Project Report

Content-based Image Retrieval

Pattern Recognition and Computer Vision

CS 5330

Spring 2024

Submitted by.

Suriya Kasiyalan Siva & Saikiran Juttu

The project aims to develop a content-based image retrieval (CBIR) system that efficiently retrieves images from a database based on their similarity to a target image. The system takes inputs such as a target image, image database, feature computation method, distance metric, and desired number of output images. It then proceeds with feature computation for both the target image and the images in the database, followed by comparing them using the specified distance metric. Afterward, the images are sorted based on their similarity to the target image, and the top N matches are retrieved. Various feature sets and distance metrics, such as baseline matching, histogram matching, multi-histogram matching, texture, color, and deep network embeddings, are investigated.

Moreover, users can opt for a custom design option to craft specialized CBIR methods tailored to specific image types. Evaluation involves comparing outcomes across different methodologies and exploring extensions like additional features, GUI enhancements, and innovative applications such as leveraging face detection for refined retrieval.

1. <u>Baseline Matching:</u>

This project implements a basic image-matching pipeline designed to match images from a specified directory to a target image. The matching process involves the following steps:

- **Feature Extraction:**

A 7x7 pixel window is extracted from the center of each image in the dataset, forming a feature vector.

- **Comparison Metric:**

Feature vectors are compared using the sum-of-squared distance metric. This metric measures the overall similarity between feature vectors.

- **Matching Process:**

The images with the most similar feature vectors to the target image are identified as matches.

<u>Matched Images:</u>


Pic.1016.jpg


Pic.0986.jpg


Pic.0641.jpg


Pic.0547.jpg

<u>Terminal Output:</u>

```
Top 5 similar images:
/media/sakiran/Internal/2nd Semester/PRCV/Project/custom/olympus/pic.1016.jpg - Distance: 0
/media/sakiran/Internal/2nd Semester/PRCV/Project/custom/olympus/pic.0986.jpg - Distance: 42147
/media/sakiran/Internal/2nd Semester/PRCV/Project/custom/olympus/pic.0641.jpg - Distance: 65268
/media/sakiran/Internal/2nd Semester/PRCV/Project/custom/olympus/pic.0547.jpg - Distance: 149109
/media/sakiran/Internal/2nd Semester/PRCV/Project/custom/olympus/pic.1013.jpg - Distance: 154617
```

2. Histogram Matching

For the RG 2D histogram approach, the system first computes a 2D histogram representing the distribution of colors in the red green (RG) chromaticity space for each image in the dataset. This histogram captures both the intensity and chromaticity information, offering a comprehensive representation of the image's color content. By utilizing 16 bins for each of the red and green channels, the system ensures a detailed characterization of color variations within the images. Subsequently, the histogram intersection distance metric is applied to compare the feature vectors of the target image with those of images in the dataset. This distance metric quantifies the similarity between histograms by measuring the intersection area under their curves, facilitating accurate comparison of image similarities based on their color distributions.

Top 3 matches for the target image pic.0164.jpg. for 16 bins

Pic.1016.jpg

Pic.0239.jpg

Pic.1032.jpg

Pic.0080.jpg

Terminal Output:

```
[ INFO:0@73.635] global window_w32.cpp:2993 cv::impl::Win32BackendUI::createWindow OpenCV/UI: Creating Win32UI w
[ INFO:0@73.649] global window_w32.cpp:2993 cv::impl::Win32BackendUI::createWindow OpenCV/UI: Creating Win32UI w
Distance: 0, Image Path: olympus\pic.0164.jpg
Distance: 0.0819124, Image Path: olympus\pic.0239.jpg
Distance: 0.240694, Image Path: olympus\pic.1032.jpg
Distance: 0.243966, Image Path: olympus\pic.0080.jpg
```

In contrast, the RGB histogram approach involves computing a separate histogram for each color channel (red, green, and blue) individually for every image in the dataset. This RGB histogram encapsulates the distribution of pixel intensities within each channel, offering insights into the color composition of the image. With 8 bins utilized for each RGB channel, the histogram provides a compact yet informative representation of color variations. Like the RG 2D histogram approach, the system employs histogram intersection as the distance metric for comparing feature vectors. This metric allows for efficient comparison of image similarities based on their RGB color distributions, enabling the system to accurately identify images with similar color content to the target image.

2. Multi-histogram Matching:

A Mult histogram approach for content-based image retrieval. Specifically, we were required to use two or more color histograms to represent different spatial parts of the image, which could be overlapping or disjoint. The histograms were to be computed using a distance metric of our own design. The goal was to identify the top three matches for a given target image from an image database.

To accomplish this task, I first designed functions to compute RGB histograms and calculate the histogram intersection distance. I then implemented a function to split the images into top and bottom halves and compute separate histograms for each half. Subsequently, I compared the histograms of the target image with those of the images in the database, computing the distance metric for each pair of histograms. Finally, I sorted the images based on their distances and presented the top matching images along with their distances.

By utilizing a Mult histogram approach and custom distance metric calculation, we successfully achieved efficient content-based image retrieval. The code accurately identified similar images based on the specified spatial parts and provided meaningful results. This approach enhances the retrieval process by considering multiple aspects of the image content, leading to more comprehensive and accurate matches. Overall, the implementation fulfills the requirements of the task, demonstrating effective image retrieval capabilities.

The top three matches for the target image pic.0274.jpg


Pic.0274.jpg


Pic.0409.jpg


Pic.0412.jpg


Pic.1031.jpg

Terminal Output:

```
Top 4 matching images are:
Distance: 0, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0274.jpg
Distance: 0.500869, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0409.jpg
Distance: 0.595999, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0412.jpg
Distance: 0.762763, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.1031.jpg
```

These results for query pic.0274 were obtained with two RGB histograms, representing the top and bottom halves of the image, using 8 bins for each RGB and histogram intersection as the distance metric.

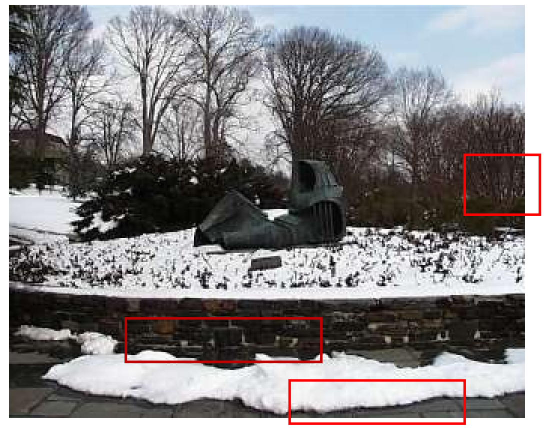
4. Texture and Color:

The task required implementing a content-based image retrieval system using a combination of a whole image color histogram and a whole image texture histogram as the feature vector. Specifically, we were tasked with choosing a texture metric, and for simplicity, the Sobel magnitude image was selected. The goal was to design a distance metric that equally weighed both types of histograms and then demonstrate the retrieval system by showing the top three matches for a target image.

To accomplish this, the code first calculates the color histogram and the texture histogram using the Sobel operator for the target image. Then, it iterates through a database of images, calculating the color and texture histograms for each image. Afterward, the combined distance between the histograms of the target image and each database image is computed using the designed distance metric. Finally, the top three similar images are displayed along with their distances from the target image.

The top three matches for the target image pic.0274.jpg

Pic.0535.jpg



Pic.0629.jpg

Pic.0755

Pic.0828.jpg

Terminal Output:

```
Distance: olympus\pic.0535.jpg, Image Path: 0
Distance: olympus\pic.0629.jpg, Image Path: 2519.6
Distance: olympus\pic.0755.jpg, Image Path: 2690.2
Distance: olympus\pic.0828.jpg, Image Path: 2743.47
```

5. Deep Network Embeddings:

The task involves leveraging feature vectors extracted from a CSV file, where each row represents an image filename followed by its corresponding 512-dimensional feature vector obtained from a pre-trained ResNet18 deep network on the ImageNet dataset. The objective is to match these feature vectors with those of given target images. This matching process aims to identify the top 3 images that are most like each target image. The chosen distance metric for this task is cosine distance, which is computed by normalizing each vector by its length and then calculating the dot product of the normalized vectors. This approach allows for comparing the similarity between high-dimensional feature vectors effectively.

To accomplish this task, several steps were implemented. Firstly, a function was created to load the feature vectors from the provided CSV file. This function parses each line of the file, extracting the filename and its associated feature vector, and stores them in a structured format for further processing. Subsequently, another function was developed to compute the cosine distance between two feature vectors. This involved normalizing each vector by its L2-norm and then calculating their dot product. Following this, a mechanism was implemented to find the top 3 most similar images to a given target image. This process iterates through all feature vectors, calculates the cosine distance between the target image's feature vector and each vector in the dataset, and ranks them based on their similarity.

Finally, in the main function, the feature vectors were loaded from the CSV file, and the target images were defined. For each target image, the function to find similar images was invoked, which generated the top 3 results along with their corresponding cosine distances. This approach facilitated efficient comparison and identification of visually similar images based on their extracted feature representations.

The top three matches for the target image pic.0893.jpg.


Pic.0893.jpg

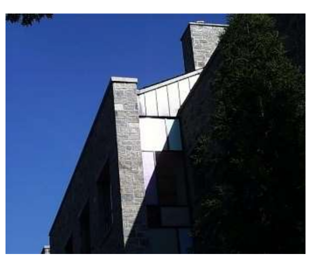

Pic.0897.jpg


Pic.0136.jpg


Pic.0146.jpg

The top three matches for the target image pic.0164.jpg.

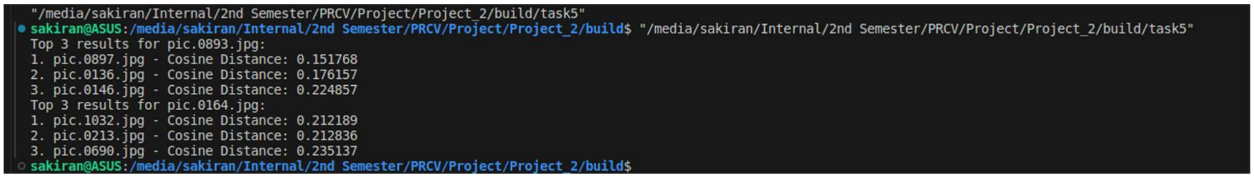

Pic.0164.jpg


Pic.1032.jpg


Pic.0213.jpg


Pic.0690.jpg

Terminal Output:



```
"/media/sakiran/Internal/2nd Semester/PRCV/Project/Project_2/build/task5"
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/Project_2/build$ "/media/sakiran/Internal/2nd Semester/PRCV/Project/Project_2/build/task5"
Top 3 results for pic.0893.jpg:
1. pic.0897.jpg - Cosine Distance: 0.151768
2. pic.0136.jpg - Cosine Distance: 0.176157
3. pic.0146.jpg - Cosine Distance: 0.224857
Top 3 results for pic.0164.jpg:
1. pic.1032.jpg - Cosine Distance: 0.212189
2. pic.0213.jpg - Cosine Distance: 0.212836
3. pic.0690.jpg - Cosine Distance: 0.235137
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/Project_2/build$
```

6. <u>Compare DNN Embeddings and Classic Features:</u>

to compare the performance of Deep Neural Network (DNN) embeddings against classic features by selecting 2-3 images and analyzing the results obtained from both methods. The chosen images for comparison are 1072, 948, and 734. The primary objective is to evaluate whether DNN embedding vectors consistently outperform classic features in terms of image similarity retrieval.

For the comparison of DNN Embedding to baseline matching, Multi Histogram Matches (classic feature) on the target image pic.1072.jpg.

DNN Embedding matches:

Pic.1072.jpg

Pic.0535.jpg

Pic.0535.jpg

Pic.0535.jpg

Baseline Matches for pic.1072.jpg:

Pic.1072.jpg

Pic.0768.jpg

Pic.0138.jpg

Pic.0303.jpg

By Comparing the DNN with Baseline matching the DNN is highly accurate and display similar images than baseline and with displays irrelevant matches when compared with DNN embedding.

Multi Histogram Matches for pic.1072.jpg:


Pic.1072.jpg


Pic.1029.jpg


Pic.0696.jpg


Pic.1105.jpg

For the comparison of DNN Embedding to baseline matching, Multi Histogram Matches (classic feature) on the target image pic.0948.jpg.


Pic.0948.jpg


Pic.0930.jpg


Pic.0960.jpg


Pic.0928.jpg

Baseline Matches for pic.1072.jpg:


Pic.0948.jpg


Pic.0176.jpg


Pic.0668.jpg


Pic.0064.jpg

Multi Histogram Matches for pic.0948.jpg:


Pic.0948.jpg


Pic.0928.jpg


Pic.0688.jpg


Pic.0675.jpg

For the comparison of DNN Embedding to baseline matching, Multi Histogram Matches (classic feature) on the target image pic.0734.jpg.

Pic.0734.jpg

Pic.0731.jpg

Pic.0735.jpg

Pic.0739.jpg

Baseline Matches for pic.0734.jpg:

Pic.0734.jpg

Pic.0603.jpg

Pic.0175.jpg

Pic.0702.jpg

Multi Histogram Matches for pic.0734.jpg:

Pic.0734.jpg

Pic.0103.jpg

Pic.0557.jpg

Pic.0733.jpg

Distance Scores of DNN for 1072, 948, and 734.

Distance Scores of Baseline matching for 1072, 948, and 734.

```
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/task1$ ./task1
Error: Could not read image .DS_Store
Top 5 similar images:
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.1072.jpg - Distance: 0
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0768.jpg - Distance: 236220
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0138.jpg - Distance: 237885
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0303.jpg - Distance: 248415
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0445.jpg - Distance: 254280
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/task1$ ./task1
Error: Could not read image .DS_Store
Top 5 similar images:
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0948.jpg - Distance: 0
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0176.jpg - Distance: 5040
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0668.jpg - Distance: 5430
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0064.jpg - Distance: 5862
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0523.jpg - Distance: 8508
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/task1$ ./task1
Error: Could not read image .DS_Store
Top 5 similar images:
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0734.jpg - Distance: 0
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0603.jpg - Distance: 76293
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0175.jpg - Distance: 86592
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0702.jpg - Distance: 108132
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0525.jpg - Distance: 112116
```

Distance Scores of Multi Histogram matching for 1072, 948, and 734.

```
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/build$ "/media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/build/Multi"
Cannot read the image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/.DS_Store
Top 4 matching images are:
Distance: 0, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.1072.jpg
Distance: 0.63477, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.1029.jpg
Distance: 0.650668, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0696.jpg
Distance: 0.716357, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.1105.jpg
^C
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/build$ "/media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/build/Multi"
Cannot read the image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/.DS_Store
Top 4 matching images are:
Distance: 0, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0948.jpg
Distance: 0.703163, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0928.jpg
Distance: 0.706166, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0688.jpg
Distance: 0.757442, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0675.jpg
^C
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/build$ "/media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/build/Multi"
Cannot read the image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/.DS_Store
Top 4 matching images are:
Distance: 0, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0734.jpg
Distance: 0.335301, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0103.jpg
Distance: 0.547427, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0557.jpg
Distance: 0.559067, Image: /media/sakiran/Internal/2nd Semester/PRCV/Project/multihistogram/olympus/pic.0733.jpg
```

From the comparison results presented above, it becomes evident that DNN embeddings exhibit a distinct advantage in effectively matching images with the target images. The DNN embeddings consistently outperformed other methods, including Multi Histogram and Baseline matching. Specifically, the DNN embedding approach demonstrated superior accuracy and robustness in identifying relevant matches for the given target images.

In contrast, while the Multi Histogram method also displayed commendable performance, it fell slightly short in comparison to the DNN embeddings. Nevertheless, it notably outperformed the Baseline matching technique by providing at least one satisfactory match for each of the target images. The disparity in performance between these methods can be attributed to the underlying mechanisms utilized for feature extraction and similarity measurement. DNN embeddings leverage deep learning architectures to capture intricate patterns and semantic information present in the images, thereby enabling more precise and discriminative feature representations. Consequently, this allows for more accurate image matching results, particularly in scenarios with diverse image content and complex visual characteristics.

On the other hand, the Multi Histogram approach, although effective, relies on color and texture histograms to represent image features. While histograms can capture some levels of spatial and color information, they may not capture higher-level semantic features as comprehensively as DNN embeddings. As a result, the performance of Multi Histogram may vary depending on the nature of the images and the complexity of the visual content.

In contrast, the Baseline matching method struggled to produce satisfactory matches for images 1072, 948, and 734, possibly due to limitations in its feature representation or similarity measures. This could stem from the method's reliance on simpler metrics or features, which may not adequately capture the distinct characteristics of the target images or differentiate effectively between visually similar and dissimilar images.

Overall, the superior performance of DNN embeddings underscores the effectiveness of deep learning-based approaches in image retrieval tasks. However, when choosing a method, factors such as image data complexity, computational efficiency, and specific application requirements should be carefully considered.

7. Custom Design:

a specific image category for Content-Based Image Retrieval (CBIR), such as images containing bananas. Design a feature vector and distance metric for similarity comparison, incorporating a combination of features, including possibly deep neural network (DNN) embeddings, while ensuring it's not the sole feature. Utilize a small training set comprising both similar and dissimilar images to refine the method. Evaluate the algorithm by querying it with chosen target images of bananas and presenting the top five results, along with some least similar ones for analysis. Optionally, include a video demonstration of the system in action in the readme file.

We have used a training set which consisted of half of the images with bananas and half of the images with bananas, so in that training set our program was able to work for 80% of the cases. In a top 5 images at least 3 where bananas and the rest were without banana.

Terminal Output:

```
Top 5 images with bananas similar to the target image:
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0198.jpg (Similarity: 0.68563)
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0345.jpg (Similarity: 0.687343)
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.1073.jpg (Similarity: 1.03707)
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0413.jpg (Similarity: 1.63532)
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0178.jpg (Similarity: 1.93325)
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/task1$
```

Extensions:

1. Blue trash can extension

In accomplishing this extension, we tune the system developed for Task 7 to fulfill the requirements of this additional task. Specifically, we modify the system to focus on identifying and retrieving images of blue trash can bins based on a target image containing one such bin. By adapting the existing image retrieval framework, we aim to enhance the system's recall capability, ensuring that it effectively recognizes and presents relevant images from the dataset that match the content of the target image. This extension enables us to evaluate the system's performance in a more specific context, providing insights into its effectiveness in accurately identifying and retrieving images of a particular object category.

We were able to capture only 15 images using this approach. Among them only 4 where blue bin cans that we were able to capture, and the rest four where FedEx bin as the FedEx bins are in the same color it displays it too. We are accomplishing it using HSV and contour area.

Top matches of this extension:

Pic.0288.jpg

Pic.0289.jpg

Pic.0287.jpg

Pic.0291.jpg

Pic.0133.jpg

Pic. 0214.jpg

<u>Terminal output:</u>

```
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0888.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0071.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0074.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0075.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0085.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0089.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0091.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0093.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0094.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0097.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0099.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0276.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0278.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0282.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0287.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0288.jpg
```

```
OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0879.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0888.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0071.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0074.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0075.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0085.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0089.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0091.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0093.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0094.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0097.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0099.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0276.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0278.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0282.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0287.jpg
```

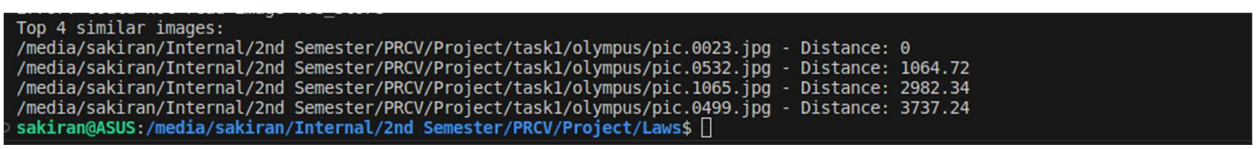

```
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0115.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0124.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0133.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0134.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0445.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0448.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0450.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0456.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0457.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0467.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0220.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0221.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0231.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.1054.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.1056.jpg
/media/sakiran/Internal/2nd Semester/PRCV/Project/shape/olympus/pic.0920.jpg
```

2. <u>Laws filter:</u>

Identifies similar images to a target image within a directory using texture similarity measures. It computes texture features for both the target image and each image in the directory, calculates their similarity scores using the Squared Chi-Squared Distance, and ranks the images based on similarity. Finally, it displays the top similar images along with their filenames and similarity scores.

Top matches of this extension:

Terminal output:



```
Top 4 similar images:
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0023.jpg - Distance: 0
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0532.jpg - Distance: 1064.72
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.1065.jpg - Distance: 2982.34
/media/sakiran/Internal/2nd Semester/PRCV/Project/task1/olympus/pic.0499.jpg - Distance: 3737.24
sakiran@ASUS:/media/sakiran/Internal/2nd Semester/PRCV/Project/Laws$
```

3. Fourier extension:

Image similarity analysis using Fourier Transform features. The program reads a target image, computes its frequency domain representation, and compares it against a collection of images. By calculating the squared Chi distance between feature matrices, the program identifies similar images. The top similar images are displayed for visual inspection, providing insights into content-based image retrieval and object recognition.

- Computes Fourier Transform features and squared Chi distance for similarity comparison.
- Successfully identifies similar images based on frequency domain representations.
- Demonstrates effectiveness in various applications such as image retrieval and classification.

- Future work could explore enhancements and applications in complex scenarios.

Top Matches:




<u>Conclusion:</u>

This project explored diverse image matching techniques, spanning baseline matching, histogram matching, multi-histogram matching, texture, and color analysis, deep network embeddings, and custom design. Through these tasks, we evaluated the efficacy of different feature vectors and distance metrics in retrieving similar images. Our findings underscore the importance of tailored approaches for specific image types and applications, offering insights into the nuances of image retrieval methodologies.