Fruits Recognition

Image processing

Nouran Hassan Ahmed 2022/00062

TABLE OF CONTENTS

DATASET

- Here is the link of the dataset that I have chosen:

  https://www.kaggle.com/datasets/utkarshsaxenadn/fruits-classification

Dataset Properties

This dataset consists of images of various fruits and vegetables, offering a rich collection for image recognition tasks.

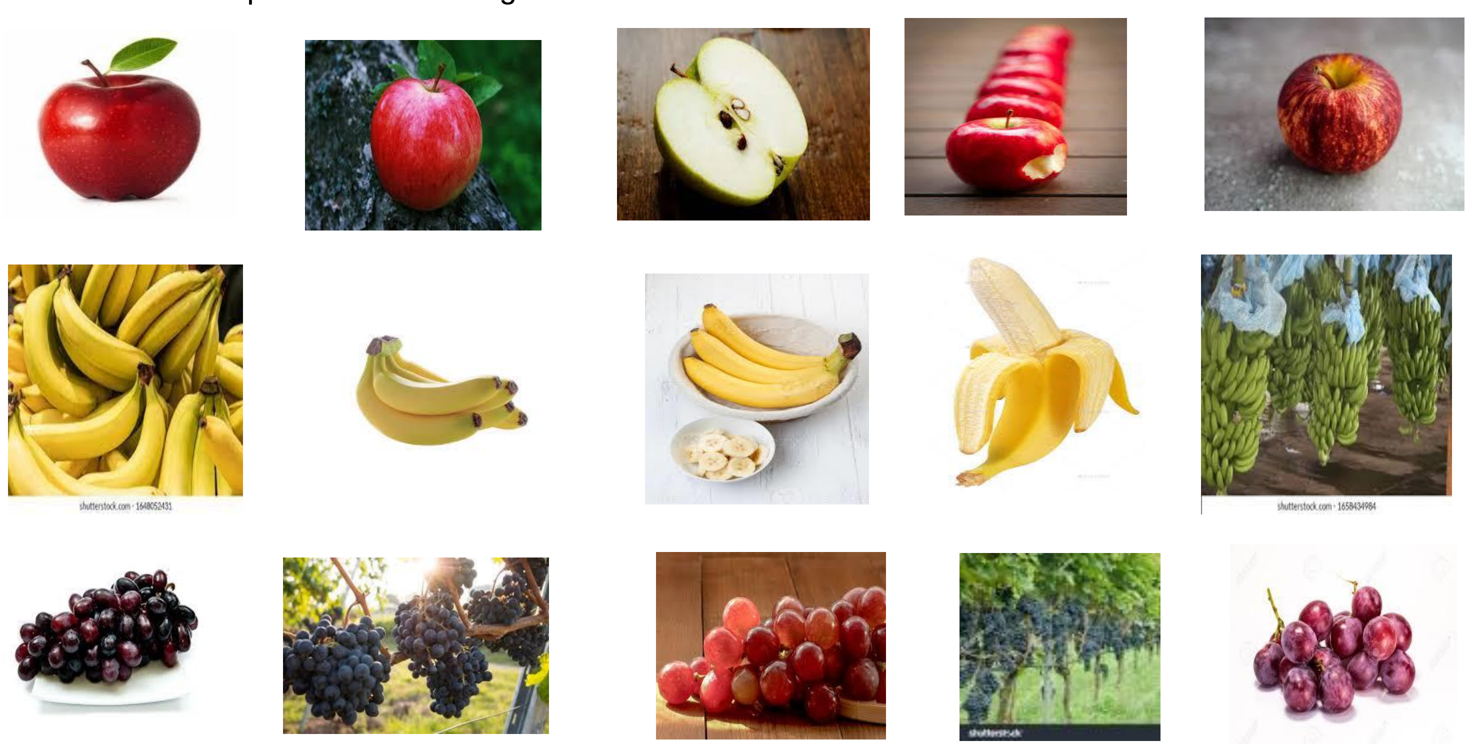
Categories

1. **Fruits :** Banana, Apple, Grapes, Strawberry, Mango

Dataset Organization

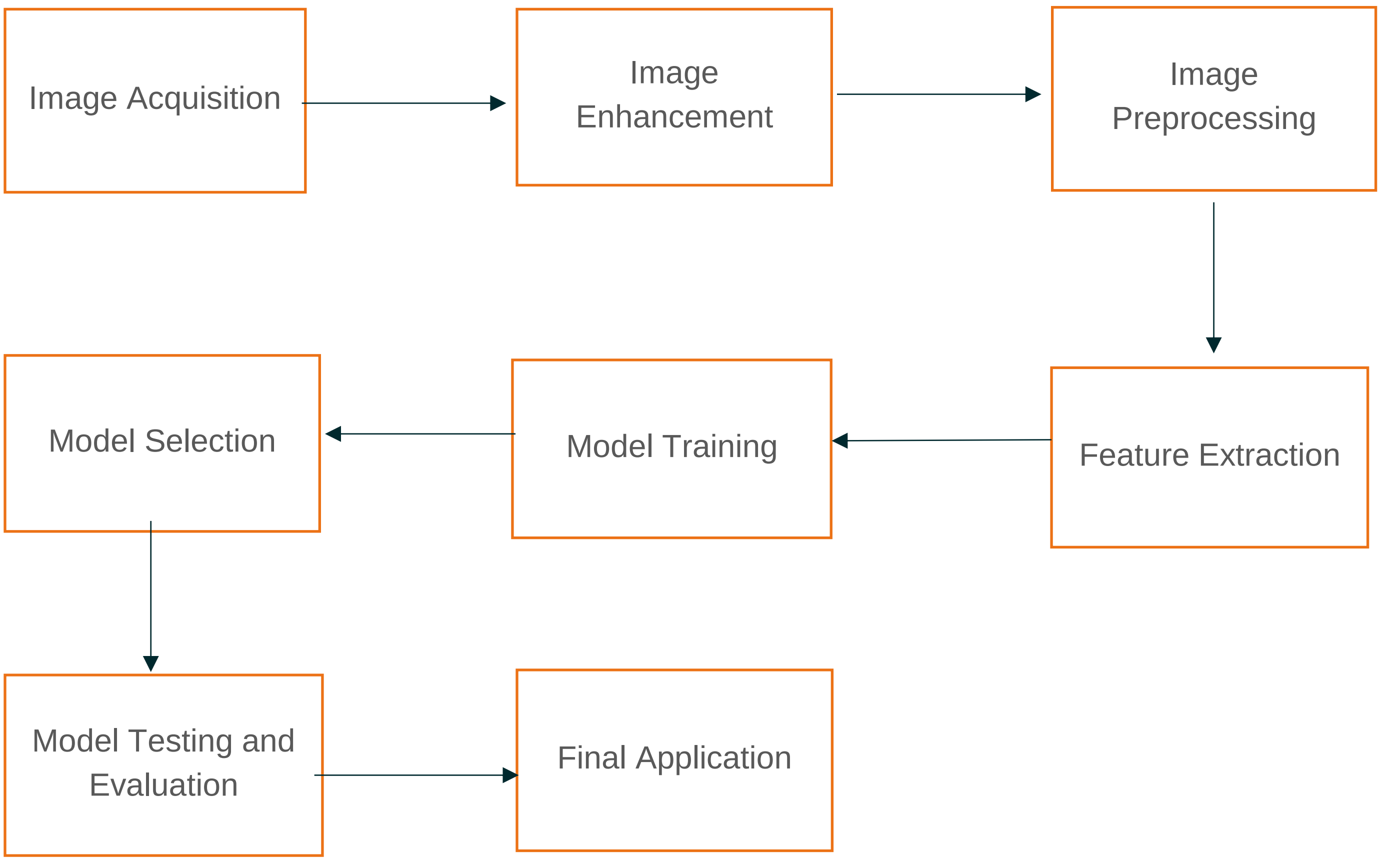
1. Each class contains **2000 images**, resulting in a total of **10,000 images in the dataset.**
2. The data is split into **three sets: training, validation, and testing**.

- Sample of dataset images:



DIAGRAM

- This diagram shows the sequence of steps I plan to follow in the project



DESCRIPTION OF EACH STEP

1. Image Acquisition

The first step in the workflow involves acquiring images from the dataset. The dataset consists of 10,000 images, categorized into five fruit types: Apples, Bananas, Grapes, Mangoes, and Strawberries. Each category contains around 2,000 images, capturing the fruits under various conditions, such as lighting, size, and orientation, to simulate real-world variability.

2. Image Enhancement

To prepare the images for further processing, several enhancement techniques are applied:

- **Resizing (cv2.resize):** Each image is resized to a fixed size of 64x128 pixels. This step standardizes the dimensions of the images, ensuring consistency across the dataset and optimizing computational efficiency. The resizing process reduces the image size while retaining essential features necessary for classification.
- **Grayscale Conversion (cv2.cvtColor):** The images are converted from their original RGB format (or BGR, in OpenCV) to grayscale. This simplifies the images, removing unnecessary color information while retaining the structural features. This reduction in complexity helps improve the efficiency of feature extraction and classification without the overhead of processing multiple color channels.
- **Histogram Equalization (cv2.equalizeHist):** After grayscale conversion, histogram equalization is applied to improve the contrast of the image. This technique enhances the visibility of features by spreading the intensity values more evenly across the image, making it more robust to lighting variations.

3. Feature Extraction

Feature extraction is a crucial step where key patterns and structures within the images are captured for classification:

- **Histogram of Oriented Gradients (HOG) (cv2.HOGDescriptor().compute):** In this step, the Histogram of Oriented Gradients (HOG) technique is applied to the resized grayscale images. HOG is particularly useful for capturing edge and texture information in images. It analyzes the gradients of intensity in localized regions of the image to detect object boundaries, which are essential for recognizing and distinguishing objects in image classification tasks.

  The key benefits of using HOG for feature extraction include its robustness to changes in lighting, scale, and slight rotations. It extracts features that are invariant to these variations, making it a strong choice for fruit recognition tasks.

4. Image Preprocessing

Image preprocessing ensures the extracted features are ready for training with machine learning models:

- **Normalization (StandardScaler):** The features extracted from the HOG technique are normalized to have a mean of 0 and a standard deviation of 1. Normalization is crucial as it ensures that all the extracted features are on a comparable scale, preventing features with larger values (e.g., pixel intensities) from dominating the model's training process. This step improves the efficiency and performance of machine learning algorithms, such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), which are sensitive to differences in feature scales.

5. Model Training

Once the images are preprocessed and features are extracted, machine learning models are trained using the processed data. Models such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forest are employed for classification. The models are trained using the preprocessed and normalized feature sets, with hyperparameters fine-tuned to optimize performance.

6. Model Selection

After training multiple models, their performance is evaluated, and the best model is selected based on accuracy, precision, recall, and F1-score. The selected model is the one that performs best at classifying the fruit types across various conditions.

7. Model Testing and Evaluation

The selected model undergoes testing and evaluation to determine its effectiveness on unseen data:

- **Confusion Matrix:** A confusion matrix is used to assess the classification performance across all fruit types. It helps in identifying misclassifications and understanding how well the model distinguishes between different fruit classes.
- **Cross-validation:** Cross-validation is employed to ensure that the model performs consistently across different subsets of the dataset. This step helps in evaluating the model’s generalization ability and prevents overfitting.
- **Performance Metrics:** The model's performance is evaluated using metrics such as precision, recall, and F1-score to assess how well the model handles class imbalances and correctly classifies fruits.