Report 2 — Content-based Image Retrieval

> Edited by Hui Hu @hui hwoo and Yao Zhong @Zhong Yao

Description

The objective of the project is to gain experience in manipulating and analyzing images at a pixel level, as well as performing image matching and pattern recognition using generic image characteristics such as color, texture, and spatial layout. The task involves finding similar images in a database using color spaces, histograms, spatial features, and texture features, without using neural networks or object recognition methods.

Tasks

Task 1 : Baseline Matching

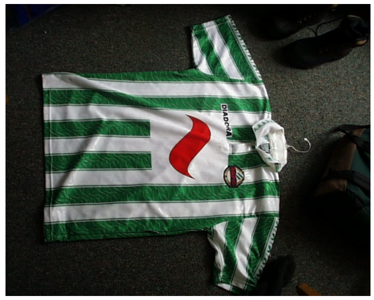
> In this task, the goal is to use the 9x9 square in the middle of the image as a feature vector and compare it to other images using the sum-of-squared-difference as the distance metric. The program should ensure that comparing the image to itself results in a distance of 0. The task is aimed at getting the overall pipeline working

Target Image:

pic.1016.jpg

Top three matches (pic.0986.jpg, pic.0641.jpg, pic.0547.jpg**):**

pic.0986.jpg



pic.0641.jpg

pic.0547.jpg

Task 2 : Histogram Matching

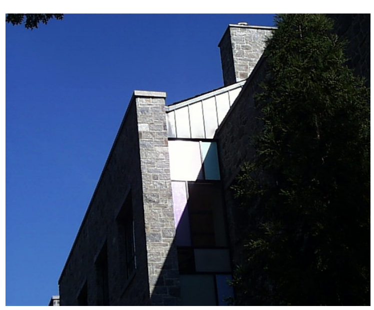
> In this task, a single normalized color histogram is used as the feature vector. We chose the RGB histogram and allocated 8 bins for each of them. Therefore, the feature vector is of length 8*8*8. And then the histogram is normalized by the total number of pixels. The histogram intersection is used as the distance metric.

Target Image:

pic.0164.jpg

Top three matches (pic.0110.jpg, pic.1032.jpg, pic.0976.jpg**):**

pic.0110.jpg



pic.1032.jpg

pic.0976.jpg

Task 3 : Multi-histogram Matching

> In this task, two or more color histograms are used as the feature vector to represent different spatial parts of the image, which can either be overlapping or disjoint. The task requires using a custom distance metric to compare the corresponding histograms. For example, histogram intersection could be used to compare the histograms and weighted averaging could be used to combine the distances between the different histograms.

We used the whole-image histogram and the histogram that looks at only the center of the image to get the feature vector and give different weight on the two histogram.

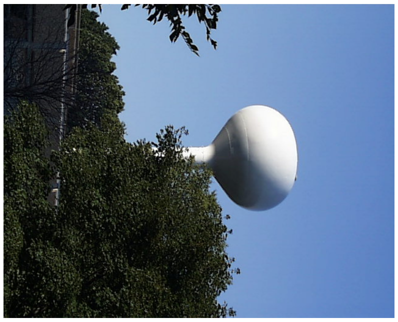
Target Image:

pic.0274.jpg

Top three matches (pic.1055.jpg, pic.0273.jpg, pic.0209.jpg**):**

pic.1055.jpg

pic.0273.jpg



pic.0209.jpg

Task 4: Texture and Color

> In this task, we combined the color histogram and texture histogram and uses the combined feature as the feature vector. As for the color histogram, we used the same

> RGB 8*8*8 with task 2. As for the texture histogram, we chose the gradient magnitude as the texture feature. The magnitude is also counted by 8 bins in each channel and then normalized. We choose the intersection for both feature vectors. And we weighted them equally by giving both of the intersections weight of 0.5. (We also tried other texture features and displayed them in the extensions).

Target Image:

pic.0535.jpg

Top three matches (pic.0733.jpg, pic.0454.jpg, pic.0255.jpg**):**

pic.0733.jpg

pic.0454.jpg

pic.0255.jpg

> What we can find different from task 2 and 3 is, in this matching, it is no longer just considering the color(the dark-red and orgrange areas). The top matches all have a **wall** that is similar to that in the target image. This **texture of the walls** is considered in the matches too.

Task 5 : Custom Design

> In this task, a combination of the color histogram and spatial variance histogram is used. The color histogram is similar to that in task 2 and task 4. And the spatial

> variance histogram is calculated in the following method: 1. use the RGB 8*8*8 bins to classify pixels into color bins. 2. count the color pixels and the mean x(row position) and y(col position) values in each color bin. 3. calculate the Euclidean distance of each pixel from the mean location of the pixels in the bin. 4. get the standard deviation of each color bin as the feature vector.

> To compare two spatial variances, we calculated the overlap(intersection) between each color bin. And to normalize the result, the total overlap rate is divided by all the bins that are calculated in the overlap rate.

> Through the training set that we used, I chose to weigh color 0.4 and spatial variance 0.6 in the final distance.

Target Image:

pic.0461.jpg

Top 10 matches

pic.0593.jpg

pic.0276.jpg

pic.0221.jpg


pic.0510.jpg


pic.0478.jpg


pic.0887.jpg


pic.1040.jpg


pic.0485.jpg


pic.0201.jpg


pic.0459.jpg

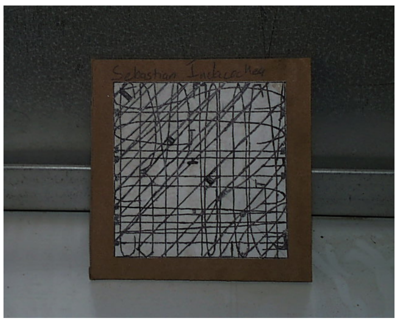

pic.0344.jpg

Top 10 matches:

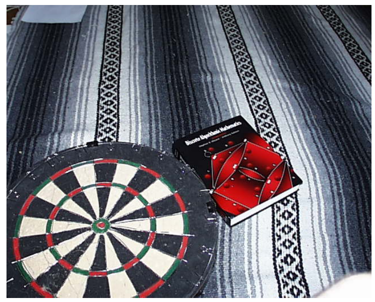
pic.0654.jpg

pic.0617.jpg

pic.1058jpg



pic.1053.jpg



pic.0176.jpg

pic.0611.jpg

pic.0160.jpg

pic.0221.jpg

pic.0681.jpg

pic.0962.jpg

Extensions

Extension 1 : explore more features and matching methods

> We explore histogram of gradient orientations, features of co-occurrence matrices (energy, entropy, contract, and max probability) and histograms of Laws filter responses, and try to find their similarity and difference.

Target Image:

pic.0535.jpg

1. **Histogram of gradient orientations**

   This feature representation is based on the gradient orientation of the pixels in an image. The gradient orientation of each pixel is calculated, and then a histogram is created to represent the distribution of gradient orientations in the image. This feature representation provides information about the direction of the edges and textures in an image.

   Top three matches (pic.0087.jpg, pic.0734.jpg, pic.0568.jpg**):**

pic.0087.jpg

pic.0734.jpg

pic.0568.jpg

2. **Features of co-occurrence matrices**

   Co-occurrence matrices are matrices that represent the relationship between pairs of pixels in an image. They are commonly used to analyze the texture of an image. The most common features extracted from co-occurrence matrices are energy, entropy, contract, and maximum probability. Energy measures the homogeneity of the texture in the image. Entropy measures the randomness or disorder of the texture in the image. Contract measures the directionality of the texture in the image. Maximum probability measures the most frequently occurring pair of pixels in the image.

Top three matches (pic.0216.jpg, pic.0572.jpg, pic.0307.jpg**):**


pic.0216.jpg


pic.0572.jp


pic.0307.jpg

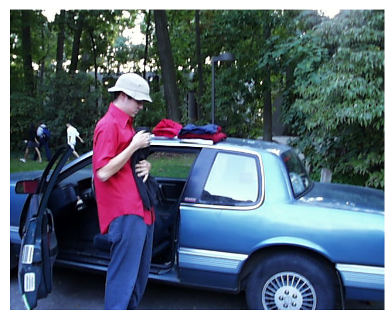
3. **Histograms of Laws filter responses**

   Laws filter responses are the responses of a set of filters to an image. The filters are designed to capture specific features of the image such as edges, lines, or textures. The responses of the filters are then used to create a histogram, which represents the distribution of the filter responses in the image. This feature representation provides information about the texture of the image and is often used for texture classification.

   Top three matches:


pic.1050.jpg


pic.0927.jpg


pic.0477.jpg

4. **Conclusion**

   In terms of similarity, all three feature representations can be used to analyze the texture and color of an image. However, they do so in different ways and provide different types of information about the image.

   In terms of differences, Histograms of gradient orientations provide information about the direction of the edges and textures in an image, while co-occurrence matrices provide information about the texture and color relationship between pairs of pixels, and histograms of Laws filter responses provide information about the texture of the image by using a set of filters.

Reflection

The process of completing this project taught me a great deal about image manipulation and analysis at the pixel level. It was also my first time working on matching or pattern recognition, which was a valuable experience. Working with different color spaces, histograms, spatial features, and texture features helped me understand how these factors play a role in determining image similarity.

This project has given me a deeper understanding of how images can be analyzed and compared to find similarity. I am confident that the skills and knowledge I gained from this project will be useful in future projects and my continued learning in computer vision.

Acknowledgement

[1] *OpenCV modules*. OpenCV. (n.d.). Retrieved February 10, 2023, from https://docs.opencv.org/4.x/