Image Similarity Analysis Based on Color Histograms and Dominant Colors

AACHABI Mohammed
Computer Science Department
Faculty of Sciences and Technology, Tangier, Morocco

Abstract

Image similarity analysis is a key domain in computer vision, used in applications such as content-based image retrieval, image library management, and visual recommendation systems. This project explores and compares images based on color characteristics using two main approaches: color histograms and dominant color extraction. The similarity is computed using histogram-based Bhattacharyya distance, dominant color distance, and a combined global metric.

Contents

Chapter 1

Introduction

The primary goal of this project is to analyze and compare a set of images based on their color characteristics to determine their similarity. The color features used for this analysis include:

- Color histograms for each color channel (Red, Green, Blue) to quantify color distribution.
- Dominant colors extracted using KMeans clustering, focusing on significant colors only.

Chapter 2

Project Structure

The project is organized as follows:

- **DVI1**: Folder containing 16 reference images for similarity comparison, categorized by types such as animals, flowers, ocean, and sky.
- **DVI2**: Folder with 4 test images, one per category, used for similarity testing.

2.1 Objectives

The specific objectives of this project include:

- Extracting color characteristics (histograms and dominant colors) for each image.
- Comparing images based on Bhattacharyya distance for histograms, Euclidean distance for dominant colors, and a global metric combining both.
- Visualizing and ranking the most similar image pairs for each metric.

Chapter 3

Methodology

3.1 Step 1: Loading and Preprocessing Images

Images from the specified directory are loaded and harmonized in format for analysis.

Figure 3.1: Sample image from DVI1 folder

3.2 Step 2: Color Histogram Calculation

Histograms are calculated for each color channel (Red, Green, Blue) using OpenCV, then normalized to enable consistent comparisons across images.

Listing 3.1: Python Code for Calculating Color Histograms

```
def calculate_color_histogram(image_path):
    img = cv2.imread(image_path)
    chans = cv2.split(img)
    histograms = {color: cv2.normalize(cv2.calcHist([chan], [0],
        None, [256], [0, 256]), hist).flatten() for chan, color in
         zip(chans, ('b', 'g', 'r'))}
    return histograms
```

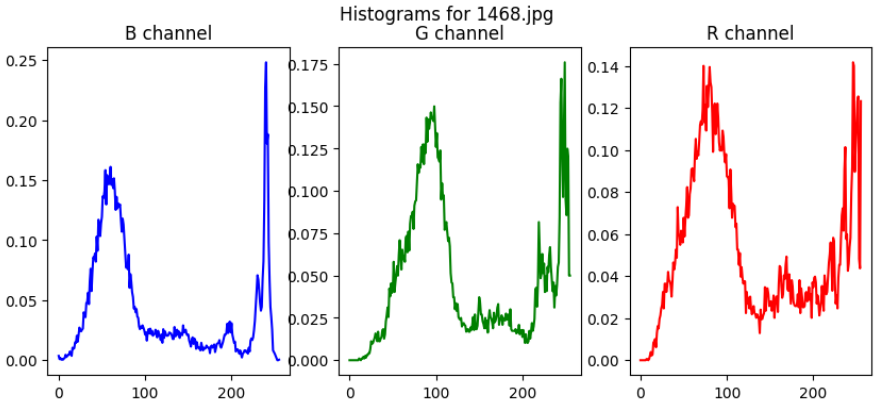

Figure 3.2: Color Histogram for an image

3.3 Step 3: Dominant Color Extraction

- KMeans clustering is applied to segment each image into color clusters.
- A threshold of 5% is used to retain only significant colors, filtering out noise and preserving the most representative colors.

Listing 3.2: Python Code for Finding Dominant Colors

```
def find_dominant_colors(image_path, k=20, threshold=0.05):
    img = cv2.imread(image_path)
    img = img.reshape((-1, 3))
    kmeans = KMeans(n_clusters=k, random_state=0).fit(img)
    dominant_colors = [kmeans.cluster_centers_[idx].tolist() for
       idx, count in Counter(kmeans.labels_).items() if count /
       total_pixels > threshold]
    return dominant_colors
```

Figure 3.3: dominants color

3.4 Step 4: Similarity Distance Calculations

Two types of distances are used to quantify similarity:

- **Bhattacharyya Distance** between color histograms for each channel, indicating similarity between color distributions.
- **Euclidean Distance** between dominant colors to measure similarity in main color tones.

- **Global Metric** combining normalized distances from histograms and dominant colors.

Listing 3.3: Python Code for Calculating Bhattacharyya Distance

```
def bhattacharyya_distance(hist1, hist2):
    return cv2.compareHist(hist1, hist2, cv2.HISTCMP_BHATTACHARYYA
        )
```

Chapter 4

Results

The results are displayed in three categories: histogram similarity, dominant color similarity, and global metric similarity.

Figure 4.1: Comparison of Similar Images Based on Global Metric

Chapter 5

Conclusion

This project demonstrates the effectiveness of color-based features for image similarity analysis. While histograms provide a broad sense of color distribution, dominant colors capture the main tones, and combining both provides a balanced similarity metric.

Chapter 6

Future Work

Further improvements could involve:

- Testing with different clustering methods for color extraction.
- Exploring other similarity metrics and multi-channel histograms.
- Extending the analysis to texture and shape features.