Words and Pictures HW 1
(Piyush Kansal and Vinay K)

Question 1

For this question, we resized the image to a 32 X 32 scale, as required, and computed the SSD for each pixel between a query image and a database image. The SSD values were then sorted in increasing order, and the first 10 values were picked to get the 10 closest matches to a given query image.

An output of a trial run is given below.

Query
PATH: contains path of our local folder which contained the test data
imageRetrieval_image('img_bags_clutch_1.jpg', 'PATH/images/', 'PATH/queryimages/');

Query Image

Matched Images

crocs

Question 2

In this question, the lexicon was first computed. The lexicon was generated based on the image descriptor files, that were given.

The lexicon was stored in a hash table, to cut down the insert time and query time, to a very large extent.

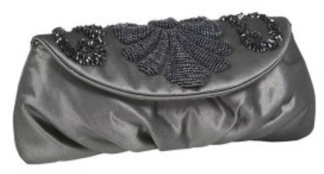
The stop-words were removed from the lexicon using an existing file. We got this file from a NLP project conducted by UC Berkeley, and it is quite exhaustive.

Additionally, any extra whitespaces were also ignored, and added to the lexicon. Puncatuation was removed using the strip_punctuation file. This file was part of the demo code that Prof. Tamara Berg had shared with us.

We also considered using Porter Stemming algorithm to stem the words of the lexicon, however the results were inconsistent such as *absorption* being stemmed to *absoprt*, and *achievement* being stemmed to *achiev*. Hence, our lexicon generation logic does not have any stemming algorithms in it.

Query
PATH: contains path of our local folder which contained the test data
imageRetrieval_text('img_bags_clutch_800.jpg', 'PATH/images/', 'PATH/queryimages/');

Query Image



Output

```
name: 'descr_bags_clutch_106.txt'
date: '11-Sep-2012 00:06:31'
bytes: 326
isdir: 0

name: 'descr_bags_clutch_112.txt'
date: '11-Sep-2012 00:06:31'
bytes: 326
isdir: 0

name: 'descr_bags_clutch_120.txt'
date: '11-Sep-2012 00:06:31'
```

```
bytes: 326
isdir: 0

name: 'descr_bags_clutch_134.txt'
date: '11-Sep-2012 00:06:31'
bytes: 326
isdir: 0

name: 'descr_bags_clutch_142.txt'
date: '11-Sep-2012 00:06:31'
bytes: 326
isdir: 0

name: 'descr_bags_clutch_141.txt'
date: '11-Sep-2012 00:06:31'
bytes: 128
isdir: 0

name: 'descr_bags_clutch_158.txt'
date: '11-Sep-2012 00:06:31'
bytes: 326
isdir: 0

name: 'descr_bags_clutch_172.txt'
date: '11-Sep-2012 00:06:31'
bytes: 326
isdir: 0

name: 'descr_bags_clutch_126.txt'
date: '11-Sep-2012 00:06:31'
bytes: 102
isdir: 0

name: 'descr_bags_clutch_35.txt'
date: '11-Sep-2012 00:06:31'
bytes: 326
isdir: 0
```

Question 3

For this question, we used hash table for to really scale up our algorithm's performance. And it did perform well. We were able to generate the resultant 60 images for 6 different values of alpha in just 3 minutes.

We tried 6 different values of alpha: 0.5, 0.15, 0.25, 0.35, 0.45, 0.65 and we used the same formula which is mentioned over hw1 website. The reason for choosing these values is that text based retrieval generally tends to be more accurate and so higher weightage has been given to it. And due to the same reason, we incremented alpha at a slow pace in the beginning and then in the last, we incremented alpha by 0.20 and took final alpha value as 0.65. As per our understanding, going beyond the value of 0.65 did not make much sense because more weightage was given to image based retrieval.

An output of a trial run is given below.

Query
PATH: contains path of our local folder which contained the test data
imageRetrieval_combined('img_bags_clutch_1.jpg', 'PATH/images/', 'PATH/queryimages/');

Query Image

Matched Images for alpha = 0.5

```
'img_bags_clutch_615.jpg'
'img_bags_clutch_856.jpg'
'img_womens_flats_867.jpg'
'img_womens_flats_897.jpg'
'img_bags_hobo_61.jpg'
'img_womens_pumps_857.jpg'
'img_womens_pumps_841.jpg'
'img_bags_clutch_463.jpg'
'img_womens_flats_51.jpg'
'img_womens_flats_749.jpg'
```

Matched Images for alpha = 0.15

```
        'img_bags_clutch_615.jpg'
        'img_bags_clutch_463.jpg'
        'img_bags_clutch_856.jpg'
        'img_bags_clutch_296.jpg'
        'img_bags_clutch_144.jpg'
        'img_bags_clutch_308.jpg'
        'img_bags_clutch_551.jpg'
        'img_bags_clutch_354.jpg'
        'img_womens_flats_897.jpg'
        'img_womens_flats_867.jpg'
```

Matched Images for alpha = 0.25

```
        'img_bags_clutch_615.jpg'
        'img_bags_clutch_856.jpg'
        'img_bags_clutch_463.jpg'
        'img_womens_flats_897.jpg'
        'img_womens_flats_867.jpg'
        'img_bags_hobo_61.jpg'
        'img_womens_flats_51.jpg'
        'img_bags_clutch_144.jpg'
        'img_bags_clutch_551.jpg'
        'img_bags_clutch_296.jpg'
```

Matched Images for alpha = 0.35

```
        'img_bags_clutch_615.jpg'
        'img_bags_clutch_856.jpg'
        'img_bags_clutch_463.jpg'
        'img_womens_flats_867.jpg'
        'img_womens_flats_897.jpg'
        'img_bags_hobo_61.jpg'
        'img_womens_flats_51.jpg'
        'img_womens_pumps_857.jpg'
        'img_womens_pumps_841.jpg'
        'img_bags_clutch_144.jpg'
```

Matched Images for alpha = 0.45

```
        'img_bags_clutch_615.jpg'
        'img_bags_clutch_856.jpg'
        'img_womens_flats_867.jpg'
        'img_womens_flats_897.jpg'
        'img_bags_hobo_61.jpg'
        'img_bags_clutch_463.jpg'
        'img_womens_pumps_857.jpg'
        'img_womens_pumps_841.jpg'
        'img_womens_flats_51.jpg'
        'img_womens_flats_749.jpg'
```

Matched Images for alpha = 0.65

```
        'img_bags_clutch_615.jpg'
        'img_bags_clutch_856.jpg'
        'img_womens_flats_867.jpg'
```

```
'img_womens_flats_897.jpg'
'img_bags_hobo_61.jpg'
'img_womens_pumps_857.jpg'
'img_womens_pumps_841.jpg'
'img_womens_flats_51.jpg'
'img_womens_flats_749.jpg'
'img_bags_clutch_463.jpg'
```

Best Result: We got the best result with alpha = 0.15