Dual-Model Poisonous Plant Identification System with Custom CNN and Gemini 1.5 Flash API Integration

This project presents a robust and educational Python-based graphical user interface (GUI) application designed to identify poisonous plants through a dual-model system. It combines a custom-built Convolutional Neural Network (CNN) developed from scratch with the powerful multimodal AI capabilities of Google's Gemini 1.5 Flash API. By utilizing two different AI paradigms, the application enhances the reliability of its plant classification results and enables users to compare model outputs in real-time.

The primary objective of this system is to assist users in distinguishing between poisonous and non-poisonous plant species based on image inputs. The application accepts uploaded plant images and processes them through both the in-house CNN and the Gemini API, providing users with two parallel results: one from a traditional computer vision pipeline and another from an advanced, cloud-based AI model. The results are displayed side-by-side along with visual cues, confidence scores, and detailed descripti...

## The custom CNN is implemented without the use of high-level deep learning frameworks such as TensorFlow or PyTorch. Instead, it is built from the ground up using NumPy, allowing for full transparency into each mathematical operation and layer behavior. The architecture includes:

* - Five 3×3 convolution filters designed to detect specific visual features (e.g., red and green color intensities, texture patterns, and edge orientations)
* - ReLU activation for non-linearity
* - 2×2 max pooling for dimensionality reduction
* - Two fully connected layers (one hidden layer with 64 neurons and an output layer with softmax for multi-class classification)

This architecture supports classification across 8 distinct plant categories—four poisonous (Poison Ivy, Poison Oak, Poison Sumac, Water Hemlock) and four non-poisonous (Dandelion, Clover, Mint, Sunflower). Extensive feature extraction is conducted, including color histogram analysis in RGB and HSV color spaces, contour-based shape analysis (e.g., circularity, perimeter), texture metrics, and Canny edge detection.

## Gemini 1.5 Flash integration is achieved through base64-encoded image transmission to the Google Generative Language API. The system includes well-crafted prompt engineering to instruct Gemini to:

* - Identify the plant species
* - Determine if the plant is poisonous
* - Provide confidence levels
* - Deliver detailed descriptions and reasoning

Structured parsing routines then interpret the natural language response from Gemini and extract key elements like plant name, toxic status, toxicity explanation, and confidence percentage.

## The application is constructed using:

* - Python 3.x
* - tkinter (8.6+) for GUI
* - PIL/Pillow (8.0+) for image rendering and manipulation
* - OpenCV (4.5+) for feature extraction and preprocessing
* - NumPy (1.19+) for matrix and filter operations
* - scikit-learn (0.24+) for model evaluation utilities
* - requests (2.25+) for API communication

Key Features

* - Custom CNN-based Image Classification:
* - Entirely built from first principles using NumPy
* - Feature detection filters manually designed
* - Multi-layer architecture with ReLU and softmax
* - Offline plant identification capability
* - Gemini 1.5 Flash API Integration:
* - Cloud-based multimodal AI with image understanding
* - Structured prompt engineering for toxic assessment
* - Extracts detailed reasoning and toxicity levels
* - User Interface:
* - Dual identification buttons for CNN and Gemini
* - Color-coded output (Red = Poisonous, Green = Safe, Purple = Gemini)
* - Real-time progress bar and result area with scroll capability
* - Visual feedback through confidence scores and model descriptions
* - Advanced Feature Extraction:
* - RGB & HSV histogram analysis
* - Shape descriptors: contour-based circularity, compactness
* - Edge detection: Canny and Sobel filters
* - Adaptive resizing and preprocessing pipeline
* - Error Handling and Robustness:
* - File format validation
* - API failure detection and retry logic
* - Graceful handling of image load issues
* - CNN model state can be saved and reloaded
* - Educational Value:
* - Clear visualization of how handcrafted filters work
* - Comparison between classic ML and commercial-grade AI
* - Ideal for teaching computer vision, deep learning basics, and AI ethics

Educational and Practical Significance

## This hybrid framework illustrates the synergy between custom algorithm design and cutting-edge cloud AI services. It is ideal for:

* - Educators and learners exploring foundational machine learning and AI integration
* - Field researchers and botanists needing offline or online tools for rapid plant identification
* - General public and hikers for real-time toxic plant recognition and risk prevention

Future Enhancements (Planned)

* - Add support for real-time webcam input
* - Expand plant database with crowdsourced annotations
* - Integrate TensorFlow/PyTorch for future CNN versions
* - Include map-based geolocation for plant sightings
* - Enable offline cache of Gemini API results