Project -2 Report - Content-Based Image Retrieval Pattern Recognition and Computer Vision (CS-5330)

Team Members:
Poojit Maddineni (NUID: 002856550)
Venkata Satya Naga Sai Karthik Koduru (NUID: 002842207)

1. **Project Description:**
   In this project, we delved into various aspects of image manipulation, analysis, and pattern recognition, offering a comprehensive learning experience in computer vision. Here's a summary of the key skills and concepts we learned:

   - **Image Processing Fundamentals:** We learned how to manipulate and analyze images at the pixel level, gaining insights into color spaces, histograms, spatial features, texture features, and embeddings.
   - **Feature Extraction and Representation:** We explored different methods for extracting features from images, including classic features like color, texture, and spatial layout, as well as deep network embeddings from pre-trained models like ResNet18.
   - **Distance Metrics and Similarity Measures:** We investigated various distance metrics for comparing image features, such as sum-of-squared-difference, histogram intersection, and cosine distance, understanding how they impact similarity assessments.
   - **Pattern Recognition Techniques:** By implementing matching algorithms and evaluating image similarities, we gained practical experience in pattern recognition and content-based image retrieval.
   - **Integration of Classic and Deep Learning Techniques:** This project integrated both classic computer vision techniques and deep learning approaches, enabling us to compare and contrast their performance in image analysis tasks.

   Overall, this project provided us with a hands-on opportunity to apply theoretical knowledge to real-world image datasets, fostering a deeper understanding of image processing techniques and their practical implications in computer vision applications.

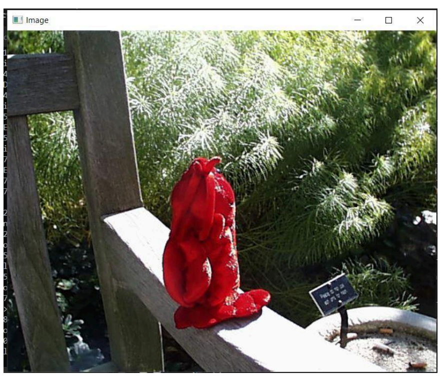
Tasks:

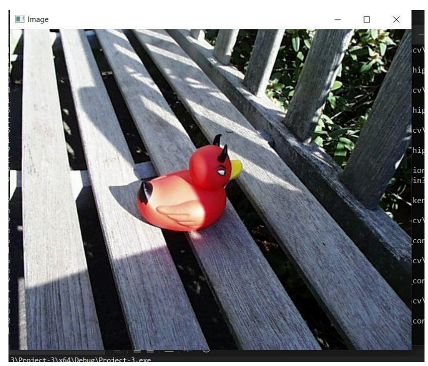
Task - 1: Baseline Matching:

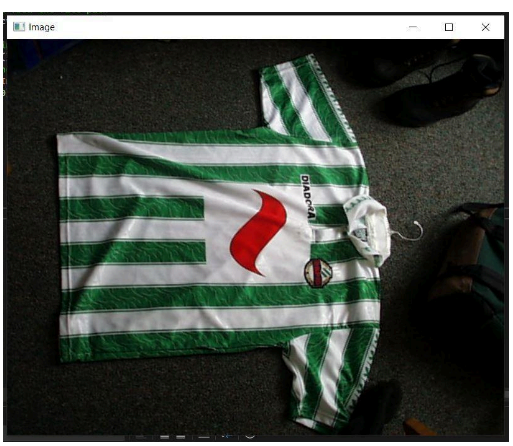
The task requires the use of the 7x7 square in the middle of the images as a feature vector and use the target image" T" and find the top 3 closest matches to the target image with similar features" F" as the target image from a data set of images" B". The distance metric used to compute the distances is "sum-of-squared-difference".

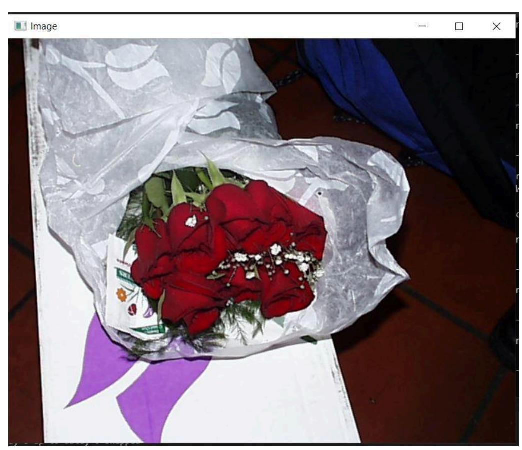
Results:

The results we got are the same as those given in the project handout. The retrieval is so accurate because all images have a red color region at the center of the images where the 7x7 square region is considered.









(From left to right: Target image: pic.1016.jpg, Top 3 matches: pic.0986.jpg, pic.0641.jpg, pic.547.jpg)

```
Sorted vector:
Image Path: C:\Users\pooji\Downloads\Test_data\pic.1016.jpg, Distance: 0
Image Path: C:\Users\pooji\Downloads\Test_data\pic.0986.jpg, Distance: 14049
Image Path: C:\Users\pooji\Downloads\Test_data\pic.0641.jpg, Distance: 21756
Image Path: C:\Users\pooji\Downloads\Test_data\pic.0547.jpg, Distance: 49703
Image Path: C:\Users\pooji\Downloads\Test_data\pic.1013.jpg, Distance: 51539
Image Path: C:\Users\pooji\Downloads\Test_data\pic.0233.jpg, Distance: 55806
Image Path: C:\Users\pooji\Downloads\Test_data\pic.1012.jpg, Distance: 74286
```

The Top 3 images list.

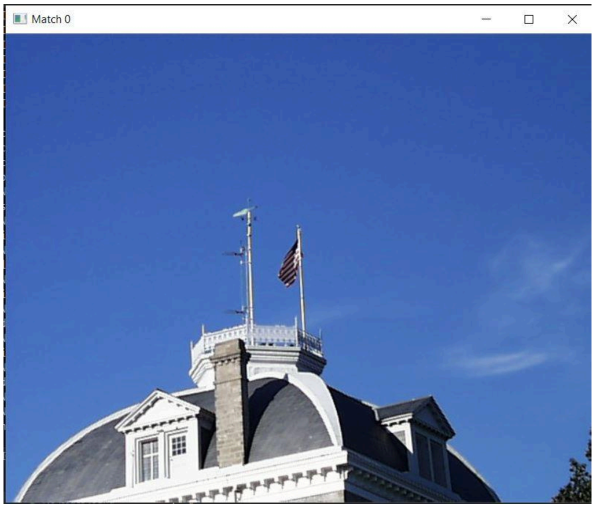
Task - 2: Histogram Matching:

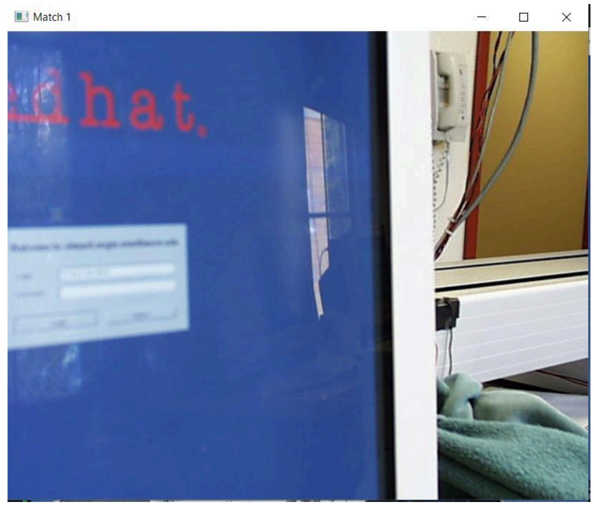
In the task, a 2D histogram is computed using only the Hue (H) and Saturation (S) values from the HSV color space. The distance metric considered to calculate the distances between the feature vectors F target image T and the other image in the data set B is the Histogram intersection.

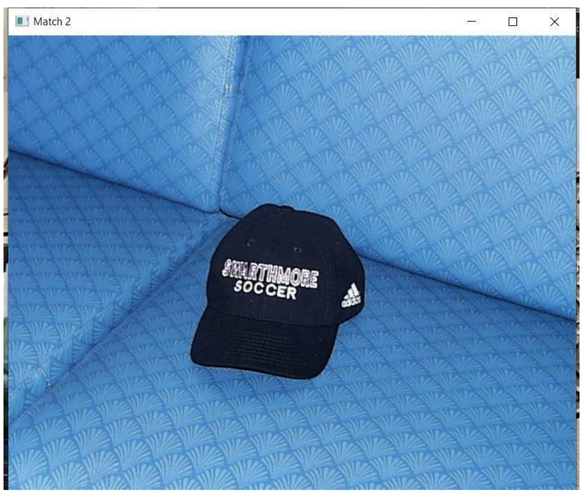
This choice captures color information while ignoring brightness variations (Value, V). The resulting histogram represents the distribution of colors in the image based on hue and saturation, providing a compact representation of its color content. This approach allows for efficient comparison of images based on their color distributions while disregarding variations in brightness.

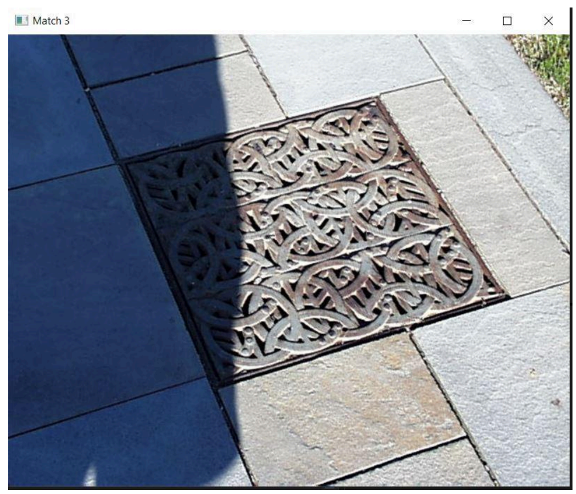
Results:

The top 3 results seem closer to that of the target image as Hue represents the dominant wavelength of light, so in an image with more blue tones, the hue values associated with those blue colors will be more prominent. Therefore, in an image with more blue, the hue values associated with blue hues will be more prevalent, potentially leading to a higher concentration of hue values within the blue range in the histogram representation. Also, due to different lighting conditions, the saturation values of the dominant color, in this case, vary in different images. So, when Histogram matching is done, the images with a more prominent dominant color are matched closely.









(From left to right: Target image: pic.0164.jpg, Top 3 matches: pic.0080.jpg, pic.0599.jpg, pic.166.jpg)

```
Top three matches for the target image pic.0164:
Match1, Filename: C:\Users\pooji\Downloads\Test_data\pic.0164.jpgDistance : 0.996094
Match2, Filename: C:\Users\pooji\Downloads\Test_data\pic.0080.jpgDistance : 0.997496
Match3, Filename: C:\Users\pooji\Downloads\Test_data\pic.0599.jpgDistance : 0.997609
Match4, Filename: C:\Users\pooji\Downloads\Test_data\pic.0166.jpgDistance : 0.998031
```

Top Image matches list.

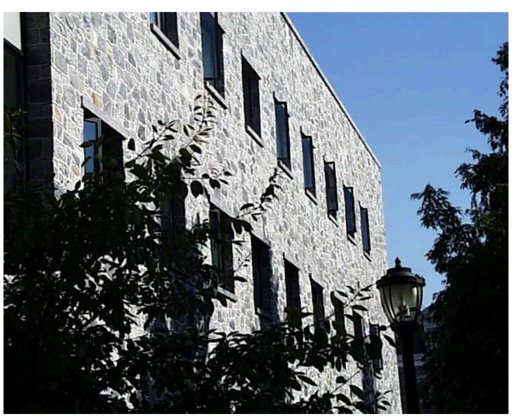
Task 3: Multi-histogram Matching:

In this task, we have computed two 2D histograms for the top and bottom halves of a given image. These histograms represent color distributions, calculated using the RGB color space. By utilizing the histogram intersection distance metric, the code measures the similarity between the target image's histograms and those of images in a dataset. The process includes loading a target image, dividing it into halves, and calculating histograms for each half. Then, histograms are computed for the dataset images' corresponding halves.

Results:

The results demonstrate a high level of accuracy in identifying similar images within the dataset based on color distribution similarities in specific image regions. Although the order of the top matching images differed slightly from the project handout, the matches were nonetheless perfect. The process of computing histograms for the top and bottom halves of the target image and subsequently calculating the intersection distance yielded precise results. This approach not only effectively identifies similar images within the dataset but also showcases the robustness of using histogram intersection distance as a metric for image similarity assessment. Overall, the method employed in this task proves to be reliable and efficient in identifying relevant images based on their color distributions.



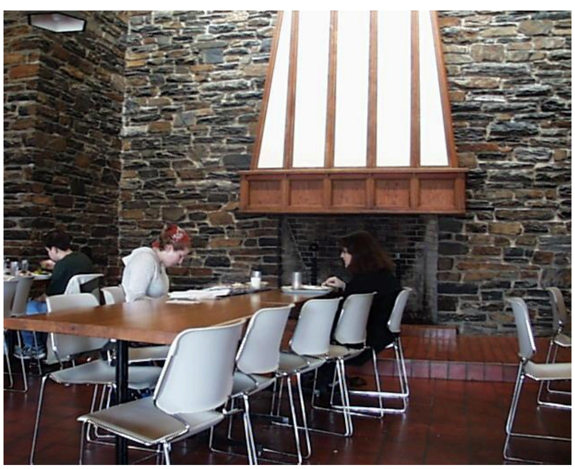



(From left to right: Target image: pic.0274.jpg, Top 3 matches: pic.0409.jpg, pic.1031.jpg, pic.0273.jpg)

Task 4: Texture and Color:

In this task, we utilized two distinct histograms: a color histogram and a texture histogram. The color histogram, calculated in the HSV color space, provides a representation of the distribution of colors within the image. On the other hand, the texture histogram was generated by applying the Sobel operator to detect gradients in the grayscale version of the image, subsequently computing the histogram of gradient magnitudes. To evaluate the similarity between the target image and the dataset images, we employed the Euclidean distance metric, considering both color and texture histograms. By combining the distances obtained from these two histograms using a weighted average approach with equal weights (0.5 and 0.5), we effectively measured the overall similarity between images. This comprehensive approach ensures a robust assessment of image similarity, facilitating accurate content-based image retrieval.

Results:
The texture analysis component, prominently showcased in the matching images, effectively identified features like brick walls, demonstrating the robustness of the texture histogram computation using the Sobel operator. Furthermore, the color analysis revealed a predominant presence of whites and browns in the matching images, aligning with the target image's color distribution. Despite slight variations in the order of matching images compared to the project handout, the retrieved images exhibited significant similarities, affirming the accuracy of the approach. These results underscore the utility of incorporating both color and texture analysis, as demonstrated by the code, to achieve precise and relevant image retrieval based on content characteristics.



(From left to right: Target image: pic.0535.jpg, Top 3 matches: pic.1033.jpg, pic.140.jpg, pic.1049.jpg)

Task 5: Deep Network Embeddings

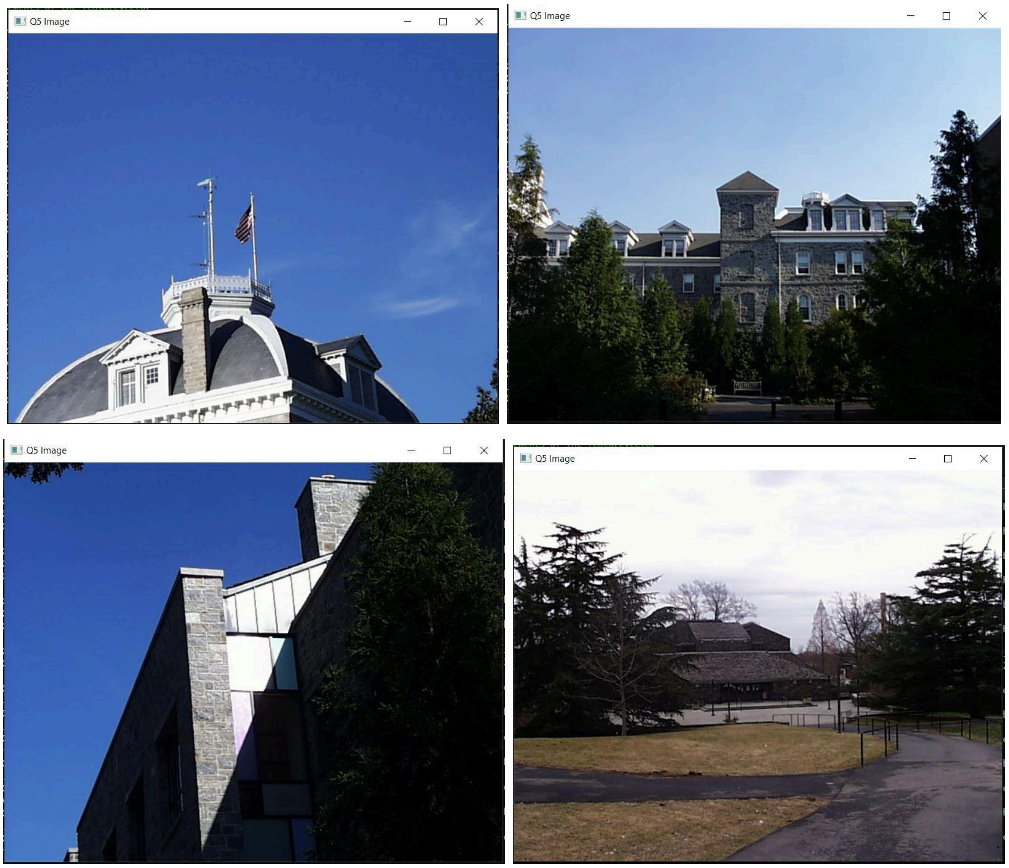
This task entails comparing image feature vectors from a CSV file, each row containing a filename and 512 values representing ResNet18 output on ImageNet.Sum square distance is used as the distance metric. This comparison facilitates tasks like image similarity search and content-based retrieval.

Results:

When the target image *pic.0893.jpg* is passed the program returns the top 3 matches that are very similar, although the color of the images is different it was able to return the right images because ResNet18, is pre-trained on the ImageNet dataset. Its training primarily focuses on learning features representing various visual patterns and concepts in natural images. These features include edges, textures, shapes, object parts, and high-level semantic representations of objects and scenes. So, it was able to retrieve similar images. Similarly, it was able to match the texture in the second image case and retrieved similar images with stone walls with a background blue sky.

(From left to right: Target image: *pic.0893.jpg*, Top 3 matches: pic.0897.jpg, pic.0136.jpg,pic.0885.jpg)

```
Distance: 0 for image pic.0893.jpg
Distance: 168.773 for image pic.0897.jpg
Distance: 208.361 for image pic.0136.jpg
Distance: 250.273 for image pic.0885.jpg
Distance: 255.215 for image pic.0135.jpg
```



(From left to right: Target image: *pic.0164.jpg*, Top 3 matches:pic.0213.jpg, pic.1032.jpg, pic.0543.jpg)

```
Distance: 0 for image pic.0164.jpg
Distance: 149.049 for image pic.0213.jpg
Distance: 172.578 for image pic.1032.jpg
Distance: 179.482 for image pic.0543.jpg
Distance: 199.441 for image pic.0690.jpg
Distance: 202.683 for image pic.0624.jpg
```

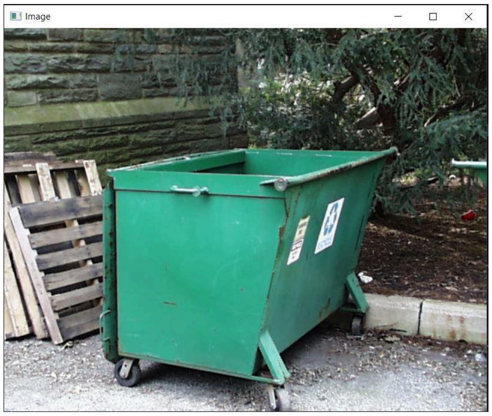
Task 6: Compare DNN Embeddings and Classic Features

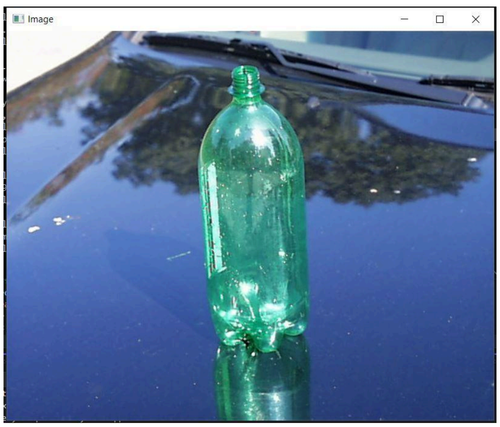
In all the comparisons the Target Image is the image in the top left position. The comparison of outputs between DNN and classical features is depicted below

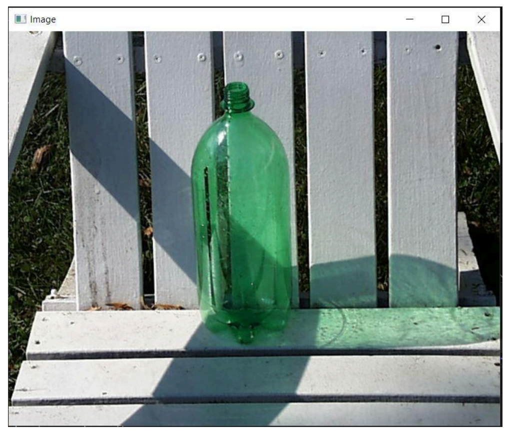
Case - 1: Target Image: pic.0752

Task-5 Deep Network Embeddings Output:

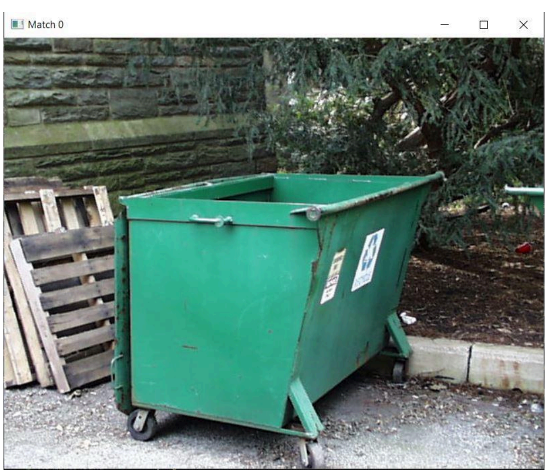
Task-1 Baseline Matching :

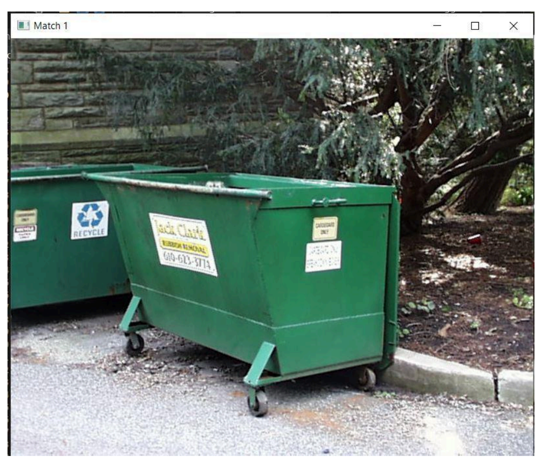


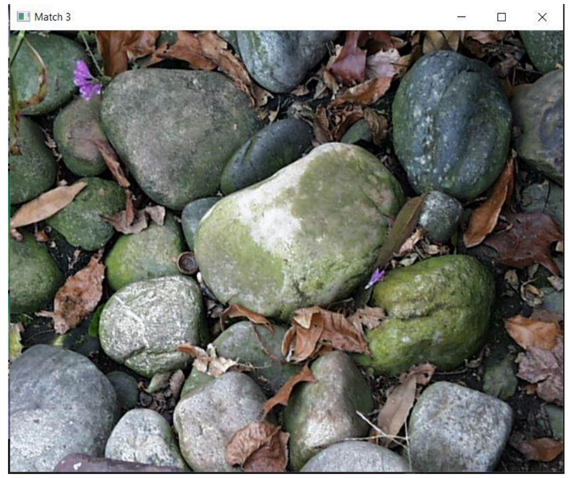




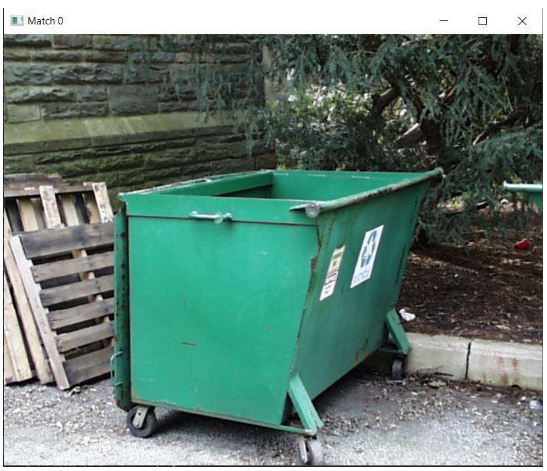
Task-2:

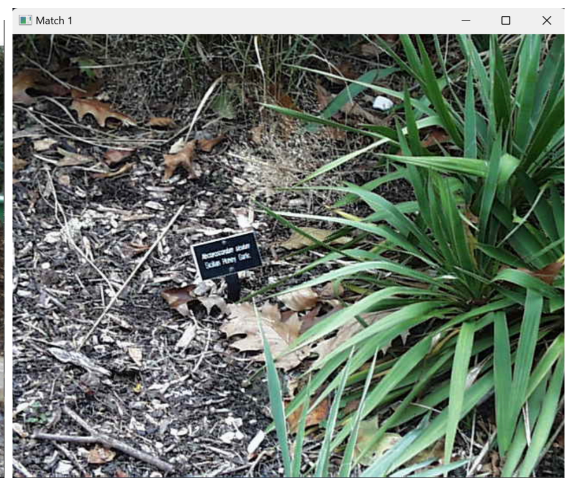


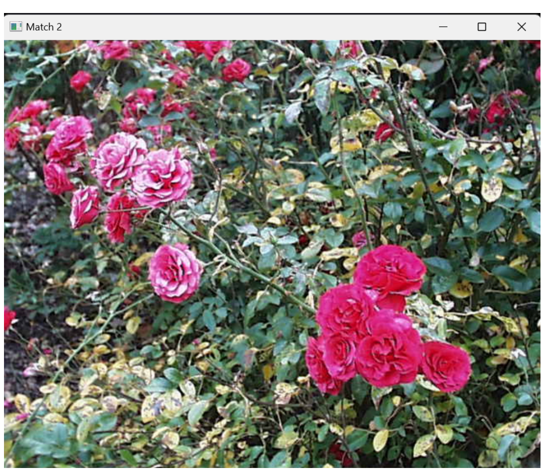


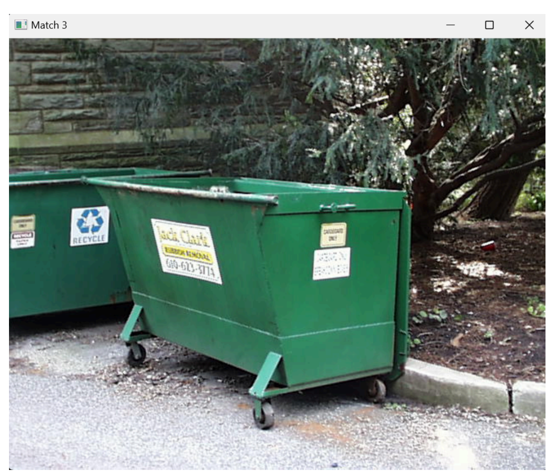


Task-3 Multi-histogram Matching:

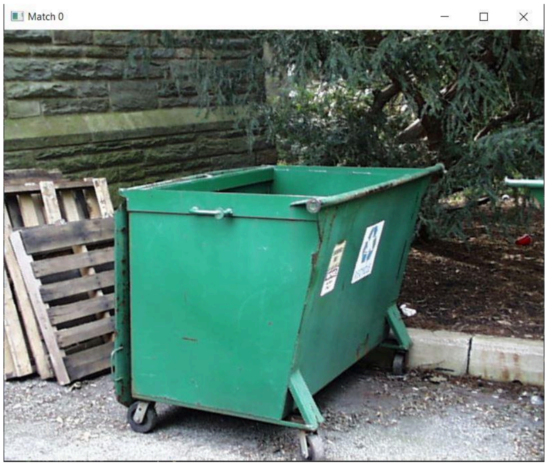








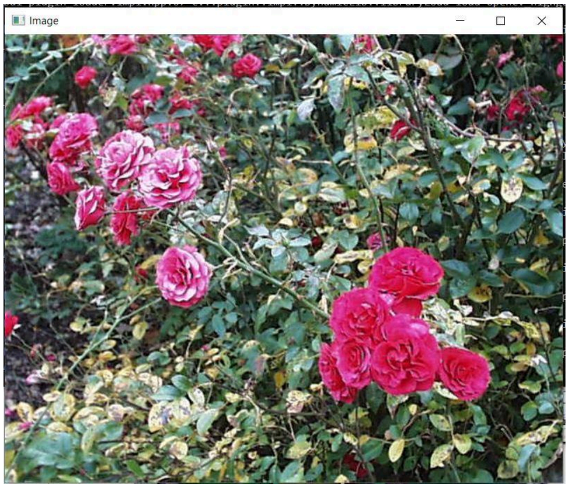
Task-4 Texture and Color:

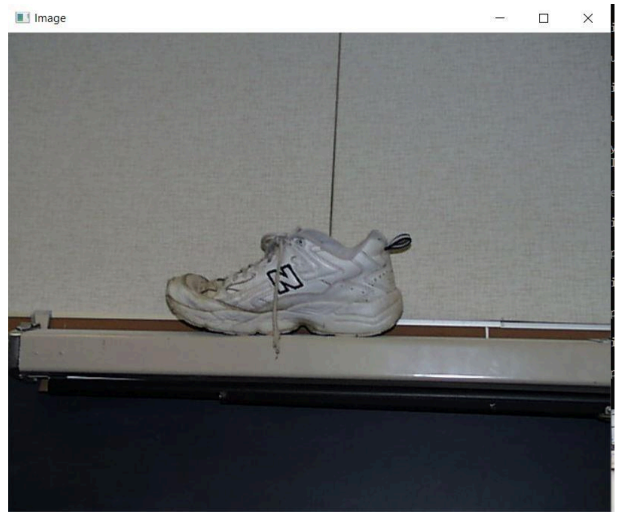


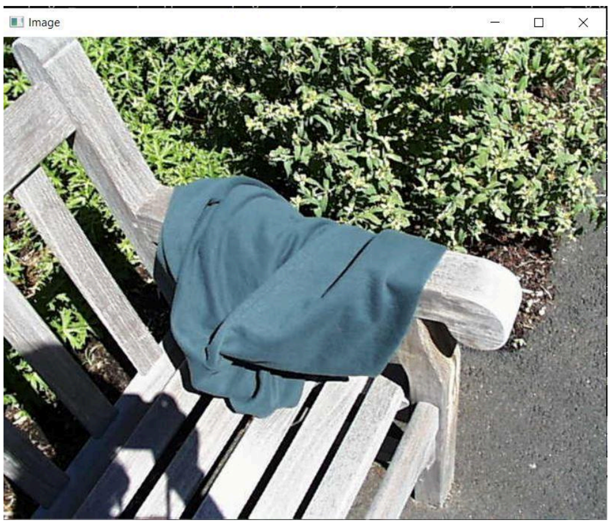
Case-2 Target Image: Pic.1072:

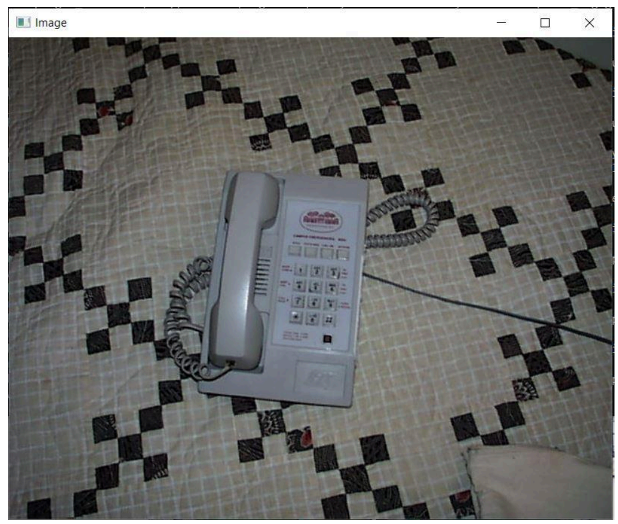
Task-5

Task-1

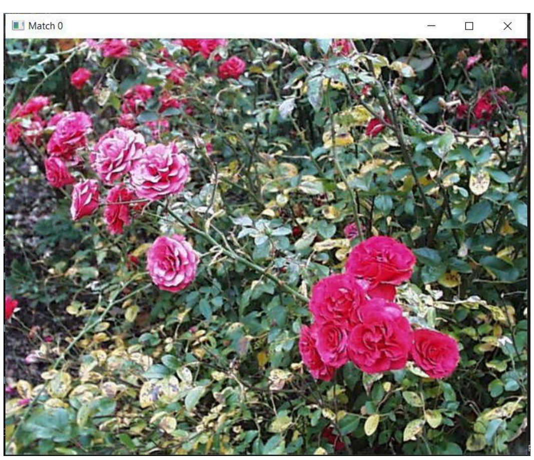


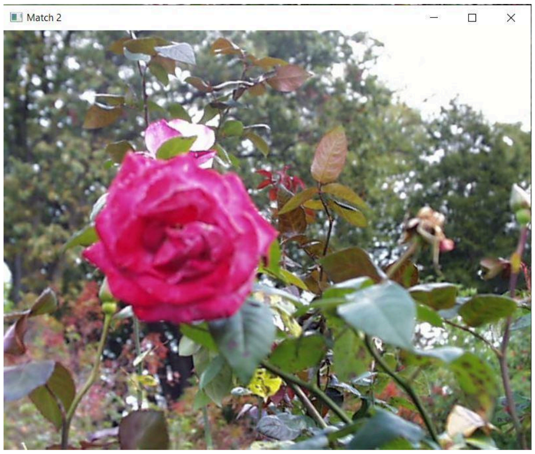


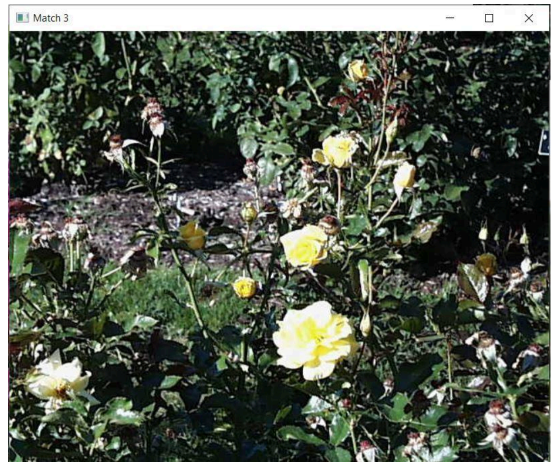




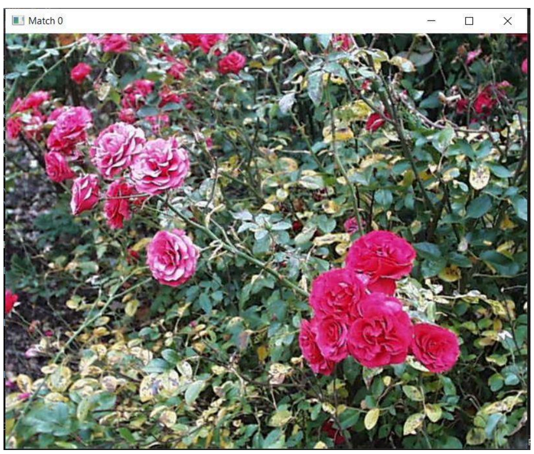
Task-2

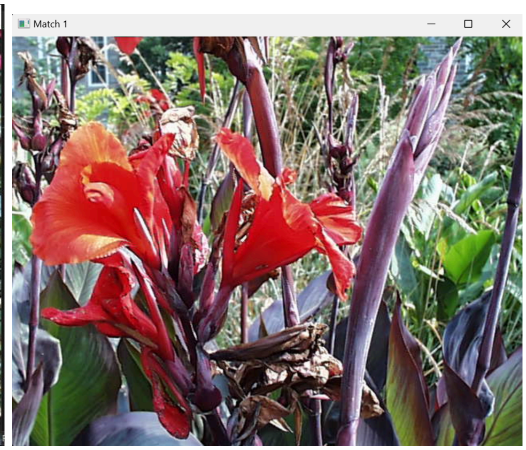


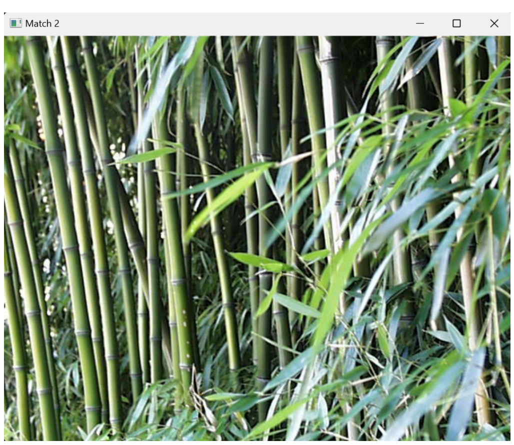


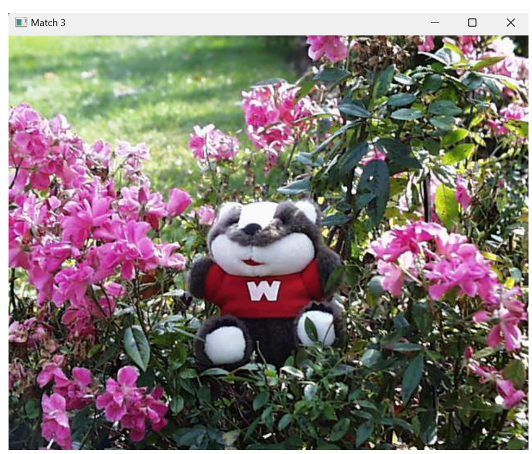


Task-3:

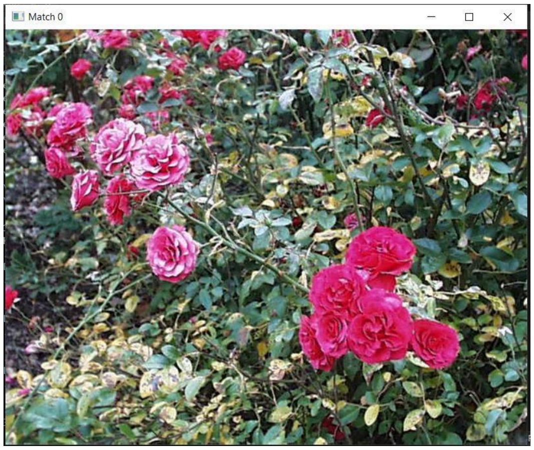


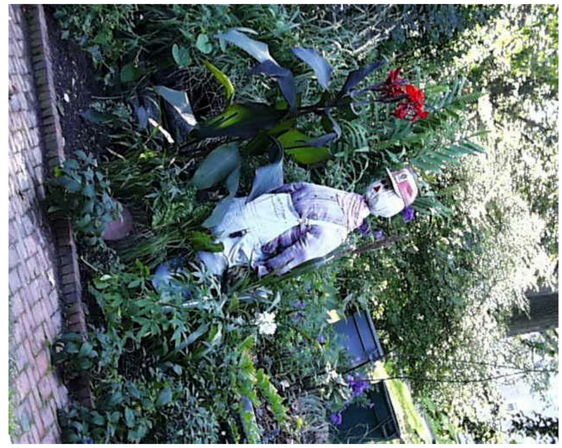






Task-4:





Is the DNN Embedding Vector always better?

In comparing the results between DNN embeddings and classical feature vectors, distinct patterns emerge. DNN embeddings consistently yield more accurate matches compared to traditional methods (1-4). While baseline matching, which focuses on a small 7x7 square at the image center, often produces inaccurate results unless the target image's feature vectors closely resemble those of the database images, histogram matching significantly outperforms baseline matching in detection accuracy. Program 2 demonstrates consistent matches, albeit heavily reliant on color and potentially overlooking crucial features like texture. Multi-histogram matching exhibits improvement, especially in identifying images dominated by specific colors such as pic.1072. Notably, the program integrating both texture and color features excels further, allowing for adaptable bias adjustment between the two, resulting in diverse and effective matching outcomes.

On a closing note, DNN embeddings are adept at capturing complex visual data and intricate patterns, rendering them valuable for diverse image analysis tasks. Their capacity to extract high-level semantic features and abstract representations makes them particularly effective in handling complex and diverse image datasets. However, classical features may outperform DNN embeddings in certain scenarios, such as when dealing with images exhibiting simple or repetitive patterns, or with small and homogeneous datasets. Moreover, the computational demands of DNN embeddings, necessitating substantial resources for both training and inference, must be taken into account. Thus, the selection between DNN embeddings and classical features hinges on the specific requirements and characteristics of the task at hand, underscoring the importance of empirical evaluation for optimal feature extraction method selection.

Task - 7: Custom Design:

We integrated both color and texture analysis techniques to enhance image characterization and similarity retrieval. It leverages a 2D color histogram in the HSV (Hue, Saturation, Value) color space to capture the distribution of colors in the target and dataset images, providing insights into their color composition and variations. Additionally, the code employs the Gabor filter to extract texture features from the images, yielding a texture histogram that reflects the textural properties of the scenes. By combining these two histograms, the algorithm comprehensively characterizes images based on both color and texture attributes

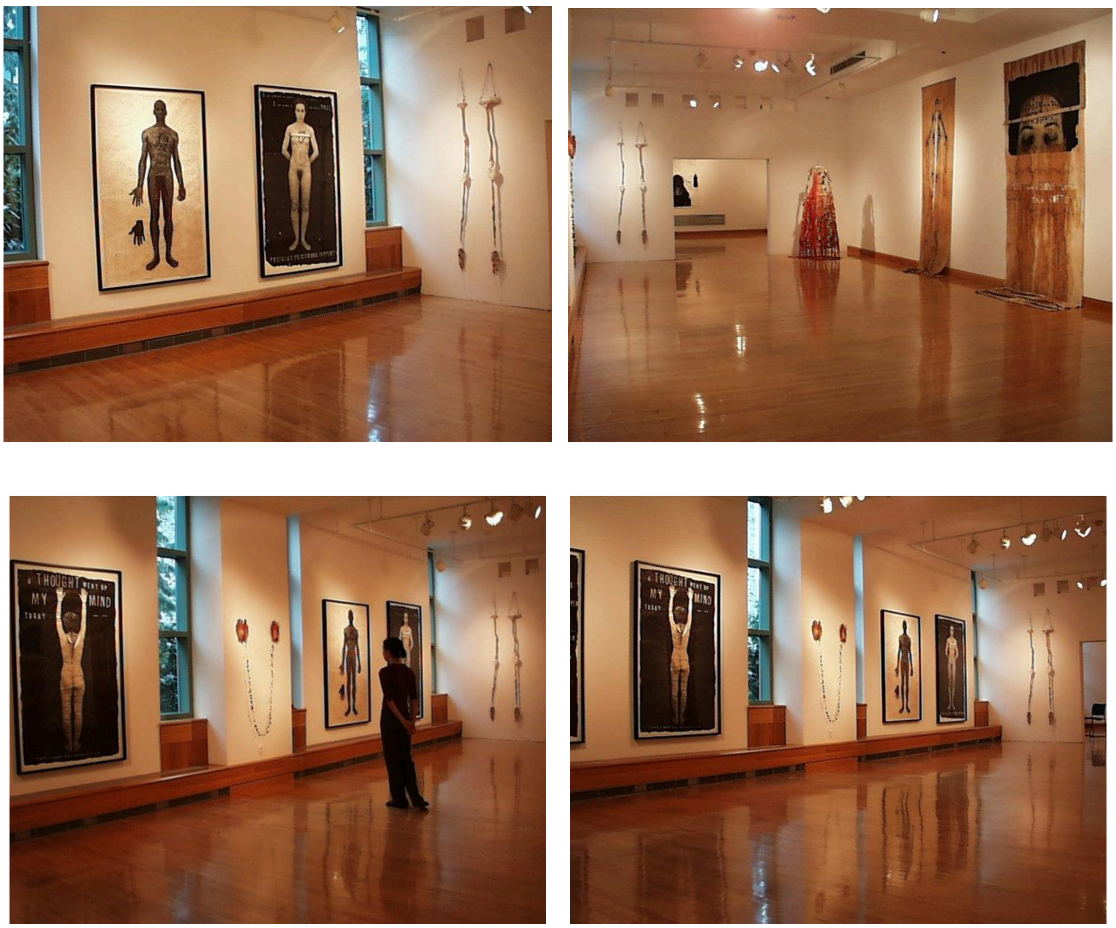
Results:

In our approach for image retrieval, we used the Gabor filter and 2-D color histogram, to retrieve the images based on the target image. Leveraging the spatial recognition and edge detection capabilities of the Gabor filter, we meticulously detected intricate textures and patterns within the target image. This allowed us to capture the essence of the scene, from the detailed Egyptian paintings to the nuanced texture of the wooden flooring.

Additionally, we calculated the histogram of pixel intensities for each greyscale image, allowing us to quantitatively analyze the distribution of intensity values across the images. This step provided valuable insights into the overall intensity characteristics of the images, further enriching our feature extraction process.

In parallel, we computed a 2D color histogram for each image, capturing the distribution of color values in both the hue and saturation dimensions. This comprehensive color representation enabled us to effectively characterize the color composition of the museum scene, enhancing our ability to identify images with similar color distributions.

Moreover, by seamlessly integrating these two powerful techniques, our methodology excelled in capturing both color and texture features with precision. This holistic approach not only facilitated the identification of matching images but also underscored the versatility of our algorithm in handling diverse image retrieval tasks.



(From left to right: Target image: *pic.019.jpg*, Top 3 matches:pic.025.jpg, pic.0017.jpg, pic.0014.jpg)

Extension - 1 (Fourier Transform-Based Features):

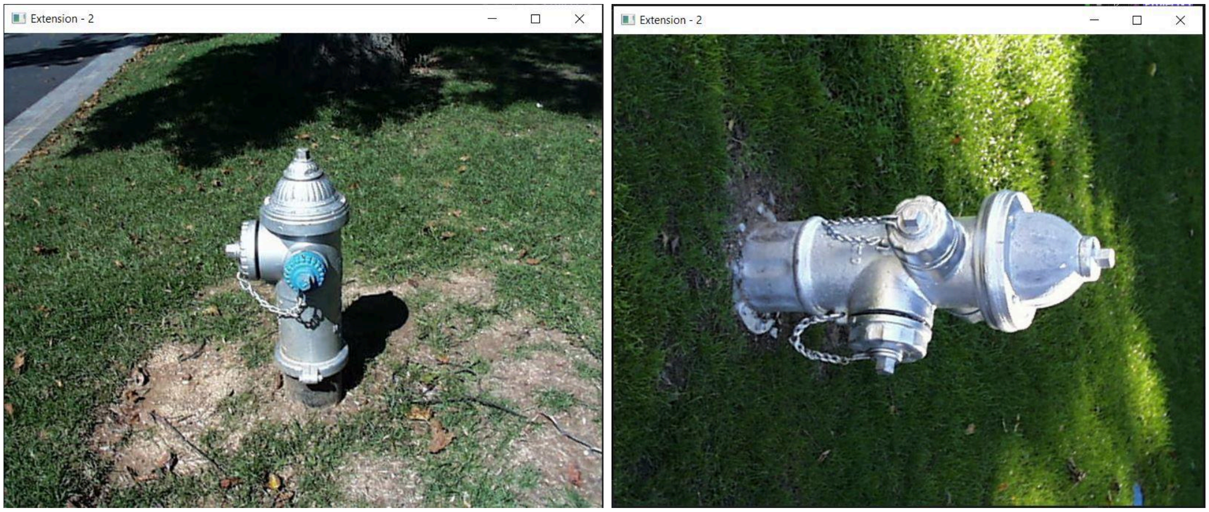
This task is aimed at implementing an image retrieval system utilizing Fourier Transform-based features. The code reads a target image and computes Fourier features to represent its frequency characteristics. These features are compared with those of images in a directory using the sum of squared differences as a distance metric. The comparison enables the identification of similar images based on their frequency domain representations. Unlike traditional color-based methods such as HSV or RGB, this approach focuses on extracting and comparing frequency features. By analyzing the frequency components of images, this system facilitates efficient image similarity search and retrieval, offering a unique perspective on image analysis and comparison.

Results:

The algorithm successfully identified images with similar frequency characteristics to the target image of a red toy duck on a road. The accuracy of the first match, depicting a red duck sitting on a bench, can be attributed to the close resemblance in frequency patterns between the target and matched images. The second and third matches, showcasing similar red toys in different contexts, demonstrate the ability of Fourier Transform to capture common frequency components shared among visually related images. Fourier Transform enables the analysis of image features in the frequency domain, allowing for robust comparison and retrieval of images based on their underlying frequency characteristics.

(From left to right: Target image: pic.0984.jpg, Top 3 matches: pic.986.jpg, pic.1018.jpg, pic.1019.jpg)

Extension-2 Cosine Distance:



(From left to right: Target image: pic.0893.jpg, Top 3 matches: pic.0897.jpg, pic.0136.jpg, pic.0146.jpg)

```
The Top3 Matches for the target image are:
Image Path: pic.0893.jpg
Image Path: pic.0897.jpg
Image Path: pic.0136.jpg
Image Path: pic.0146.jpg
```

We have observed to compared to sum square distance, cosine distance yielded better results when computing the similarity between images using DNN embedding vectors is consistent with the characteristics of cosine distance. Cosine distance considers the angle between vectors rather than their magnitudes, making it robust to variations in scale and well-suited for high-dimensional data like DNN embeddings. In the context of image similarity, DNN embedding vectors encode complex visual features and cosine distance effectively captures the similarity in the orientation of these feature vectors, we think this was the reason for more accurate matches. This preference for cosine distance underscores its utility in scenarios where the relative orientation of vectors is more informative than their absolute magnitudes, as often seen in tasks involving image processing and pattern recognition. Therefore, we think leveraging cosine distance with DNN embeddings aligns well with the goal of capturing nuanced visual relationships and improving the accuracy of similarity assessments between images.

Acknowledgment:

We would like to express our gratitude to Professor Bruce Maxwell for his lectures. His teaching was instrumental in helping us think constructively about our solution. We also extend our thanks to Stack Overflow and to all the developers and contributors on the platform. The wealth of resources available there was invaluable during our debugging process. We appreciate the assistance of OpenAI's ChatGPT, which was useful for conducting deeper research on the general topic. Additionally, we recognize the contributions of the numerous technical writers on Medium who have shared their expertise in computer image processing through their writings.

References:

- PyImageSearch-https://pyimagesearch.com/2020/04/06/blur-and-anonymize-faces-with OpenCV-and-python
- GeeksforGeeks-https://www.geeksforgeeks.org/measure-execution-time-with-high-precision
- YouTube - https://www.youtube.com/watch?v=WXV06M1Qzlo
- YouTube - https://www.youtube.com/watch?v=8jLOx1hD3_o
- YouTube - https://www.youtube.com/watch?v=HS4c3kBEWr4
- AutomaticAddison - https://automaticaddison.com/how-the-Sobel-operator-works/
- OpenCV Doc - https://docs.opencv.org/3.4/db/d64/tutorial_js_colorspaces.html
- https://code.visualstudio.com/docs/cpp/cpp-debug
- https://www.reddit.com/r/computervision/
- https://en.wikipedia.org/wiki/Image_histogram#:~:text=In%20the%20field%20of%20computer,for%20peaks%20and%2For%20valleys.
- https://medium.com/@sasasulakshi/opencv-image-histogram-calculations-4c5e736f85e