CS 5330 PATTERN RECOGNITION AND COMPUTER VISION
PROJECT-2

Anuj Patel 002874710
Ruchik Jani 002825482

1) Project description:

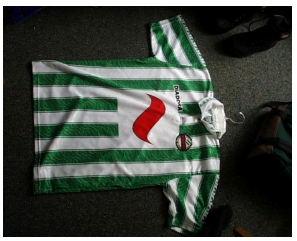
This project focuses on content-based image retrieval (CBIR), aiming to find database images similar to a target image. It utilizes classic features like color, texture, and spatial layout, along with deep network embeddings from a pre-trained ResNet18 model. The process involves computing features for both target and database images, comparing them using chosen distance metrics, and ranking database images by similarity. Tasks include baseline matching with a 7x7 central square and sum-of-squared-difference, histogram matching with normalized color histograms and histogram intersection, multi-histogram matching using spatially diverse color histograms, and texture and color matching with combined histograms and chosen texture metrics. Deep network embeddings are also employed, comparing feature vectors with chosen distance metrics. Custom designs allow experimentation with tailored feature vectors and distance metrics. The project aims to provide a comprehensive understanding of CBIR techniques' effectiveness across various scenarios.

2) Image descriptions for project tasks:

Task 1 - Baseline Matching:


pic.1016.jpg


pic.0986.jpg


pic.0641.jpg


pic.0547.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
1
Running Baseline Features Comparison...
Top 4 matches for the target image /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.1016.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.1016.jpg (Distance: 0)
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0986.jpg (Distance: 118.528)
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0641.jpg (Distance: 147.499)
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0547.jpg (Distance: 222.942)
```

Output Screenshot with top matches and distances

Feature Type: 7x7 middle pixel
Distance Metric: Sum of Squared Differences

The leftmost image is the target image, and the other 3 images match the most with the target, i.e., these are most similar to the target image. The order of similarity is from left to right, with the 2nd from left being the most similar and the rightmost being the least similar to the target image.

When we run our code, a menu is displayed on the terminal which prompts the user to select which matching method is to be executed. When the user types ‘1’ which corresponds to Task-1 in the menu, the corresponding distance metric and feature type are computed and the top 3 matches are displayed along with the distances.

Task 2 Histogram Matching:


pic.0164.jpg


pic.0110.jpg

pic.0976.jpg


pic.0898.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
2
Running RGB Histogram comparison...
Match 1: /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0164.jpg
Match 2: /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0110.jpg
Match 3: /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0976.jpg
Match 4: /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0898.jpg
```

Output Screenshot with top matches

Feature Type: RGB Histogram
Distance Metric: Sum of Squared Differences

The leftmost image is the target image, and the other 3 images match the most with the target, i.e., they are most similar to the target image. The order of similarity is from left to right, with the 2nd from left being the most similar and the rightmost being the least similar to the target image.

When we run our code, a menu is displayed on the terminal which prompts the user to select which matching method is to be executed. When the user types ‘2’ which corresponds to Task-2 in the menu, the corresponding distance metric and feature type are computed and the top 3 matches are displayed.

The images *pic.0976.jpg* and *pic.0898.jpg* are detected because the *pic.0976.jpg* contains a sign with blue background and the image *pic.0898.jpg* has a surface that reflects the blue sky.

Task 3 - Multi-histogram Matching:


pic.0274.jpg


pic.1055.jpg


pic.0409.jpg


pic.0482.jpg

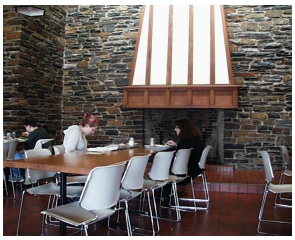
```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
3
Running Task 3: Multi-histogram Matching...
Top three similar images based on combined whole and center histograms:
Image: /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0274.jpg
Image: /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.1055.jpg
Image: /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0409.jpg
Image: /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0482.jpg
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$
```

Output Screenshot with top matches

Feature Type: Whole-image and Centre-image histogram
Distance Metric: Histogram intersection with weighted average

The image order and code working is similar to Task-1 & 2. Now the user has to enter 3 to run this task. In this task, we computed the central region histogram using a region defined by a rectangle that starts at 1/4th of the image's width and height and extends to half the width and height, thus capturing the central half of the image. Both the histograms are 3D color histograms with 8 bins per channel. The distance metric used is histogram intersection, computed by the **cv::compareHist** function with the **cv::HISTCMP_INTERSECT** method. The distances (intersection values) for the whole-image histogram and the central region histogram are combined using a simple average to get a single distance measure for each image compared to the target. This approach gives equal weight to the global color distribution and the specific color distribution in the central region, assuming both are equally important for determining image similarity. The *pic 0482.jpg* was detected because of the central image histogram feature. The central half of the image has a blue colored background similar to the target image.

Task 4 - Texture and Color:


pic.0535.jpg


pic.0206.jpg


pic.1046.jpg


pic.0842.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
4
Running Task 4: Texture and Color...
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0206.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.1046.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0842.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0798.jpg
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$
```

Output Screenshot with top matches

Feature Type: Whole-image and Texture histogram
Distance Metric: Histogram intersection with Bhattacharya distance

The image order and code working is similar to the previous tasks. Now the user has to enter 4 to run this task. The Bhattacharyya Distance is a measure used to quantify the similarity or divergence between two histograms. In code, the **computeColorHistogram** function analyzes color distribution in the HSV color space, while **computeTextureFeatures** extracts texture details using gradient magnitudes and orientations via the Sobel operator. Similarity between histograms is quantified with the compareHistograms function using the Bhattacharyya distance. Finally, **findSimilarImages** orchestrates the comparison, merging color and texture similarities to identify the top N matching images in a specified directory. This approach effectively balances color and texture information, showcasing the utility of these functions in comprehensive image similarity evaluation. Task-4 considers both color and texture information, offering a more comprehensive representation of the image content compared to Task-2 & Task-3. As a result, the top matches in this approach better captures the overall similarity in color and texture with the target image.

Task 5 - Deep Network Embeddings:

Target images which are similar to pic.0893.jpg


pic.0893.jpg


pic.0897.jpg


pic.0136.jpg

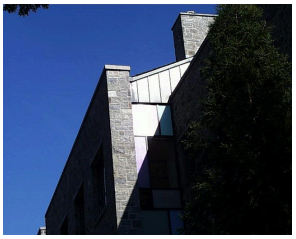

pic.0146.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
5
Running DNN Embeddings comparison...
Similar images to /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0893.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0897.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0136.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0146.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0135.jpg
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$
```

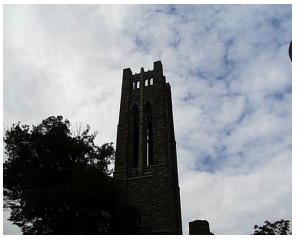
Output Screenshot with top matches

Target images which are similar to pic.0164.jpg


pic.0164.jpg


pic.1032.jpg


pic.0213.jpg


pic.0690.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
5
Running DNN Embeddings comparison...
Similar images to /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0164.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.1032.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0213.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0690.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0426.jpg
```

Output Screenshot with top matches

Feature Type: Deep Neutral Network Embeddings
Distance Metric: Cosine Distance

The image order and code working is similar to the previous tasks. Now the user has to enter 5 to run this task. In this task, the **loadFeatures** function reads a CSV file to load 512-dimensional DNN embeddings (ResNet18 outputs) for each image into a map, linking filenames to their feature vectors. The **split** function assists in parsing the CSV by separating filenames from their corresponding feature values. The **cosineDistance** function then calculates the similarity between feature vectors of the target and other images using cosine distance, a method effective in high-dimensional spaces for capturing orientation similarity. Finally, **findSimilarImagesDNN** utilizes these components to identify and rank the top N images most similar to a given target image based on their cosine distances.

Compared to Tasks 1 - 4, this task produces matches that align well in terms of semantic content, offering a more abstract and context-aware similarity assessment. The top matches include images with similar semantic content or visual concepts, even if they differ in color, texture, or structure.

Task 6 - Compare DNN Embeddings and Classic Features:

pic.1072.jpg (DNN Embeddings)




pic.1072.jpg *pic.0143.jpg* *pic.0863.jpg* *pic.0329.jpg*

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
5
Running DNN Embeddings comparison...
Similar images to /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.1072.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0143.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0863.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0329.jpg
```

Output Screenshot for pic.1072.jpg with top matches using DNN embeddings

pic.1072.jpg (Classic Features)




pic.1072.jpg *pic.0709.jpg* *pic.0552.jpg* *pic.0036.jpg*

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
6
Running Tak 6 : classic features comparison...
Top similar images to /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.1072.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.1072.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0709.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0552.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0036.jpg
```

Output Screenshot for pic.1072.jpg with top matches using classic features

pic.0948.jpg (DNN Embeddings):


pic.0948.jpg


pic.0930.jpg


pic.0960.jpg


pic.0928.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
5
Running DNN Embeddings comparison...
Similar images to /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0948.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0930.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0960.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0928.jpg
```

Output Screenshot for pic.0948.jpg with top matches using DNN embeddings

pic.0948.jpg (Classic Features):


pic.0948.jpg


pic.0948.jpg


pic.0948.jpg

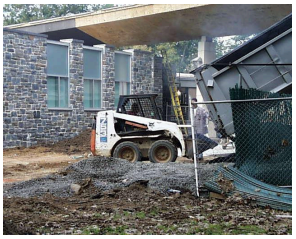

pic.0948.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
6
Running Tak 6 : classic features comparison...
Top similar images to /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0948.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0948.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0111.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0659.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0440.jpg
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$
```

Output Screenshot for pic.0948.jpg with top matches using classic features

pic.0734.jpg(DNN Embeddings):


pic.0734.jpg


pic.0731.jpg


pic.0735.jpg


pic.0739.jpg

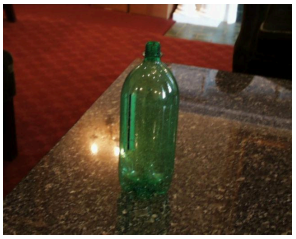
```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
5
Running DNN Embeddings comparison...
Similar images to /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0734.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0731.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0735.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0739.jpg
```

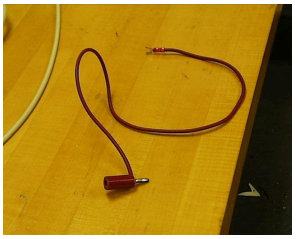
Output Screenshot for pic.0734.jpg with top matches using DNN embeddings

pic.0734.jpg(Classic Features):


pic.0734.jpg

pic.0036.jpg


pic.0913.jpg


pic.0048.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
6
Running Tak 6 : classic features comparison...
Top similar images to /home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0734.jpg:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0734.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0036.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0913.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0048.jpg
```

Output Screenshot for pic.0734.jpg with top matches using classic features

Feature Type: Color Histogram and Local Binary Patterns
Distance Metric: L2 Norm (Euclidean Distance) between 2 histograms

The image order and code working are similar to the previous tasks. Now the user has to enter 6 to run this task.

In this task, the **computeHistogram** function calculates color histograms in the HSV color space for each image, capturing color distribution, while **computeLBP** generates Local Binary Patterns to describe texture details. The **featureDistance** function then measures Euclidean distances between corresponding histograms of a target image and other images in a dataset, assessing both color and texture similarity. Finally, **findSimilarImagesClassic** aggregates these distances to identify and rank the top N most similar images to the target, combining color and texture features for a comprehensive similarity assessment. This approach allows for a nuanced comparison of images based on classic visual features, offering an alternative to deep learning-based methods.

The discrepancy in similarity results between Task-5 and Task-6 largely stems from the intrinsic differences in how these methods process and interpret image content. DNN embeddings, derived from a deep neural network trained on a diverse dataset like ImageNet, capture complex, high-level abstractions of images, making them highly effective at identifying perceptual similarities across varying conditions. The use of cosine distance further enhances this by focusing on the orientation of high-dimensional feature vectors, which correlates well with visual similarity. In contrast, classic features such as color histograms and Local Binary Patterns primarily capture basic color and texture information, which, while useful, may not encapsulate the broader context or nuanced differences between images as effectively. Consequently, the Euclidean distance metric used with these lower-dimensional features might not capture the perceptual similarities as accurately as the cosine distance in high-dimensional spaces, leading to less satisfactory results when comparing similar images using deep embeddings.

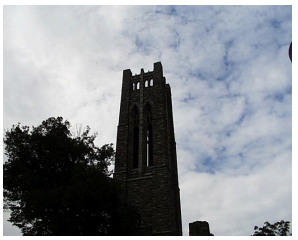
Task 7 - Custom Design:

pic.0144.jpg(Similar Images)

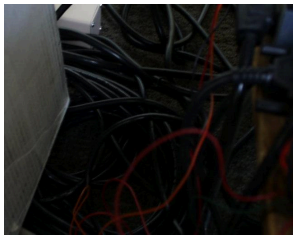
pic.0144.jpg

pic.0949.jpg

pic.0981.jpg

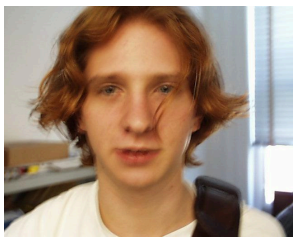


pic.0690.jpg



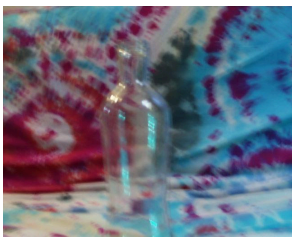
pic.0079.jpg

pic.0144.jpg(Least Similar Images)



pic.0088.jpg

pic.0880.jpg



pic.0102.jpg

pic.0549.jpg

pic.0915.jpg

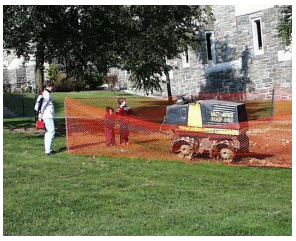
```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
7
Running Task 7: Combined Features...
Top similar images based on combined features:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0144.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0949.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0981.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0690.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0079.jpg

Least similar images:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0088.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0880.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0102.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0549.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0915.jpg
```

Output Screenshot for pic.0144.jpg with most similar and least similar matches

pic.0368.jpg(Similar Images)

pic.0368.jpg

pic.0142.jpg



pic.0875.jpg

pic.0711.jpg

pic.0429.jpg

pic.0368.jpg(Least Similar Images)

pic.0030.jpg

pic.0953.jpg

pic.0623.jpg

pic.0364.jpg

pic.0475.jpg

```
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ ./match
Enter:
1 Baseline Matching
2 Histogram Matching
3 Multi-histogram Matching
4 Texture and Color
5 Deep Network Embeddings
6 Classic Features
7 Custom Design
7
Running Task 7: Combined Features...
Top similar images based on combined features:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0368.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0142.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0875.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0711.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0429.jpg

Least similar images:
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0030.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0953.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0623.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0364.jpg
/home/newusername/Downloads/Project-2222/Project-2/data/olympus/pic.0475.jpg
newusername@anuj:~/Downloads/Project-2222/Project-2/bin$ q
```

Output Screenshot for pic.0368.jpg with most similar and least similar matches

Feature Type: Color histogram, local binary patterns, and Hu Moments
Distance Metric: Euclidean distance and weighted distance between two image features

The image order and code working are similar to the previous tasks. Now the user has to enter 7 to run this task. By evaluating the Content-Based Image Retrieval (CBIR) system that combines color histograms, Local Binary Patterns (LBP) for texture, and Hu Moments for shape descriptors, we select the grass and flower image as a target image.

The **calculateHistogram, calculateLBP,** and **calculateHuMoments** are used to compute different types of features for an image. These computed features are then combined in the **calculateWeightedDistance** function to calculate a single weighted distance metric representing the similarity between two images. Finally, **findSimilarImages** utilizes the computed features and the weighted distance metric to find and rank similar images to a target image.

The code for this task utilizes multiple features to capture different aspects of image content (color, texture, shape). Weighted distance calculation allows adjusting the importance of each feature, providing flexibility in similarity computation. However, the computational complexity may increase with larger datasets due to feature computation and distance calculation for all images.

3) Short reflection on learnings:

During this project, we learned a lot about how computers capture and process images. It was fascinating to see how a computer can find images that are similar by examining their colors, and shapes, and even using advanced methods that mimic how human brains work (deep network embeddings). We discovered that the way we choose to look at the pictures really matters. Some methods are better for certain types of images than others. For example, simple color checks might work for finding pictures with the same dominant colors, but if we are looking for specific patterns or details, we might need to use more complex techniques. We also learned that organizing and preparing the pictures before searching can save a lot of time. This project was a great way to see how technology can mimic human abilities to recognize and categorize images, and it made us aware of the challenges and possibilities in the field of image recognition.

4) Acknowledgement:

We have referred to the following sources to complete this project:

1. https://opencv.org
2. Computer Vision by Linda Shapiro and George Stockman
3. www.google.com
4. https://www.youtube.com/watch?v=gpu9p3d53fg&list=PLkmvobsnE0GHMmTF7GTzJnCISue1L9fJn&index=10