- React-based web interface for public use.
- RESTful backend serving real time ML predictions.
- Static or dynamic plant care guidance.
- Optional: show top 3 predictions (e.g., “Did you mean X or Y?”) .

6. Timeline (2 Weeks)

Days	Task
1–2	Dataset finalization, cleanup, model selection
3–5	Model training & evaluation (MobileNetV2/ResNet)
6	Prepare plant care JSON file
7–9	Build and test backend API with ML integration
10–12	Design and build React frontend
13	Full-stack integration & testing. Final polishing, UI tweaks, and documentation