Michael McQuade

A01677104

Robot Vision

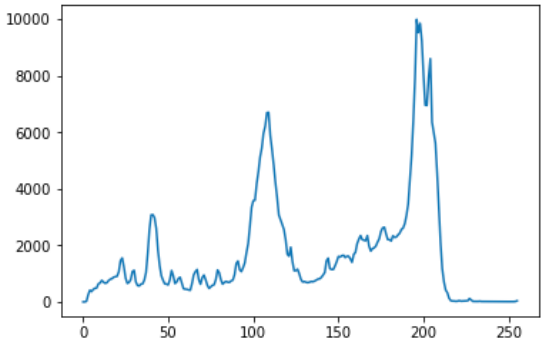
Exam 1

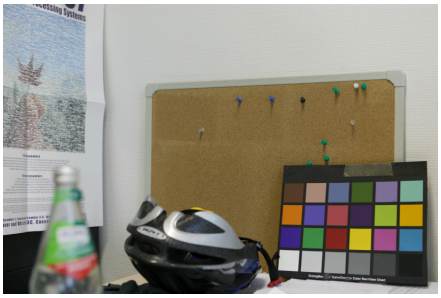
1. In problem 1, I created such a program which allows the user to import an image they specify by path into the program. The program also requests from the user a correlation method with which it will compare the image to a database of images. The program then converts the image to a histogram and runs through the database, comparing each image in the data base to the user's image. It generates a file which is ordered from most similar (highest correlation) to least similar (lowest correlation).
   Pictured is an example output file using one of the pictures in the comparison database as input. This yields a 1.0 score using the correlation method of comparison.

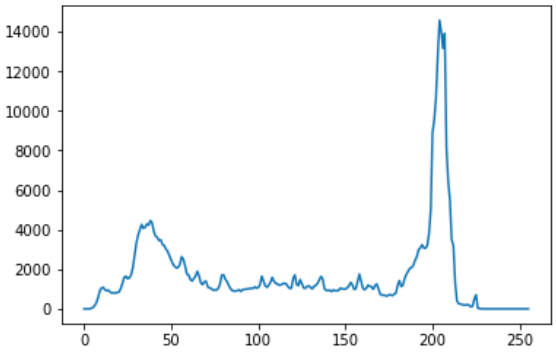
Exam1Output.txt

```
Value: 1.0 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5524.tif
Value: 0.5928380552 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5533.tif
Value: 0.442597888256 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5586.tif
Value: 0.439125447707 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5556.tif
Value: 0.42806002941 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5563.tif
Value: 0.359921878972 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5535.tif
Value: 0.337808938065 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5579.tif
Value: 0.312164911146 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5531.tif
Value: 0.302645444265 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5554.tif
Value: 0.290165805364 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5564.tif
Value: 0.287683106002 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5553.tif
Value: 0.250724834549 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5584.tif
Value: 0.228431223014 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5544.tif
Value: 0.21872355328 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5539.tif
Value: 0.213382527373 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5528.tif
Value: 0.211881987823 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5541.tif
Value: 0.206693607832 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5565.tif
Value: 0.202643440867 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5599.tif
Value: 0.199700106217 File: /Users/michaelmcquade/Desktop/DataBase_1/IMG_0351.tif
Value: 0.192774351343 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5571.tif
Value: 0.191109640372 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5574.tif
Value: 0.186228145599 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5583.tif
Value: 0.180351806424 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5538.tif
Value: 0.177097187965 File: /Users/michaelmcquade/Desktop/DataBase_1/IMG_0372.tif
Value: 0.171884783231 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5611.tif
Value: 0.158990652471 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5552.tif
Value: 0.14216134883 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5582.tif
Value: 0.119467605122 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5577.tif
Value: 0.117085470352 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5588.tif
Value: 0.105767459922 File: /Users/michaelmcquade/Desktop/DataBase_1/8D5U5603.tif
```

   For the same input, here is the picture, its histogram, and the second closest match's picture and histogram, with a score of .59 correlation:






In the third problem, I had a folder of pictures that I needed to utilize in my program in order to determine optical flow. First, I created a class that would allow me to get each frame as I needed it. This class, Frames, provides a method which loads an image from the folder into an array, and returns the array to the caller. The next time it is called it returns the next image array in the folder. It achieves this by being a static method and storing the list of frames and current frame number as static class variables. This class is never instantiated. Later, I convert the image I receive to grayscale for processing in the LK Optical Flow method. I then allow the user to select a region of interest using cv2's selectROI method. This returns four points which help define the rectangle the user selected:


I then take these points and build an array containing an equally-separated list of points that can be used as "features" for the Lukas Kanade optical flow algorithm.

After passing the equally space points from the ROI to the Lukas Kanade method, I receive a new array containing the transformed points. As you can see below:

```
Out[18]: array([[[  241.,    66.]],

                [[  241.,    69.]],

                [[  241.,    72.]],

                ...,
                [[ 1192.,   288.]],

                [[ 1192.,   291.]],

                [[ 1192.,   294.]]], dtype=float32)
```

```
Out[16]: array([[[  232.81585693,    66.14873505]],

                [[  232.99090576,    69.25843048]],

                [[  232.95457458,    72.22951508]],

                ...,
                [[ 1168.65112305,   289.95358276]],

                [[ 1177.89904785,   289.44223022]],

                [[ 1157.08740234,   281.70239258]]], dtype=float32)
```

These points represent the new positions of the points after retrieving the next frame and comparing the positions in the new frame.