Content-based Image Retrieval (CBIR)

Rahul Kumar
Northeastern University

1. SUMMARY

The purpose of this project is to implement Content-based Image Retrieval techniques that search through the Image database and provides the best match to the queried image. It is based on the query-by-example method in which the user provides a target image and the objective is to display the most similar based on the contents and features of the image. The CBIR is implemented in mainly two parts: feature extraction to form feature vector and feature vector comparison using distance Metric. The feature extraction method adopted in this project is color feature extraction, texture feature extraction, and spatial variance extraction. The features are compared using distance metrics such as Euclidean distance, Manhattan distance, and Histogram intersection. Multiple implementations are done using various combinations of extraction and comparison techniques and corresponding results are discussed.

2. CBIR TECHNIQUES

2.1. Baseline Matching

In this method, a 9x9 square in the middle of the image is selected as the feature vector and the Sum of Squared distance (Euclidean distance) is opted as the distance metric to compare the target image feature vector with the feature vector of images in a database and provides the top three matches.

Target Image (pic.1016)

1st match (pic.0986)

2nd match (pic.0641)

3rd match (pic.0547)

Fig.1. Baseline Matching

Since the feature vector is formed by the middle part of the image with all 3 color channels and thus the retrieved image has a central part in red color same as the target image.

2.2. Histogram Matching

A single normalized 3D color histogram (8x8x8) is used as a feature vector and histogram intersection is used as the distance metric. A color histogram provides the no. of pixels that are present in a defined color range and then the histogram is normalized by dividing with the total pixel count [1]. The histogram intersection correlates the two histograms and measures the overlap between histograms [1]. The retrieved image is shown below for the target image pic. 0164.

Target Image (pic.0164) | 1st match (pic.0110) | 2nd match (pic.1032) | 3rd match (pic.0092)

Fig 2. Histogram Matching

The whole color image is used to form a feature vector and thus the retrieved images have the same color distribution as the target image.

2.3. Multi-histogram Matching

To improve the search result using color histogram, two 3D color histograms are formed using the top half and bottom half of images, and the corresponding feature vector is formed. Then both top and bottom feature vectors are compared with corresponding feature vectors using histogram intersection and the final distance is calculated providing equal weights to each. The retrieved image is shown below for target image pic.0274.

Target Image (pic.0274) | 1st match (pic.0273) | 2nd match (pic.1031) | 3rd match (pic.0409)

Fig 3. Multi-histogram Matching

The 2 color histograms from the top half and bottom half mostly contain sky(blue) and trees (green) respectively and hence the retrieved images contain the same color information and are identified as the best matches.

2.4. Texture and Color Matching

A whole image 3D color histogram and a whole image 1D texture histogram are used as a feature vector and Euclidean distance is designed as a distance metric that provides equal weight to both histograms while calculating the distance. The texture feature is generated using the Sobel magnitude which detects the edges and the texture feature is stored in a 1D histogram. The retrieved image is shown below for target image pic.0535.

Target Image (pic.0535) | 1st match (pic.0171) | 2nd match (pic.0628) | 3rd match (pic.0338)

Fig. 4. Texture and Color Matching result for target image pic.0535

Target Image (pic.0535) | 1st match (pic.0285) | 2nd match (pic.0628) | 3rd match (pic.0952)

Fig. 5. Histogram Matching result for target image pic.0535

Target Image (pic.0535) | 1st match (pic.0285) | 2nd match (pic.0628) | 3rd match (pic.0952)

Fig. 6. Multi-histogram Matching result for target image pic.0535

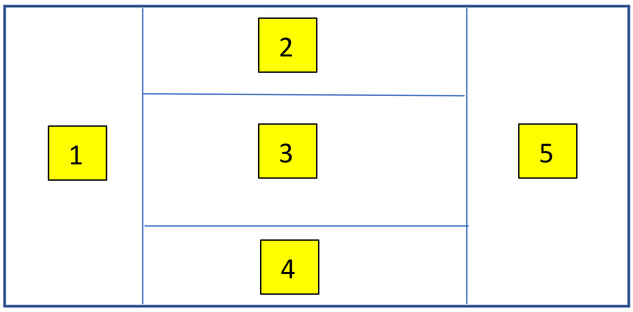
As can be seen from fig.5 and fig. 6, both provide the same result as both retrieval method depends on only color histogram but fig. 4 shows some different results which consider texture feature also in addition to color feature. The texture in the target image is retained in the retrieved images using texture and color histogram in fig. 4 which are somewhat lost in the results of color histogram matching.

2.5. Custom Matching

In the custom matching technique, CBIR is to be implemented to detect sunsets. Five 3D color histograms are used to extract color features, one 1D histogram is used to extract texture using a canny edge detector and Weighted Euclidean distance is used as a distance metric. The image is retrieved using a dataset containing some sunset images along with other categories of images (Olympus database). The complete dataset link is mentioned in the readme file.

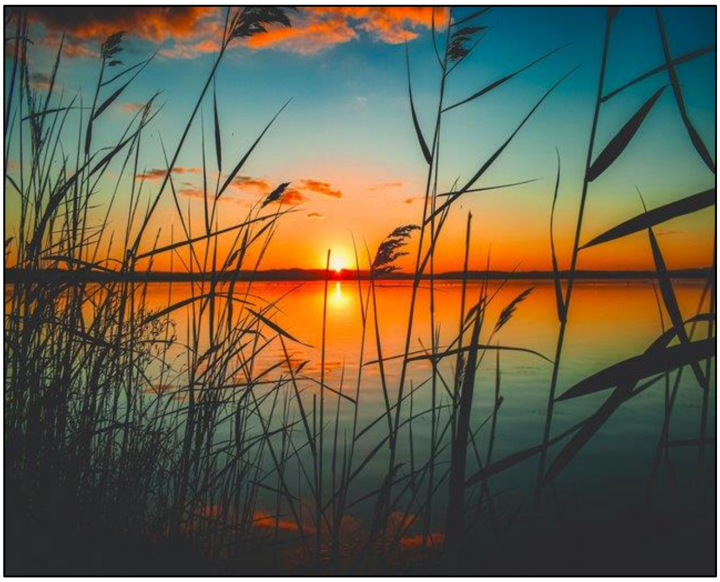
Algorithm adopted:

- The image is divided into 5 parts as shown below to build 5 color histograms and a feature vector is formed using histograms:



- A canny detector is applied to the image to capture the texture features and 1D texture histograms are formed.
- After forming the feature vector, weighted Euclidean distance is used as a distance metric. It is assumed that the chances of the sun being present in region 3 is higher than in Region 2,4 and the least chance to be present in Region 4,5. And thus weight is provided accordingly.
- Since color distribution is more important in sunsets than texture distribution and more weight is provided to color than texture.

The top 10 retrieved images are shown below for 2 target images as shown below:



Target Image 1

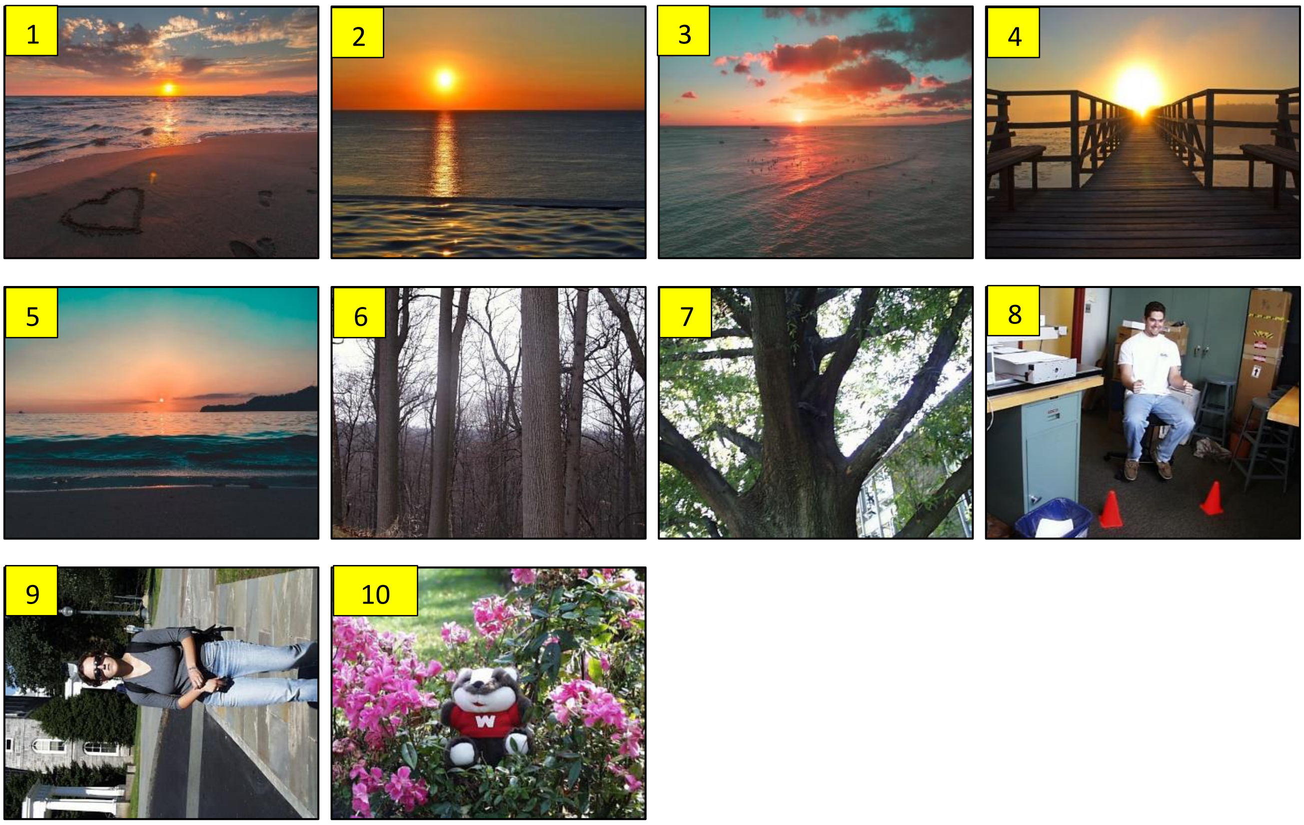


Fig. 7. Top 10 matches using Custom matching for target image 1

Target image 2

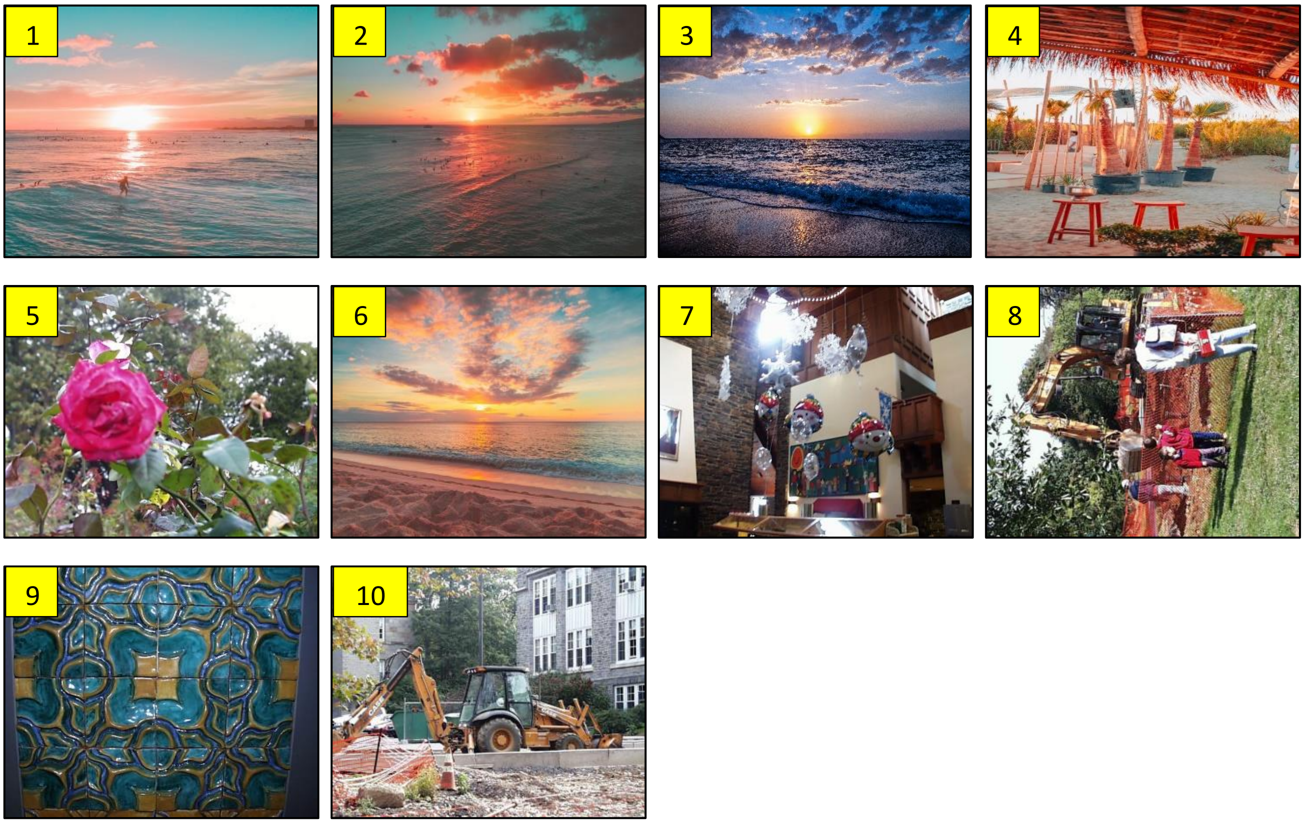


Fig. 8. Top 10 matches using Custom matching for target image 2

The applied algorithm for CBIR performed well and was able to retrieve similar sunset images for 50% of the top 10 images. The result can be improved by incorporating complex texture features and shape feature extraction methods.

2.6. Gabor feature Matching

As an extension, a bunch of Gabor filters is used to extract features [3] and to form the feature vector and Manhattan distance as a distance metric. 6 Gabor filter is applied with different parameters to capture complex textures and the chosen parameters are mentioned below for each filter:

1. Kernel size 5x5, theta pi/6, sigma 2, lambda 0.5
2. Kernel size 11x11, theta pi/3, sigma 2.5, lambda 0.5/2
3. Kernel size 17x17, theta pi/2, sigma 3, lambda 0.5/4
4. Kernel size 23x23, theta pi*2/3, sigma 3.5, lambda 0.5/8
5. Kernel size 29x29, theta pi*5/6, sigma 4, lambda 0.5/16
6. Kernel size 35x35, theta pi, sigma 4.5, lambda 0.5/32

After obtaining the 6 different filtered images, a single image is generated by combining all 6 images using the L2 norm. A feature vector is obtained by storing the texture data in a 1D histogram. The features are compared by implementing the Manhattan distance.

Target Image (pic.0746)	1st match (pic.0747)	2nd match (pic.0157)	3rd match (pic.0755)

Fig. 9. Gabor feature Matching result for target image pic.0746

Fig. 9 shows that the Gabor feature extraction provides a better result than the Sobel magnitude feature extraction. With the help of multiple Gabor filters, the textures are extracted in multiple orientations and can provide much better performance.

2.7. Spatial variance with Color Texture Matching

In this extension, the objective is to detect images with bananas. To achieve the result following algorithm is applied:

1. RGB image is converted to an HSV image and the range of Hue, saturation, and value is fixed to capture only the yellow pixels[2].
2. A 3D color histogram is obtained using only yellow pixels present in the central part of the image.
3. Pixel coordinates of yellow pixels are located and spatial variance in X and Y directions is calculated if the number of pixels present is below a certain limit, then the high variance is set to those images.
4. A canny detector is applied to capture the texture of the central part of the image.
5. Using the color histogram, texture histogram, and spatial variance, a feature vector is formed.
6. Weighted Manhattan distance is used as a distance metric to obtain the final result with the most weight provided to spatial variance and the least to texture feature.

Target Image (pic.0345)	1st match (pic.0344)	2nd match (pic.0467)	3rd match (pic.0242)

Fig. 9. Gabor feature Matching result for target image pic.0746

The method used to identify similar objects to bananas but not able to detect bananas may be due to the presence of bananas in different orientations and the scale of bananas in the image can be different which can give higher or lower spatial variance than the target image.

3. LEARNING

After completing this project, I learned about various CBIR techniques and how different combinations of feature extraction methods and distance metrics provides different results. I have learned to describe the image with color histogram, texture histogram, and Spatial variance.

ACKNOWLEDGEMENT

1. https://doi.org/10.1007/978-981-15-5761-3
2. Thawari, P. B. and Nitin J. Janwe. “CBIR BASED ON COLOR AND TEXTURE.” (2011).
3. T. Barbu, "Content-Based Image Retrieval Using Gabor Filtering," 2009 20th International Workshop on Database and Expert Systems Application, Linz, Austria, 2009, pp. 236-240, doi: 10.1109/DEXA.2009.61.