CS5330: Pattern Recognition and Computer Vision

Project 2: Content-Based Image Retrieval

Name: Krishna Prasath Senthil Kumaran

Email: senthilkumaran.k@northeastern.edu

Aim of the Project:

This project aims at teaching, how to manipulate and analyze images at a pixel level like in project 1.

A database of images called Olympus is provided and using various feature extraction functions we extract them for each image in the image set and store them separately in a ***.csv*** file. We pick a specific target image and apply the feature function to it and compare the distance between the target image and the images in the image data set.

Finally, the top N images that correlate to the features of the target image based on the color, texture, and spatial layout are outputted as the result.

Task 1: Baseline Matching

In task 1, a 9x9 square is used to extract the center of the image and is returned as a feature vector. The sum of squared difference is used as the distance metric.

Two programs are created to that the feature vector of images given in the database is stored inside the ***baseline.csv*** file. Then, in the second program, an image is given as the target image and its features are compared with the extracted features which are stored in a ***.csv*** file, and the N matches are displayed. An addon to this function is that the top N matches are appended into a text file called ***matched_images.txt***. Fig.1 shows the top three matches.

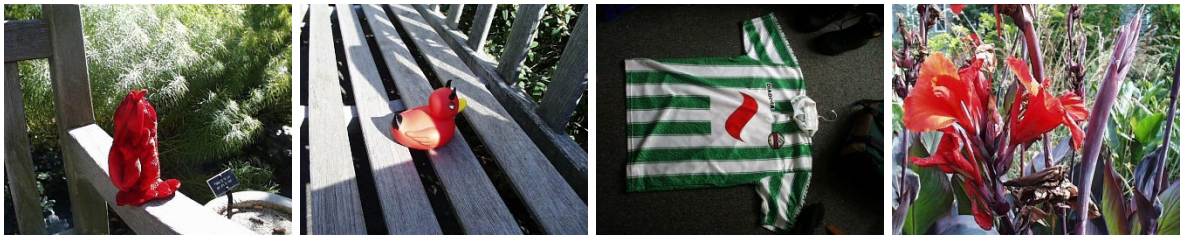

Fig.1 Top 3 matches for baseline matching.

Task 2: Histogram Matching

In task 2, a 2D color histogram is generated as a feature vector. We use histogram intersection as the distance metric to compute the distance between the target image and the image database. Fig.2 illustrates the output of histogram matching.

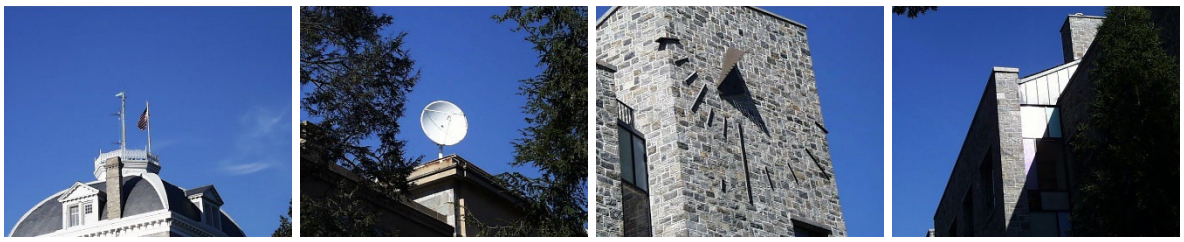

Fig.2 Top 3 matches for histogram matching.

Task 3: Multi-histogram Matching

In task 3, The function is designed to compute a multi-scale histogram of the input image, by dividing the image into four equal-sized regions and computing a histogram of each region. The resulting feature vectors are then concatenated into a single vector and returned as output. Histogram intersection is used as the distance metric for multi-histogram matching. Fig.2 displays the top N matches.

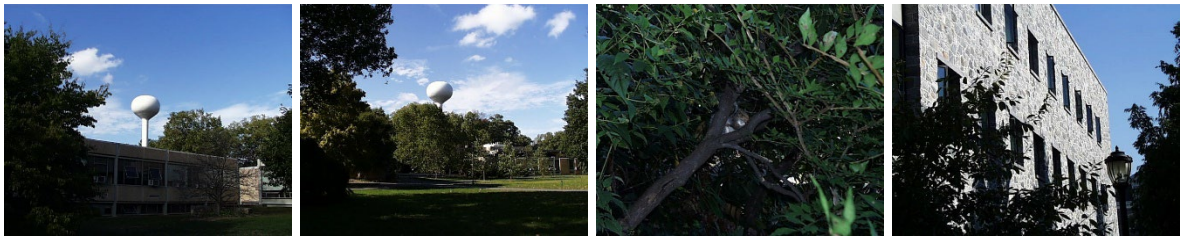

Fig.3 Top 3 matches for multi-histogram matching.

Task 4: Texture and Color

In task 4, The first function, ***"texture"***, computes a texture feature vector from the input color image. It first converts the input image to grayscale, then computes the gradient magnitude and orientation of the grayscale image. It then divides the gradient magnitude and orientation values into 8 bins each and creates a 2D histogram with these bins. The histogram is then normalized and flattened into a 1D vector, which is returned as the texture feature vector.

The second function, ***"textureAndColor",*** computes a combined feature vector from the input color image, by concatenating a texture feature vector and a color feature vector. The color feature vector is obtained by computing a histogram of the color values in the input image. The texture and color feature vectors are concatenated and returned as a single feature vector.

The third function, ***"gaborTexture"***, computes a texture feature vector using Gabor filters at different scales and orientations. It first converts the input image to grayscale and applies a set of Gabor filters with different sigma (scale) and orientation values. For each filter, it computes the mean and standard deviation of the filtered image and appends these values to the output feature vector. The output feature vector is then normalized and returned. Fig.4 illustrates the top N matches of the above functions.

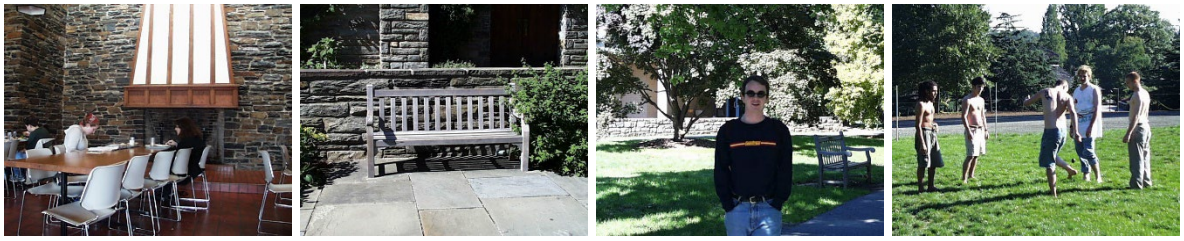

Fig.4 Top 3 matches for texture matching.

Fig.5 Top 3 matches for texture and color matching.

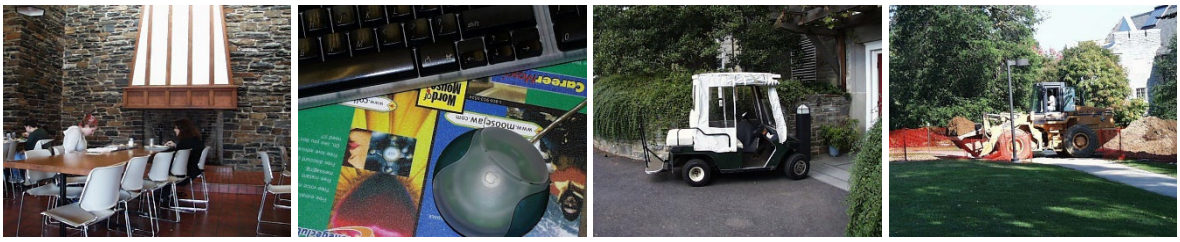

Fig.5 Top 3 matches for gabor filter.

Acknowledgment of resources or people consulted for this project:

- OpenCV Tutorials: https://docs.opencv.org/4.5.1/index.html
- Project 1 referred to the filters used in this project.