Jason Hatfield – 29163434
CSE 473 Programming Assignment #3
Due: 7/24/17

1 Image Based Search Engine

Search Engine Results

Capen results:

Query

```
queries/capen.png
 1. images\hayes4.png : 0.913
 2. images\capen5.png : 1.024
 3. images\hayes5.png : 1.242
 4. images\capen3.png : 1.362
 5. images\crosby1.png : 1.397
 6. images\hayes2.png : 1.459
 7. images\capen2.png : 1.667
 8. images\capen4.png : 1.692
 9. images\stadium2.png : 1.797
 10. images\hayes1.png : 1.843
```

1 2 3 4 5

6 7 8 9 10

Relevant Results: 4
Relevant Top 5 Results: 2
Desired Relevant Results: 5

Search Accuracy: 80% | Top 5 Accuracy: 40%

Crosby results:

Query

```
queries/crosby.png
 1. images\crosby1.png : 0.813
 2. images\hayes4.png : 1.060
 3. images\hayes5.png : 1.080
 4. images\crosby2.png : 1.096
 5. images\hayes2.png : 1.175
 6. images\diefendorf3.png : 1.335
 7. images\capen5.png : 1.419
 8. images\hayes3.png : 1.501
 9. images\capen3.png : 1.570
 10. images\stadium2.png : 1.579
```

1 2 3 4 5

6 7 8 9 10

Relevant Results: 2
Relevant Top 5 Results: 2
Desired Relevant Results: 5

Search Accuracy: 40%| Top 5 Accuracy: 40%

Diefendorf results:

Query

```
queries/diefendorf.png
 1. images\diefendorf5.png : 0.265
 2. images\diefendorf3.png : 0.477
 3. images\hayes2.png : 0.730
 4. images\capen5.png : 0.920
 5. images\crosby3.png : 0.941
 6. images\hayes1.png : 1.061
 7. images\hayes3.png : 1.077
 8. images\diefendorf2.png : 1.279
 9. images\diefendorf4.png : 1.296
 10. images\diefendorf1.png : 1.373
```

1 2 3 4 5

6 7 8 9 10

Relevant Results: 5
Relevant Top 5 Results: 2
Desired Relevant Results: 5

Search Accuracy: 100%| Top 5 Accuracy: 40%

Hayes results:

Query

```
queries/hayes.png
 1. images\hayes1.png : 0.272
 2. images\crosby4.png : 0.694
 3. images\crosby3.png : 0.720
 4. images\hayes2.png : 0.761
 5. images\crosby5.png : 0.982
 6. images\diefendorf2.png : 1.150
 7. images\hayes3.png : 1.177
 8. images\capen5.png : 1.302
 9. images\diefendorf4.png : 1.627
 10. images\diefendorf5.png : 1.709
```

1 2 3 4 5

6 7 8 9 10

Relevant Results: 3
Relevant Top 5 Results: 2
Desired Relevant Results: 5

Search Accuracy: 60%| Top 5 Accuracy: 40%

Stadium results:

Query

```
queries/stadium.png
 1. images\stadium4.png : 0.705
 2. images\stadium2.png : 1.504
 3. images\stadium3.png : 1.703
 4. images\crosby2.png : 1.705
 5. images\crosby1.png : 1.737
 6. images\stadium5.png : 1.792
 7. images\capen3.png : 1.963
 8. images\diefendorf3.png : 2.075
 9. images\diefendorf4.png : 2.088
 10. images\hayes4.png : 2.092
```

1 2 3 4 5

6 7 8 9 10

Relevant Results: 4
Relevant Top 5 Results: 3
Desired Relevant Results: 5

Search Accuracy: 80%| Top 5 Accuracy: 60%

Results Discussion:

The search program performed generally well across the query images, though not all of the results were found in the top five for any of them. My overall search accuracy ended up being 72% given 18 successful results against 25 desired successful results. The accuracy when calculating for top five results is 44% given 11 successful results against 25 desired successful results. The most successful searches featured buildings with distinct patterns or colors, while those featuring very similar design and color returned similar buildings that did not match.

Code Discussion:

The code is very well commented and organized on initial inspection. Necessary functions have been segmented into separate files and organized accordingly. Search is performed by executing search_external.py and passing arguments in the form of the dataset location, the index file, and the query image. The code generates an RGB histogram from the query image, loads the index file, then scours the index file returning the nearest ten matches by comparing the sum of squared differences. The matches are packaged into two montages of five images and displayed on screen.

The image search engine can possibly be improved by altering the search method. By detecting and indexing the sift features of each image, and using that criteria to search for matches, we should be able to increase the accuracy of the search engine. Instead of relying on the RGB distribution, which is prone to inaccuracies when dealing with a set of images with similar color tone, we can detect features and match the images to those. Additionally, the code could allow dynamic sizing of images (size not hardcoded) which would have made operation a bit smoother.

The difficulty in this assignment is in learning to adapt and run python code that we are not familiar with yet. The above suggestions would indeed simplify and improve the search especially given a more straightforward user interface. This could be achieved by implementing a GUI or centralized bit of code to handle user specific input.