Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Notes:

- I worked with Gerald D’Ascoli on this homework assignment

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Section 01

Q1.1.1

The following four filter functions serve different purposes:

1. Gaussian
2. Laplacian of gaussian
3. Derivative Gaussian in X-Direction
4. Derivative Gaussian in Y-Direction

The Gaussian filter is used similar to a high-pass filter where high frequency information and patterns are removed to the smoothing function of the filter. Low frequency information and patterns are preserved and more prominent in the image afterwards too.

The Laplacian of the Gaussian filter serves to perform the opposite of the Gaussian filter. The Laplacian of the Gaussian retains the high frequency information and patterns.

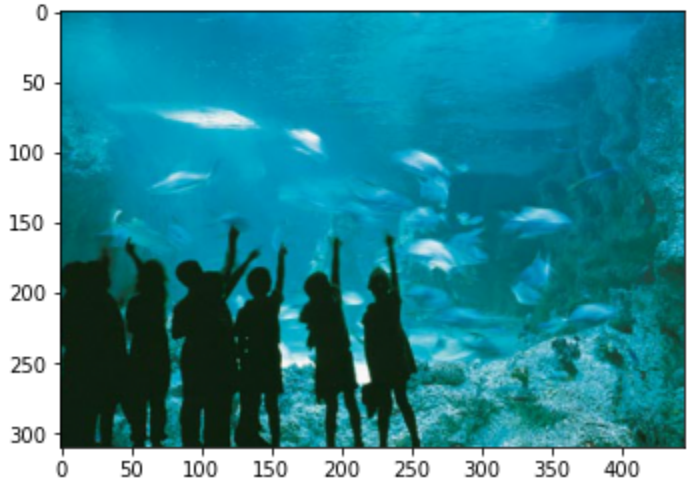
The Derivative Gaussian in the X-Direction is used to highlight vertical lines and features in images. As the x-derivative crosses a vertical feature of high and sudden pixel changes, a high derivative value is experience and preserved in the filtered image. Horizontal features and edges are lost in the output image.

The Derivative Gaussian in the Y-Direction serves to perform the opposite of its counterpart in the x-direction. Used to preserve and highlight horizontal lines and features, a high derivative is experienced in the filtered image as the kernel cross through high and sudden pixel changes in the y-direction. Vertical features and edges are lost in the output image.

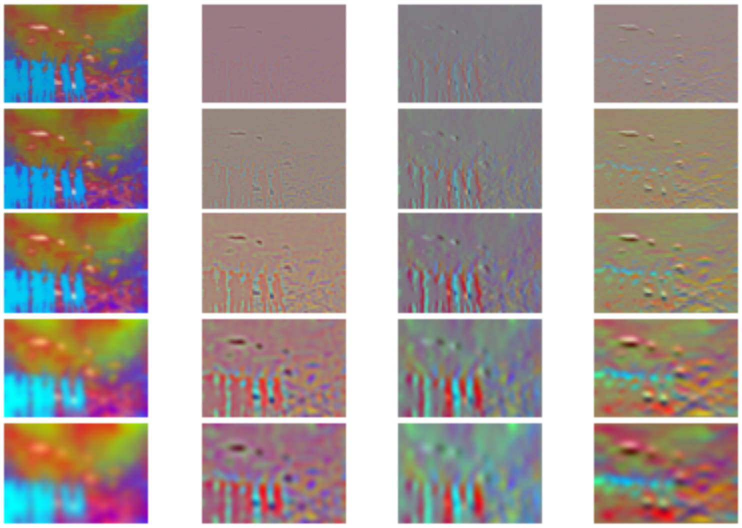
Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q1.1.2

The following image shown below (*aquarium/ sun_aztvjgubyrgvirup.jpg*) is the image used to output the 20-section filter collage:



Shown below is the 20-image filter collage outputted by the *visual_words.extract_filter_responses(image)* function:

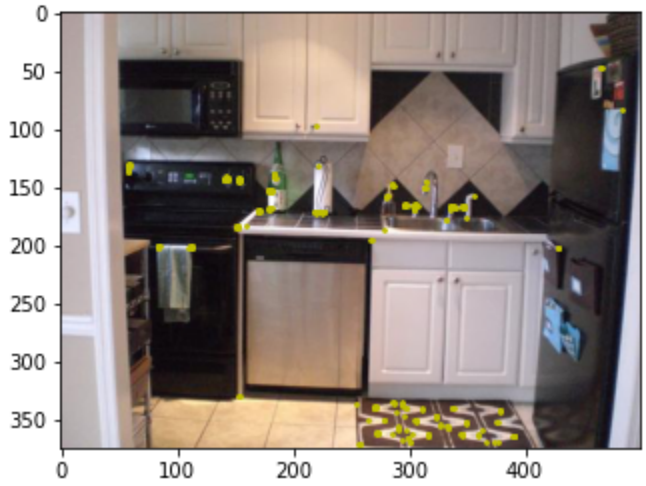


Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q1.2.1

The following kitchen image (*kitchen/sun_aaslbwtcdcwjukuo.jpg*) was used to determine its Harris Corners, which are overlaid in the image and colored in yellow circular dots. Total number of interests was set to 250 while the k-value was set to 0.05.

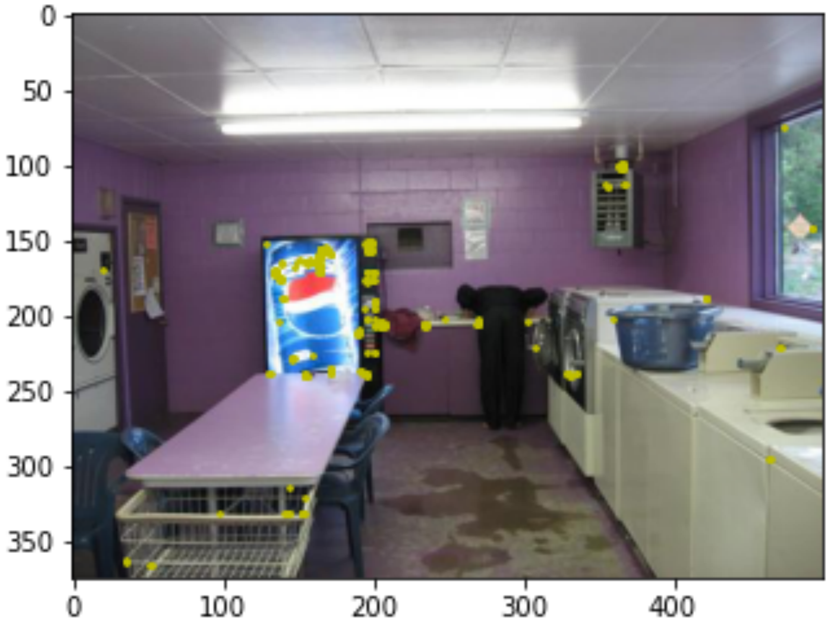
The bulk of the interest points focused on either the corners of the faucet or faucet handles. The dials on the stove-top were also studied for convention. Due to the strong corner detection at these points, and because no NMS was performed on the image prior to perform corner detection, any interest points contain more than one Harris Corner dot. This is observed through the large "blobs" at the interest points as opposed to fine dots.



The following laundromat image (*laundromat/sun_abdvppgytzcuptzn.jpg*) was used to determine its Harris Corners, which are shown over the image in yellow circular dots. Total number of interests was set to 250 while the k-value was set to 0.05.

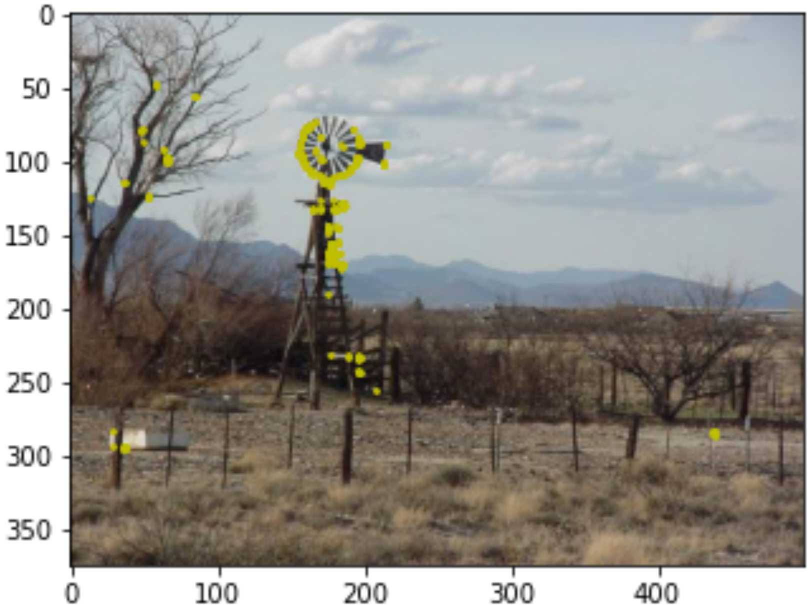
The bulk of the interest points are focused on the details of the vending machine in the background of the image. Several other interest points chosen were the hanging device at the top right corner of the image and several key corners on the washing machines and dryers on the right side of the image.

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up



The following windmill image (windmill/sun_bcceqbhowpvffuxo.jpg) was used to determine its Harris Corners, which are shown over the image in yellow circular dots. Total number of interests was set to 250 while the k-value was set to 0.05.

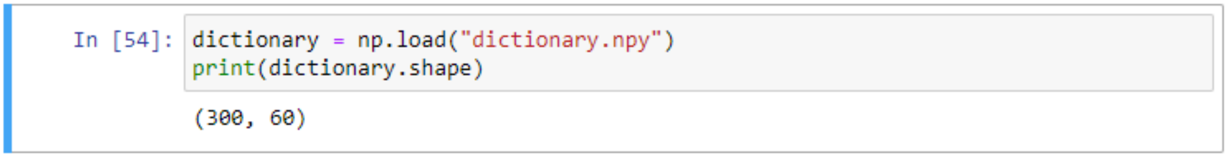
The main areas of interest from the Harris Corner detector were the corners of the tips of the windmill. Several other interest points chosen were the branches of the trees in the foreground and the base of the windmill structure.



Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q1.2.2

The figure below showcases the dictionary object outputted and used for this section. The row element of the shape is the number of KMeans clusters which were set to 300. The column element of the shape is the number of total filters there are multiplied across three image channels (I.e. 20x3=60 channels)

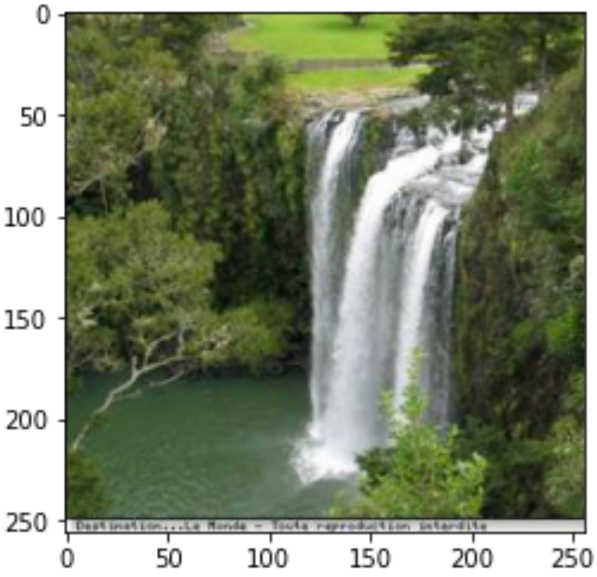


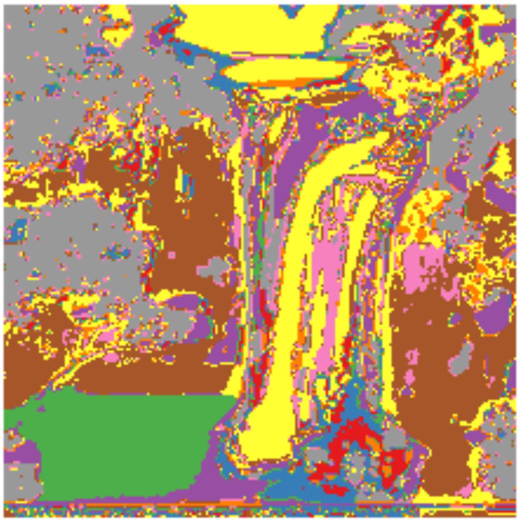
```
In [54]: dictionary = np.load("dictionary.npy")
         print(dictionary.shape)

         (300, 60)
```

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q1.3.1

The figures below show the original and visual word maps of the following image (waterfall/ sun_axetyutyqwxfhnkk.jpg):

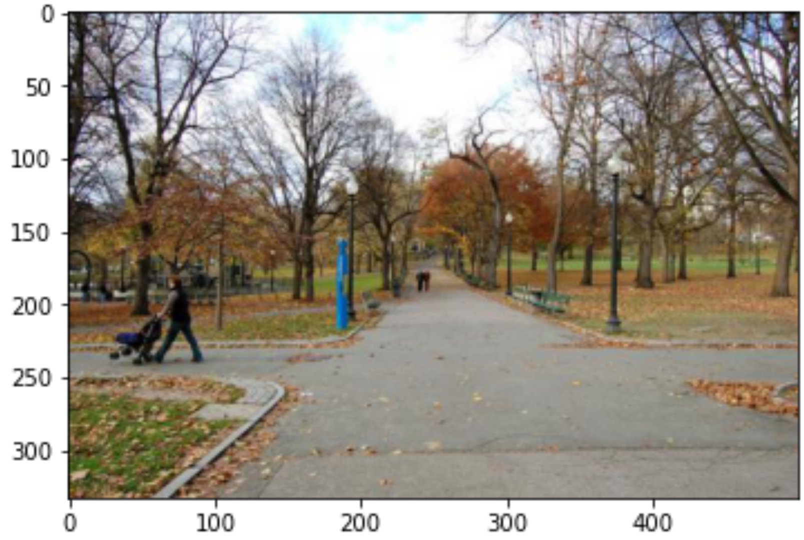


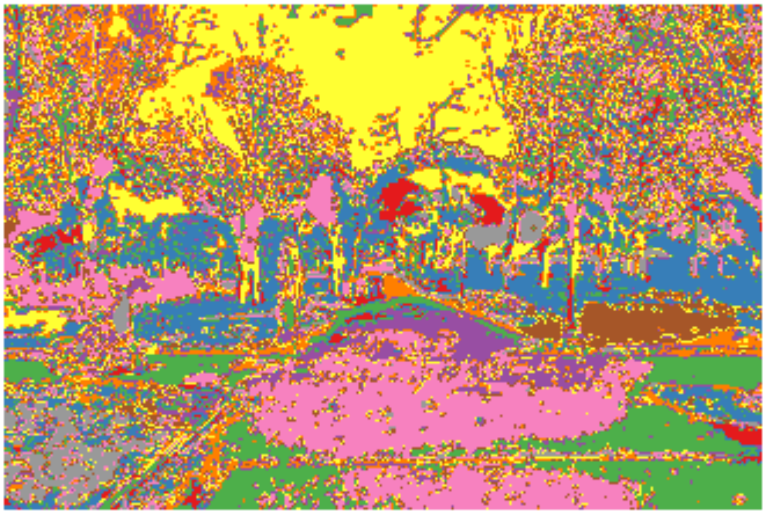


The word boundaries in the visual word map shown above perform well to denote the various boundaries of the features. For example, the trees in the foreground are contrasted with the greenery in the foreground. In addition, the waterfall feature of the image performed well to separate from the body of water below it.

The figures below show the original and visual word maps of the following image (park/ sun_bcjpwnpxcrabfwdw.jpg):

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

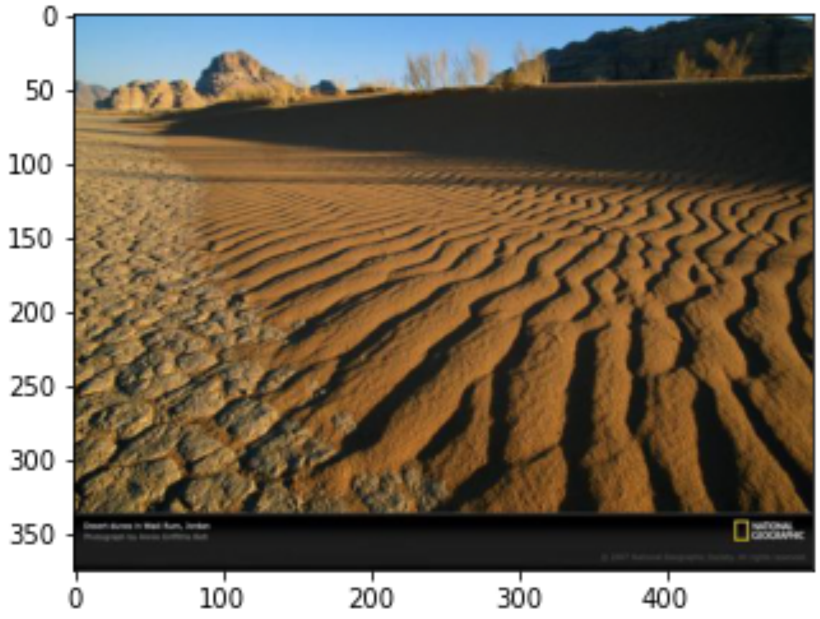


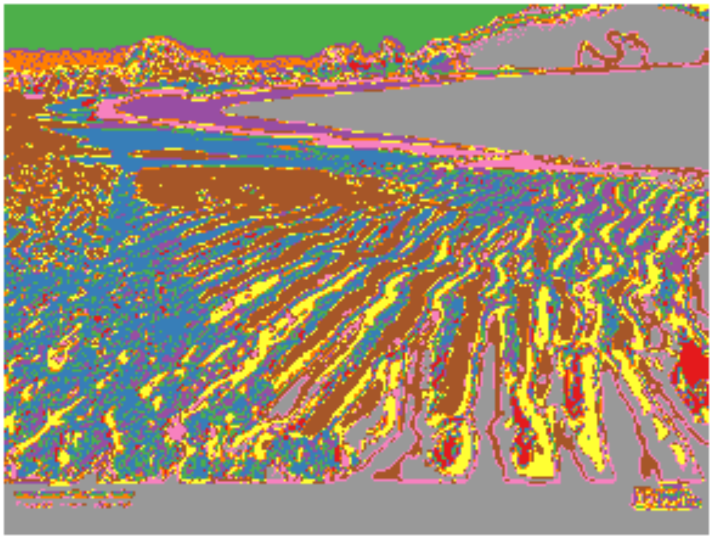


Observing the visual word map above, one can determine that the algorithm perform well to dictate the “word” boundaries that are present in the image. The boundaries of most of the branches are preserved in the visual word map, meaning someone may be able to denote that there exist various branches or leaves on the trees in the image. There are also feature boundaries around the human subject in the image, which is a good indication that the word map is performing well.

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

The figures below show the original and visual word maps of the following image (desert/ sun_biozxhxgtfrmoagh.jpg):





Observing the visual word map, one can see the various “word” boundaries and features of the image. In the foreground, the different contours of the sand and the different types of sand are translated to the word map and separated from each other. The hill in the background and the brush/shrubbery are also contoured in the word map.

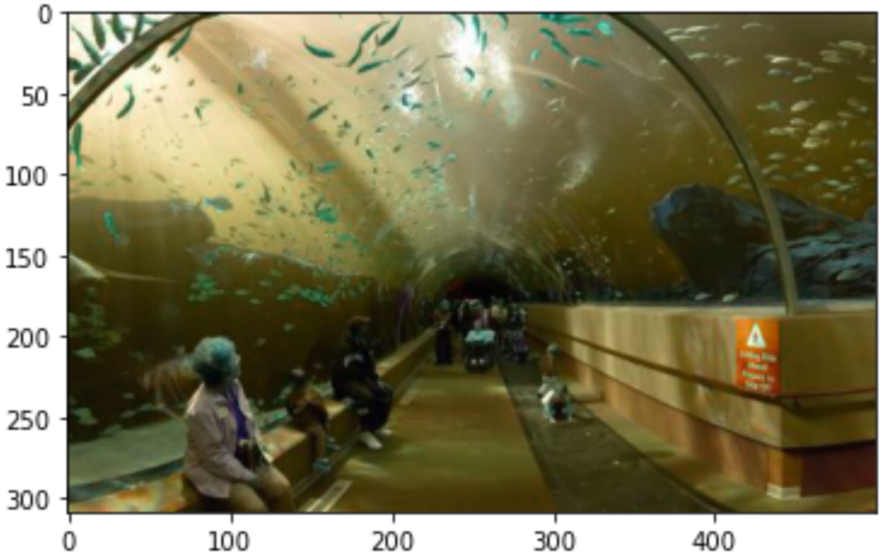
Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

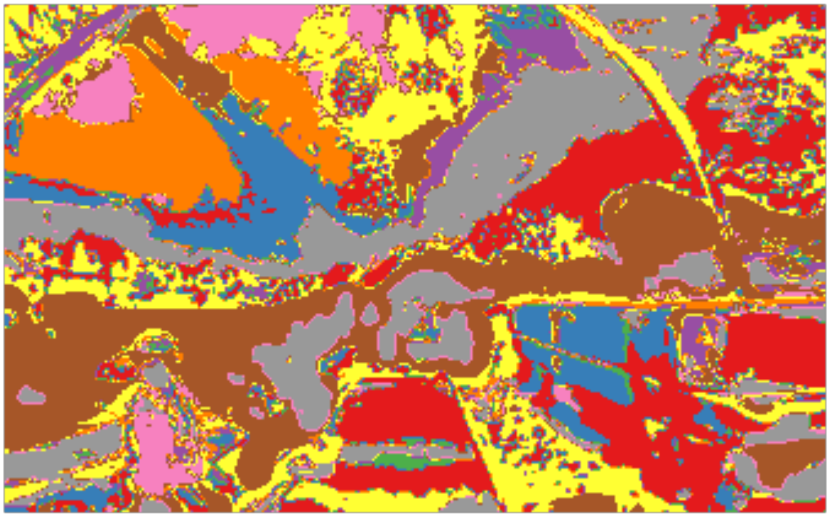
Overall, the “word” boundaries do make sense to me, and it shows that the various distinct features and boundaries of the original image are translated onto the word map which will aid significantly when processing and identifying the images later on.

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

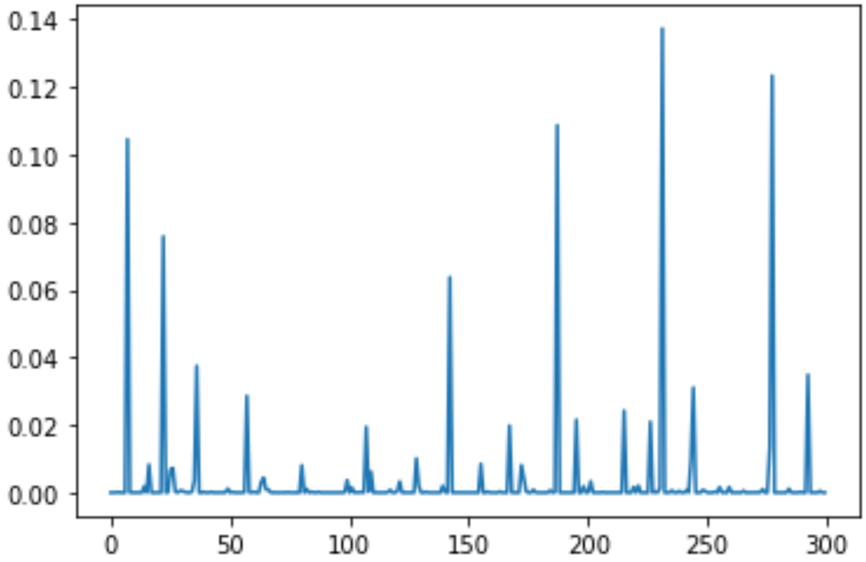
Q2.1.1

The original image, visual image, and histogram figures are shown below for the follow image (aquarium/sun_aairflxfskjrkepm.jpg):





Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

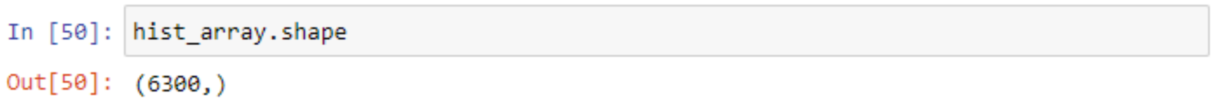


The histogram visualization shown above contains the collection of Euclidean distances from the various KMeans cluster centers, total of 300 cluster centers.

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q2.2.1

With a dictionary size of 300 and a total layer number of 3, the output of the *visual_recog.get_feature_from_wordmap_SPM()* function returns a histogram array of size 6300 as shown below:



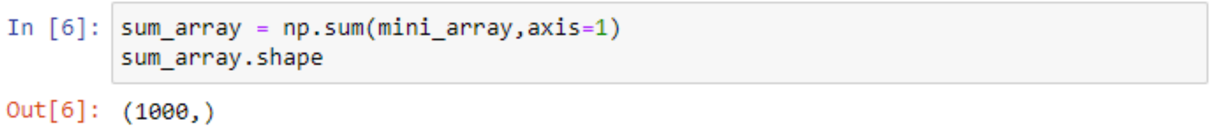
```
In [50]: hist_array.shape

Out[50]: (6300,)
```

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q2.3.1

From the (6300,) SPM histogram output of the test image and the (1000,6300) array of training histograms, the test image histogram is compared with all training images in terms of the minimum between the two. The output is a (1000,6300) array of minimums which is then summed along the column axis to output a (1000,) array of the total of the comparisons. The index of the highest value from this array denotes the most similar image that is compared to the test image.

The image below shows the shape of the *visual_recog.distance_to_set()* function and its output shape:

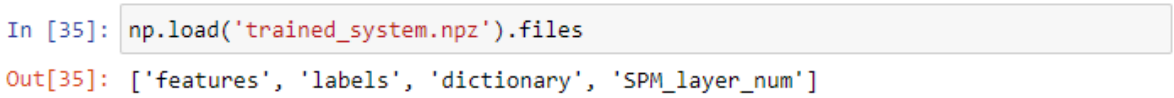


```
In [6]: sum_array = np.sum(mini_array,axis=1)
        sum_array.shape

Out[6]: (1000,)
```

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q2.4.1

The result of the *visual_recog.build_recognition_system()* is a .npz file saved to the current working directory called *trained_system.npz*. When loaded, the .npz file stores the SPM histogram features and labels of all the training images in addition to the previously outputted dictionary and desired SPM_layer_num which is equal to 2:

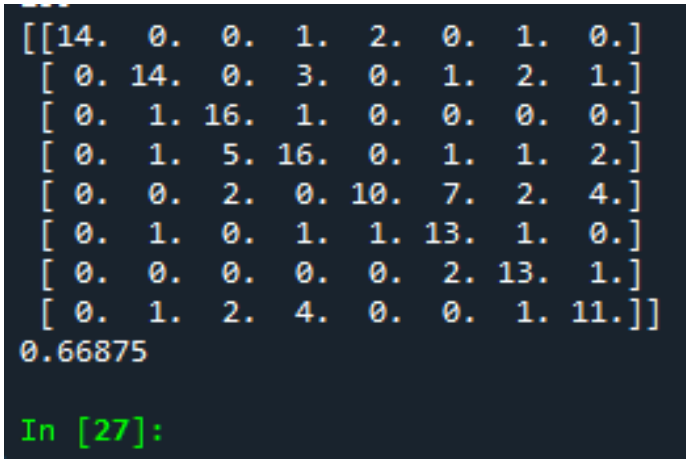


```
In [35]: np.load('trained_system.npz').files

Out[35]: ['features', 'labels', 'dictionary', 'SPM_layer_num']
```

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q3.1.1

After running the *visual_recog.evaluate_recognitionsystem()* function, the following confusion matrix and outputs were outputted, respectively. The rows indices of the confusion matrix denote the predicted label from the algorithm. Whereas the column indices denote the actual true label of the test image.



```
[[14.  0.  0.  1.  2.  0.  1.  0.]
 [ 0. 14.  0.  3.  0.  1.  2.  1.]
 [ 0.  1. 16.  1.  0.  0.  0.  0.]
 [ 0.  1.  5. 16.  0.  1.  1.  2.]
 [ 0.  0.  2.  0. 10.  7.  2.  4.]
 [ 0.  1.  0.  1.  1. 13.  1.  0.]
 [ 0.  0.  0.  0.  0.  2. 13.  1.]
 [ 0.  1.  2.  4.  0.  0.  1. 11.]]
0.66875

In [27]:
```

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q3.1.2

The following are the percent true detections for each label type:

- Aquarium: 14 – 100%
- Park: 18 – 78%
- Desert: 25 – 64%
- Highway: 26 – 62%
- Kitchen: 13 – 77%
- Laundromat: 24 – 54%
- Waterfall: 21 – 62%
- Windmill: 19 – 58%

The labels that exhibited the lowest true detection performance were the laundromats and windmills.

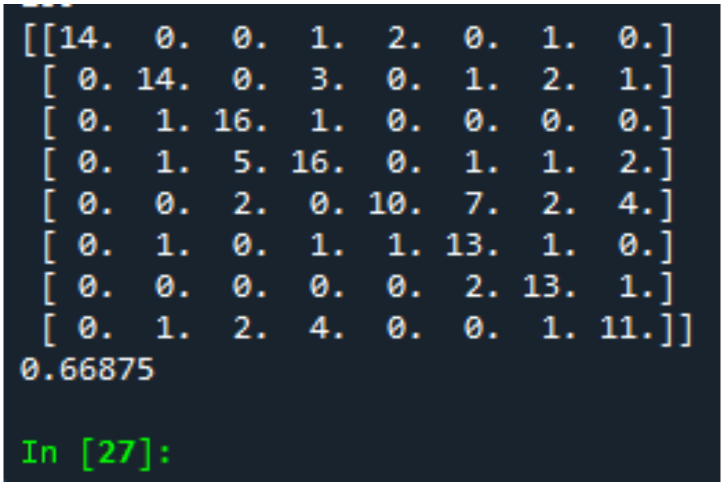
For the laundromats, one can argue that the critical features and interest points of the image set did not translate well and were not distinctive to the KMeans clustering system. Looking at the confusion matrix shown in *Q3.1.1*, most of the false detections of the laundromats were detected as kitchens as opposed to laundromats whereas there have not been many kitchen images that were classified as laundromats. This could mean that the features and interest points of laundromat images are a subset of kitchen image features and interest points whereas kitchen features and interest points are not necessarily equal to laundromat features.

In the case of windmills, there is more of a distribution of false detections. The majority of false detections of windmills are classified as kitchens. Observing the test images, one can see a strong variance between two images of windmills. Differences between windmill images include base structures, windmill blade designs, and nearby objects such as house or trees. With these significant differences, it may be difficult to determine a distinct set of features that would generalize windmills. The result is a stronger distribution of varying classifications, though a majority of classifications were denoted as true in the case of windmills.

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q3.1.3

The figure below showcases an above 65% detection of the bag of words system:



```
[[14.  0.  0.  1.  2.  0.  1.  0.]
 [ 0. 14.  0.  3.  0.  1.  2.  1.]
 [ 0.  1. 16.  1.  0.  0.  0.  0.]
 [ 0.  1.  5. 16.  0.  1.  1.  2.]
 [ 0.  0.  2.  0. 10.  7.  2.  4.]
 [ 0.  1.  0.  1.  1. 13.  1.  0.]
 [ 0.  0.  0.  0.  0.  2. 13.  1.]
 [ 0.  1.  2.  4.  0.  0.  1. 11.]]
0.66875

In [27]:
```

To achieve above 65% performance, the following settings were configured:

- ALPHA = 500 (max recommended by homework assignment)
- KMEANS_CLUSTERS = 300 (max recommended by homework assignment)
- k = 0.05 (default)

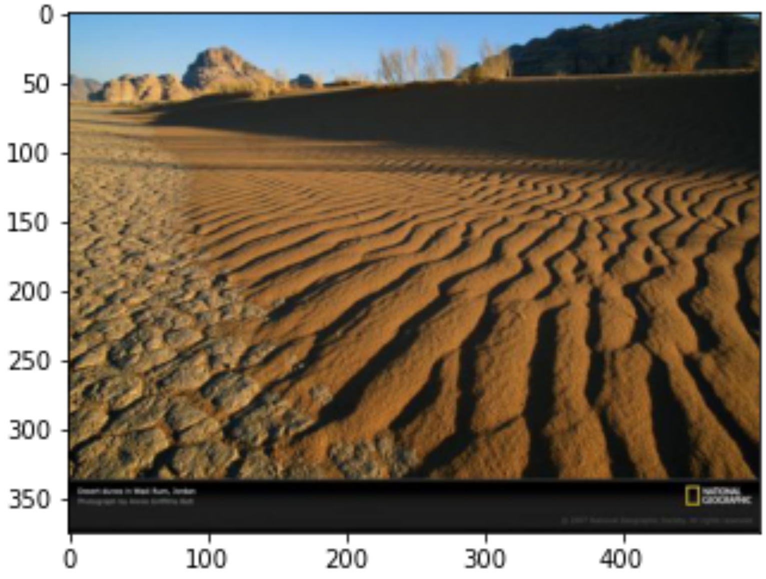
The expectations of configuring the system with these values was that the true classification performance would increase across the board of image types. What actually happened was that some image types experienced an increase in performance and some image types not seeing any or seeing very little true classification performance increase.

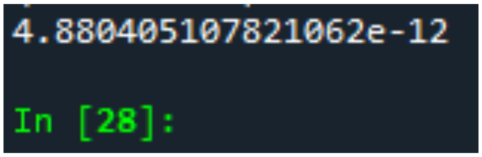
This showed that there is not a direct proportionality between increasing KMean clusters and experience higher true classification performance (same realization for alpha interest points). Improvements in performance vary across image types.

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q4.1.1

The following image shown below (desert/sun_biozxhxgtfrmoagh.jpg) was used to compare the created *network_layers.extract_deep_feature()* function and the pre-trained VGG-16 network model:



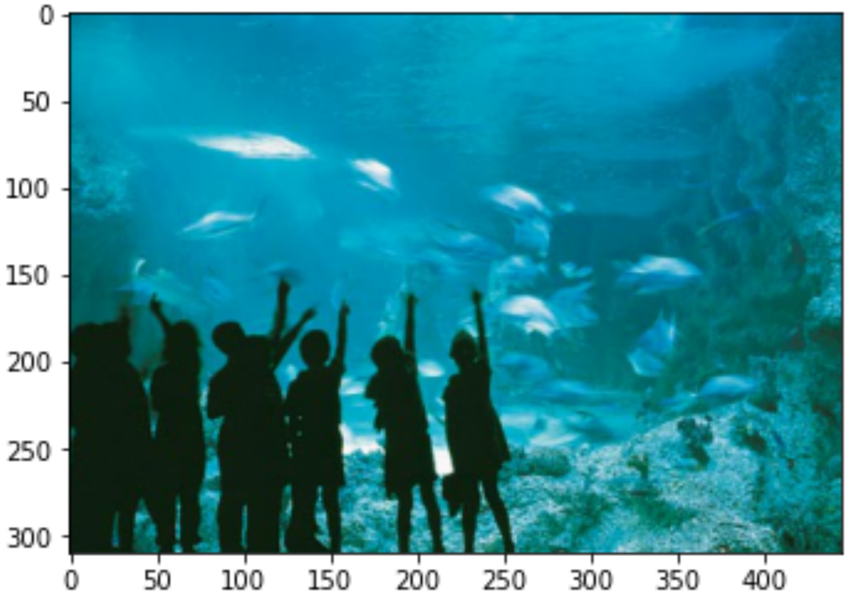


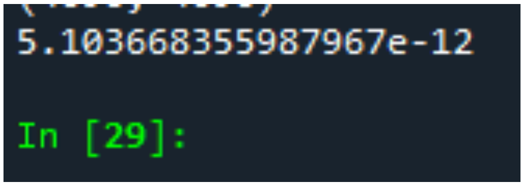
```
4.880405107821062e-12

In [28]:
```

The figure shown above displays the calculated error between the two functions used.

The following image shown below (aquarium/ sun_aztvjgubyrgvirup.jpg) was used to compare the created *network_layers.extract_deep_feature()* function and the pre-trained VGG-16 network model:

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up





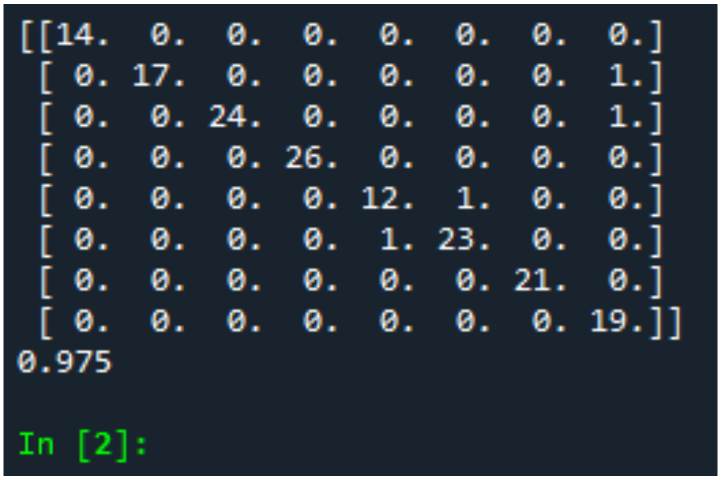
```
5.103668355987967e-12

In [29]:
```

The figure shown above displays the calculated error between the two functions used.

Feng Xiang
16-720B
October 05, 2020
HW #2 Write Up

Q4.2.1

The figure shown below displays the confusion matrix output and calculated accuracy using the pre-trained VGG16 network model.



```
[[14.  0.  0.  0.  0.  0.  0.  0.]
 [ 0. 17.  0.  0.  0.  0.  0.  1.]
 [ 0.  0. 24.  0.  0.  0.  0.  1.]
 [ 0.  0.  0. 26.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. 12.  1.  0.  0.]
 [ 0.  0.  0.  0.  1. 23.  0.  0.]
 [ 0.  0.  0.  0.  0.  0. 21.  0.]
 [ 0.  0.  0.  0.  0.  0.  0. 19.]]
0.975

In [2]:
```

Compared to the performance of the classical BoW model, the performance of the VGG16 network model is better (97.5% vs 66.9%). The significantly higher performance of the VGG16 model over the BoW model can be partly due to the following:

1. Increased number of convolutions to extract key features and patterns in image types
2. Increased layers of image processing with ReLU and max pooling

These two points separate the performance of the VGG16 model over the BoW model as these higher number of layers help to generalize the distinct features between image types and settings. In addition, these higher number of layers help to prevent false negative and false positive detections as generalizations between image types are differ more.