GreenClassify : Deep Learning-Based Approach For Vegetable Image Classification

Student Details

- **Name:** Anuja Bhikaji Chavan
- **Roll No:** 42
- **PRN:** 2022011031016
- **Institution:** D.Y. Patil Agricultural and Technical University, Talsande

---

1. Project Overview

- **Goal:** To classify vegetable images into multiple predefined categories using Deep Learning, specifically Convolutional Neural Networks (CNN).
- **Problem Type:** Multi-Class Classification (Supervised Deep Learning).
- **Tech Stack:** Python, TensorFlow, Keras, Flask, NumPy, Matplotlib, and HTML/CSS.

2. Data Collection & Preparation

- **Data Source:** A structured dataset containing categories such as Bean, Bottle Gourd, Bitter Gourd, Tomato, Carrot, and Broccoli.
- **Organization:** The dataset is divided into three main folders: **train**, **validation**, and **test**.
- **Preprocessing Steps:**
    - **Rescaling:** Pixel values were normalized between 0 and 1 using rescale = 1./255.
    - **Image Resizing:** Images were resized to **128x128** for model training and **150x150** for visualization.
    - **Data Augmentation:** To prevent overfitting, the training data

used zoom_range=0.2, shear_range=0.2, and horizontal_flip=True.

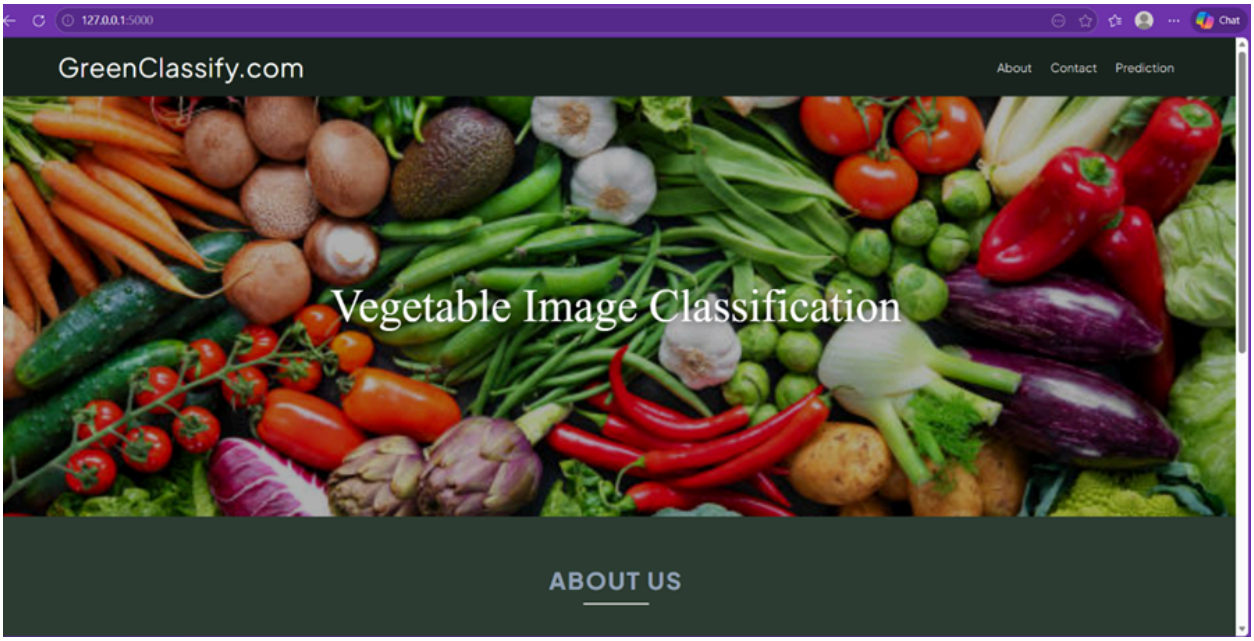
3. Model Building

- **Architecture:** A Sequential CNN model with the following layers:
    - **Conv2D (16 filters, 3x3)** + MaxPooling.
    - **Conv2D (32 filters)** + MaxPooling.
    - **Conv2D (64 filters)** + MaxPooling.
    - **Flatten & Dense:** A 128-neuron dense layer.
    - **Output Layer:** Dense layer with Softmax activation for multi-class prediction.

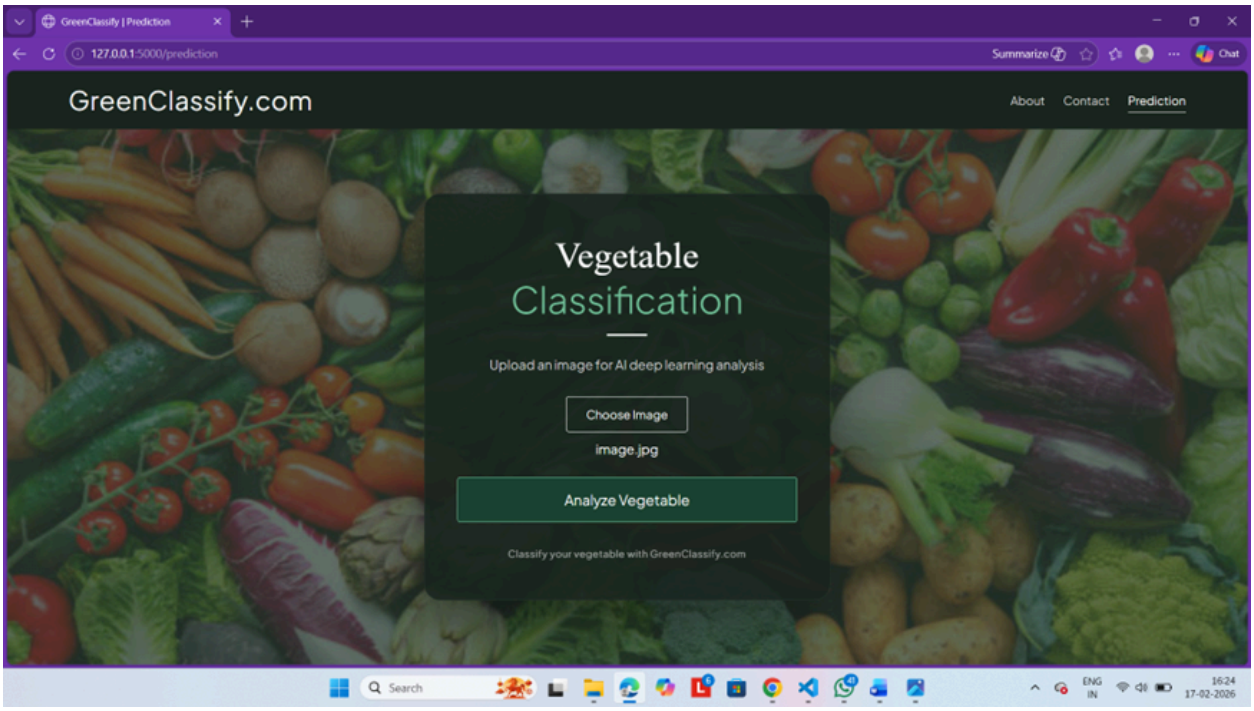
- **Compilation:** Optimized with Adam (learning rate: 0.0001) and Categorical Crossentropy loss.
- **Training:** Conducted over 3 Epochs with a batch size of 32.

4. Model Deployment

- **Web Framework:** Deployed using **Flask**.
- **Structure:**
    - **Backend (app.py):** Loads the model (vegetable_model.h5), handles image uploads, and performs inference.
    - **Frontend:** index.html (Home), prediction.html (Upload), and a results page.

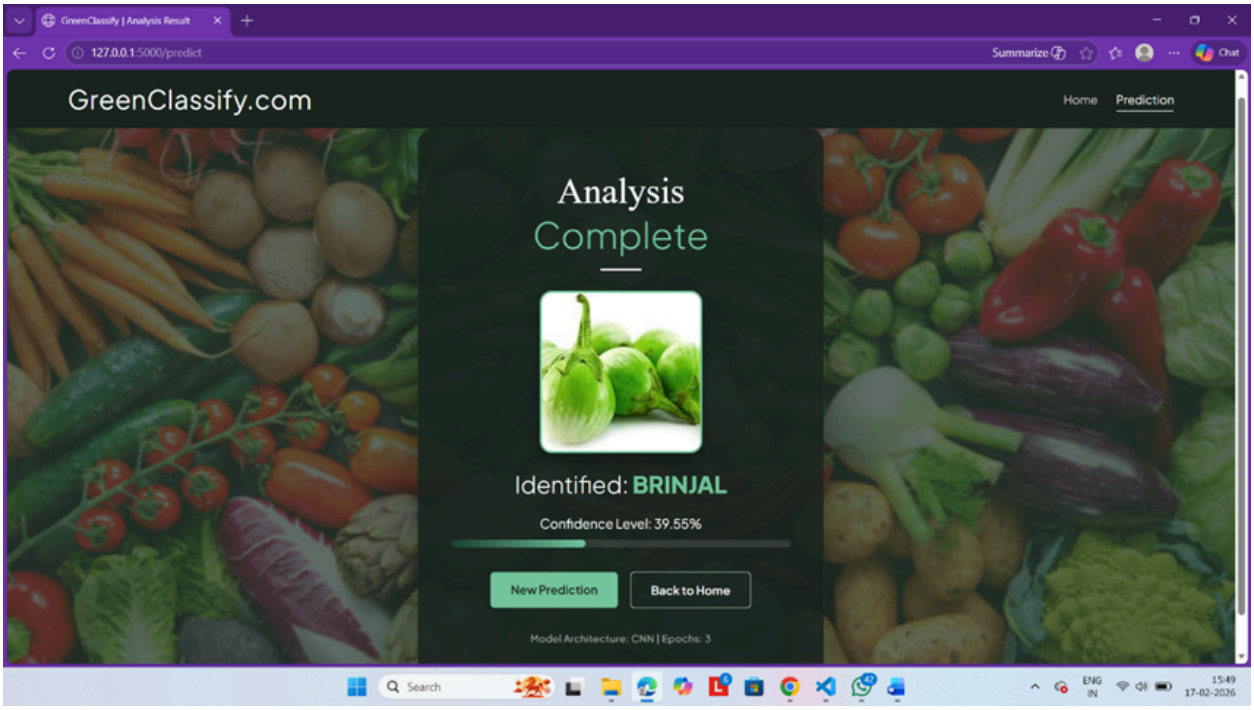


- **Prediction Logic:**
  - predictions = model.predict(img_array).
  - class_index = np.argmax(predictions).
  - confidence = float(np.max(predictions)) * 100.0 (Corrected typo from "1006.").



5. Results and Applications

- **Observations:** Training accuracy improved over epochs, though similar-looking vegetables (like Brinjal and Green Chilli) may slightly reduce confidence levels.



- **Real-World Applications:**
    - Automated sorting in processing facilities.
    - Quality control in agricultural supply chains.
    - Smart shelf management in retail.
- **Limitations:** Requires high-quality images and a larger dataset to improve accuracy for similar vegetable types.