جامعة فلسطين التقنية
PTU
Kadoorie خضوري
1930
Palestine Technical University

COMPUTER VISOIN

HANAN RADDAD
ISLAM HJSALIH

Introduction

The main goal of this project is to develop a system capable of recognizing and classifying different types of fruits using computer vision techniques. This project, completed as part of the Computer Vision course, begins by applying traditional machine learning algorithms to identify fruits based on their visual features such as color, shape, and texture. After that, the project advances to the use of deep learning models, particularly convolutional neural networks (CNNs), to achieve higher accuracy and better generalization. Through this comparison, the project demonstrates the evolution from classical machine learning approaches to modern deep learning solutions in the field of image recognition.

About Dataset

https://www.kaggle.com/datasets/ryandpark/fruit-quality-classification
Dataset
Total number of images: 19526.
Quality Classes: 3 (Bad Fruit, Good Fruit, Mixed Fruit).
Fruit Classes: 6 (apple, banana, guava, lime, orange, and pomegranate)
Image sizes:

- Bad Fruit = 256x256 pixels
- Good Fruit = 256x256 pixels
- Mixed Fruit = 256x192 pixels

Images Taken in Which Direction:

- Front Direction
- Top View
- Backward Direction
- Bottom View
- Direction Rotated 180 degrees

Fruit	BAD Glass (256 x 256)	GOOD (256 x 256)	Mixed (256 x 192)	Total
Apple	1141	1149	113	2403
Banana	1087	1113	285	2485
Guava	1129	1152	148	2429
Lime	1085	1094	278	2457
Orange	1159	1216	125	2500
Pomegranate	1187	5940	125	7252
Total	**6788**	**11664**	**1074**	**19526**

Data Preparation

https://colab.research.google.com/drive/1VKwh0ZiNis45lm96wu1rE-3QkcFr7AVq#scrollTo=fWN7QyH4zlJL

In the first phase of the project, we worked on preparing and organizing the dataset. A copy of the original data was created to ensure that all preprocessing and modification steps could be applied directly without affecting the raw data. Instead of generating new files after each operation, we performed all cleaning, transformation, and filtering processes on this working copy. This approach helped maintain data consistency and made it easier to test different preprocessing techniques efficiently.

Dataset Split

In this project, the dataset of images was divided into training and test sets. The test set consists of 195 randomly selected images (1% of the total 19,527 images). This split allows us to evaluate the model's performance on unseen data while keeping the training data separate. The selected images were moved to a separate test folder, maintaining the original folder structure.

📢Removing Duplicate Images

After splitting the dataset, it is important to ensure that the training set does not contain duplicate images, as duplicates can bias the model and reduce its ability to generalize.
The process works as follows:
Hashing each image:
Each image in the training folder is opened, converted to a consistent format (RGB), resized to a standard size, and then hashed using the MD5 algorithm. This creates a unique fingerprint for each image.
Detecting duplicates:
If an image's hash already exists in the dictionary of hashes, it is considered a duplicate.
Removing duplicates:
All identified duplicate images are deleted from the training set to keep only unique images.
✅ In this dataset, 1 duplicate image was removed, ensuring a cleaner and more reliable training set for the model.

❗ Note: We removed duplicates before resizing. In hindsight, resizing first would have detected duplicates .

📢mage Resizing

After removing duplicates, we performed resizing on all images in the training set. Each image was resized to a standard size of 128x128 pixels.
The purpose of resizing is:

1. Consistency: Ensures all images have the same dimensions, which is necessary for most machine learning models.
2. Efficiency: Reduces computational cost and memory usage during training.
3. Improved processing: Helps maintain uniformity in the dataset, allowing the model to focus on the content rather than size variations.

✅ In total, all images in the training set were resized to 128x128 pixels, preparing the dataset for model training.

📢Converting Images to Grayscale

After resizing, all images in the training set were converted to grayscale. This means that each image now contains only shades of gray instead of full color.
The benefits of converting to grayscale are:

- Simplification: Reduces the complexity of the images by removing color channels.
- Faster processing: Less data to process speeds up training.
- Focus on features: Helps the model focus on shapes, textures, and patterns rather than color variations.

✅ All images were successfully converted to grayscale, preparing the dataset for further preprocessing and model training.

📢 Balancing the Dataset by Reducing Overrepresented Classes

In the "Good Quality Pomegranate" class, there were over 5,900 images, which was significantly more than other classes. To balance the dataset and prevent the model from being biased toward this class, we removed 4,800 images.
The images were deleted evenly across the dataset by removing approximately every Nth image. This approach ensures that the remaining images still represent the diversity of the original class while maintaining a balanced dataset for training.
✅ After this process, the number of images in the "Good Quality Pomegranate" class is comparable to other classes, improving model fairness and training quality.
❗ Note: Ideally, this reduction should have been done before resizing to reduce computational load and speed up processing, but in our workflow, it was applied afterward.

📣Visualizing Sample Images from Each Class

Visualizing Sample Images from Each Class
To inspect the dataset, we randomly selected up to 10 images from each class. This provides a snapshot of the dataset while keeping the visualization manageable, regardless of the total number of images per class.
❗ Note: This visualization step is referred to at every stage of preprocessing to monitor the dataset, check for errors, and ensure the quality and consistency of images throughout the workflow.

📣Multi-Stage Filter Selection Process

The process of selecting the best filter for our images is carried out in multiple stages to ensure optimal results. These stages typically include:

1. Initial Testing: Apply several candidate filters to a small sample of images to visually inspect their effects.
2. Comparison and Evaluation: Assess the filtered images based on clarity, feature enhancement, and suitability for the task.
3. Refinement: Adjust filter parameters or combine multiple filters to enhance results further.
4. Final Selection: Choose the filter (or combination) that provides the best balance of visual quality and preprocessing benefit for the dataset.

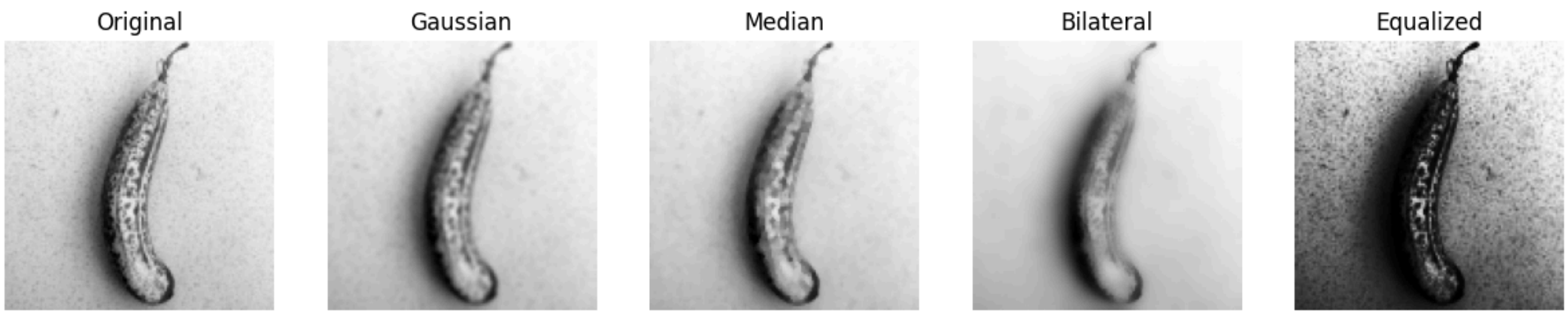
This systematic, multi-stage approach helps achieve consistent and high-quality image preprocessing before model training.

1-Filter Comparison on a Sample Image

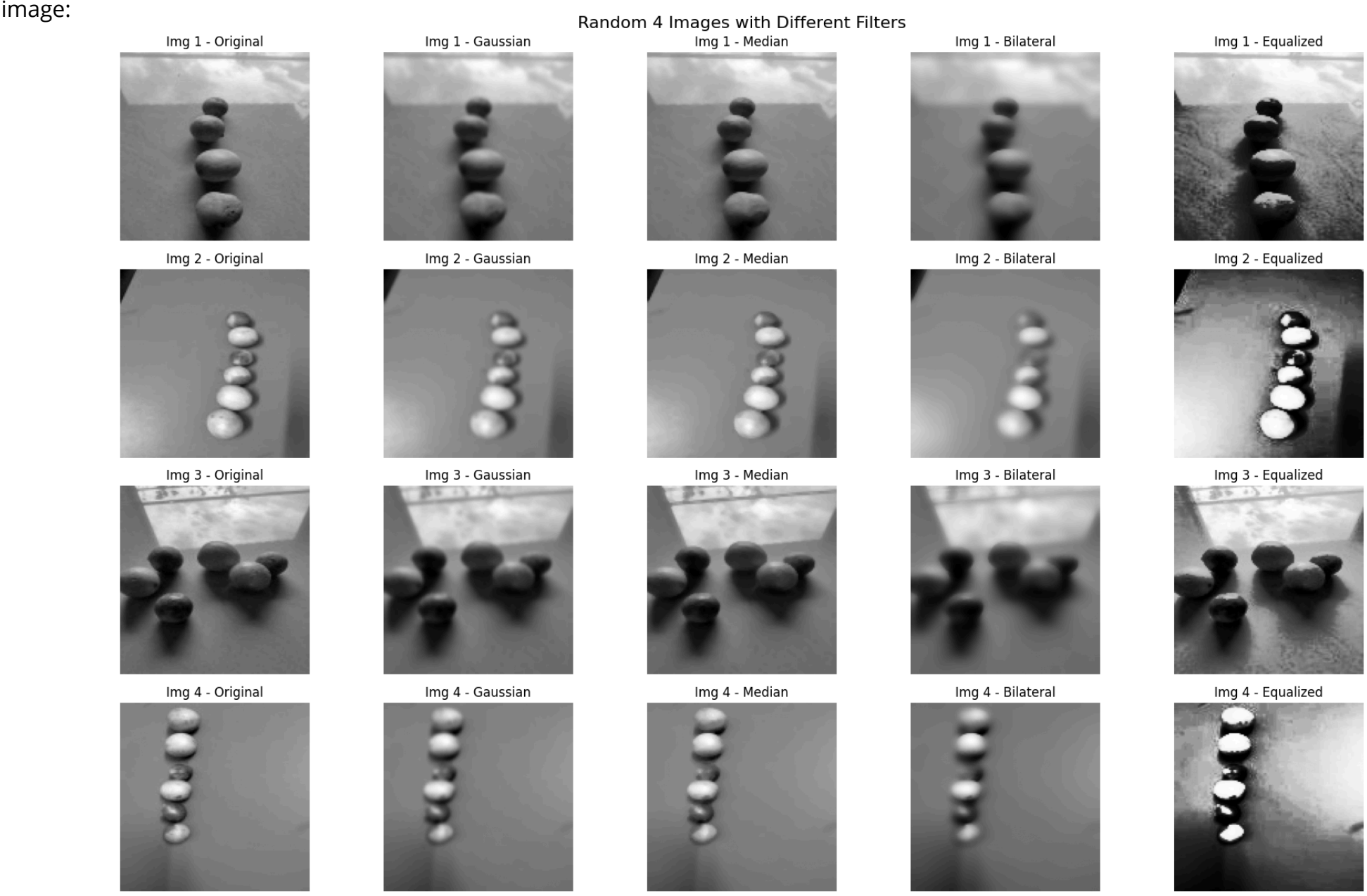
To select the most suitable filter, we applied different filters to a sample grayscale image, including:

- Gaussian Blur: Smooths the image and reduces noise.
- Median Blur: Preserves edges while reducing salt-and-pepper noise.
- Bilateral Filter: Smooths while preserving edges better than Gaussian.
- Histogram Equalization: Enhances contrast in the image.

The filtered images were displayed side by side with the original, allowing visual comparison. This step helps to assess which filter best enhances image quality and features important for model training.



To further evaluate filter performance, we selected 4 random images from a category (e.g., lemons) and applied the following filters to each image:

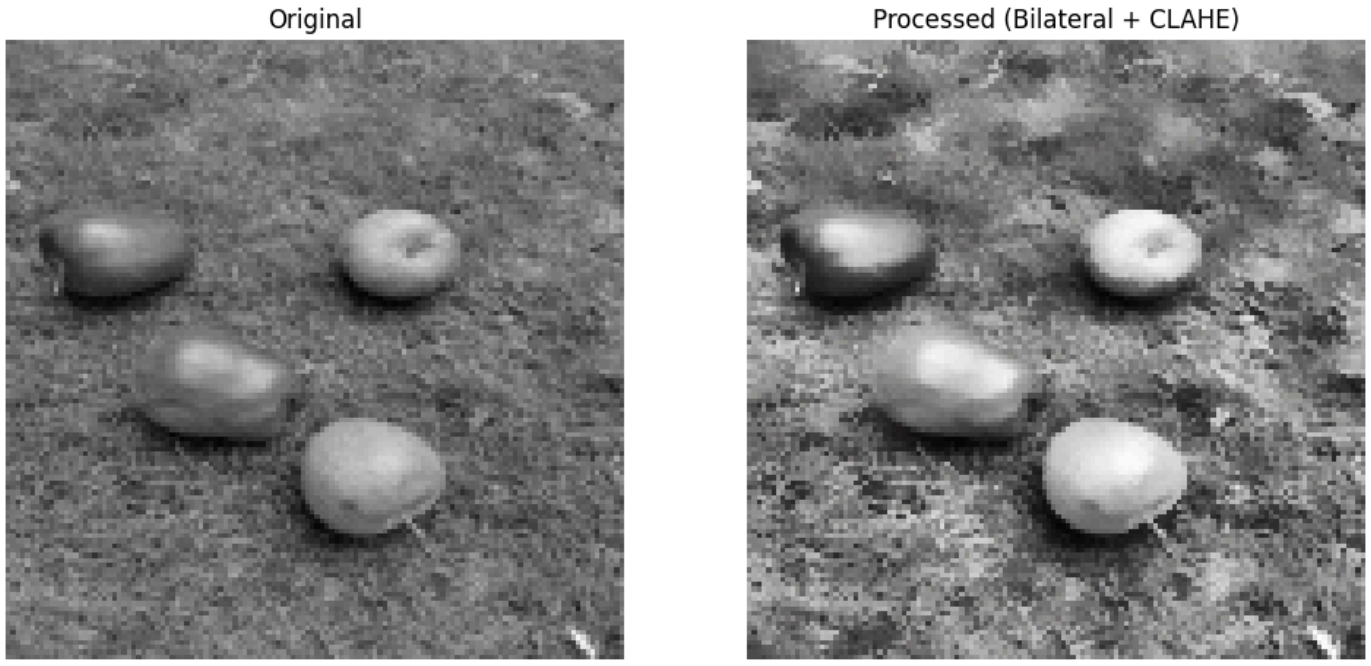


Visualizing Sample Images from Each Class

Enhancing Image Quality with Bilateral Filter and CLAHE
To improve image quality and contrast, we applied a Bilateral Filter followed by CLAHE (Contrast Limited Adaptive Histogram Equalization) on a sample image.

- Bilateral Filter: Smooths the image while preserving edges, reducing noise without blurring important features.
- CLAHE: Enhances local contrast in the image by applying histogram equalization adaptively, making features more distinguishable.

The result shows a side-by-side comparison of the original image and the processed image. This combination of filters improves visual quality and highlights important details, which is beneficial for feature extraction and model training
❗ Note: This combination of filters (Bilateral + CLAHE) was tested based on recommendations from AI assistants after providing them with a set of sample images..



Applying Bilateral Filter and CLAHE to the Entire Dataset

Applying Bilateral Filter and CLAHE to the Entire Dataset
After testing the Bilateral Filter + CLAHE on sample images, we applied it to all images in the training dataset. Before processing, we adjusted the filter parameters (e.g., d=9, sigmaColor=25, sigmaSpace=25 for Bilateral Filter, and clipLimit=3.0, tileGridSize=(6,6) for CLAHE) based on the sample results to ensure optimal enhancement.
✅ All images were successfully processed, improving contrast and preserving edges consistently across the dataset.