<u>16-720 Computer Vision HW1</u>

Assignment 1
Name : Venkata Sai Gangadhar Nageswar
Andrew ID : vnageswa

Q1.1.1)
All the four filters are used for various purposes.

1. **<u>Gaussian Filter</u>**:
   It blurs the image by applying the Gaussian blurring kernel on the image. As a result, the image becomes more smooth. This is done mainly for two purposes.
   a. To smoothen the image in order to reduce noise, such as salt and pepper noise, in the image.
   b. To blur the image which helps in learning larger features

   Effect of filter scale:
   - The scale determines the amount of blurring. Using larger scales will result in highly blurred images. This can help in the segmentation of images because larger features can be detected

2. **<u>LoG (Laplace of Gaussian) Filter</u>**:
   It applies Gaussian (smoothens first) and then applies Laplace to the image. This detects the edges in the images in all directions. Edges are computed to learn the shape of the features present in the image.

   Effect of filter scale:
   - The scale determines the amount of blurring done before applying the Laplace transform. Using larger scales will result in detecting edges in larger features. If the scale is small, then the edges of smaller features will be detected

3. **<u>DoG (Derivative of Gaussian) in X and Y directions Filter</u>**:
   It applies Gaussian (smoothens first) and then takes the derivative of the image. This detects the features in the images in their respective 'x' and 'y' directions.
   Effect of filter scale:
   - Same as LoG. larger scales will result in detecting edges in larger features.

Q1.1.2)

Parameters Used:
Filters = [1, 3, 7, 10, 20]
Alpha = 25
K = 10

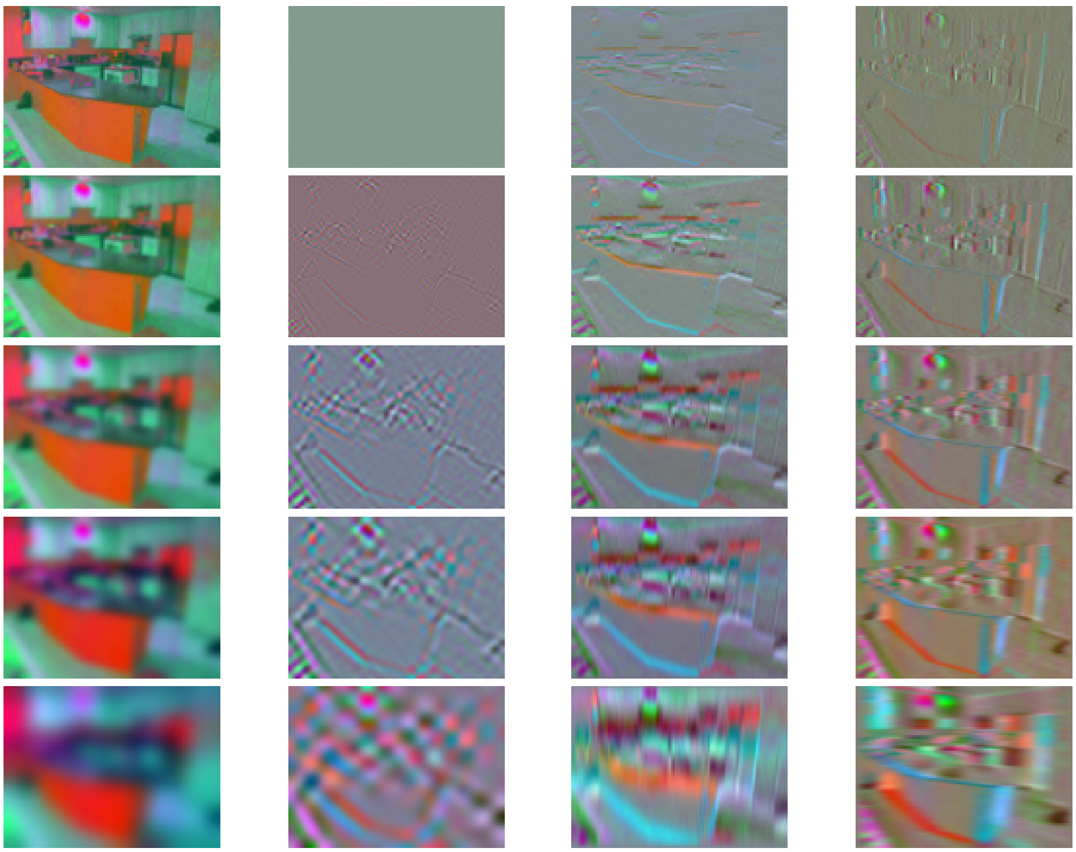

Fig1. Filter Responses of “***kitchen/sun_aasmevtpkslccptd.jpg***”
(from left to right) Gaussian, LoG, DoG_x, DoG_y

Q1.3)

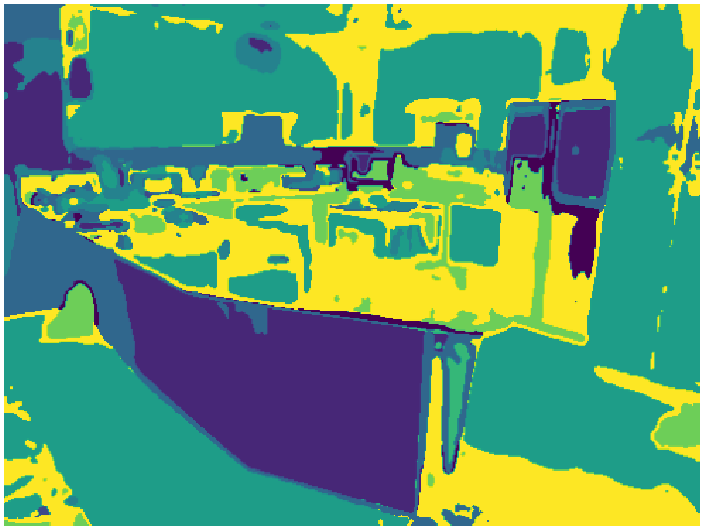

Fig2. Wordmap of ***'kitchen/sun_aasmevtpkslccptd.jpg'***

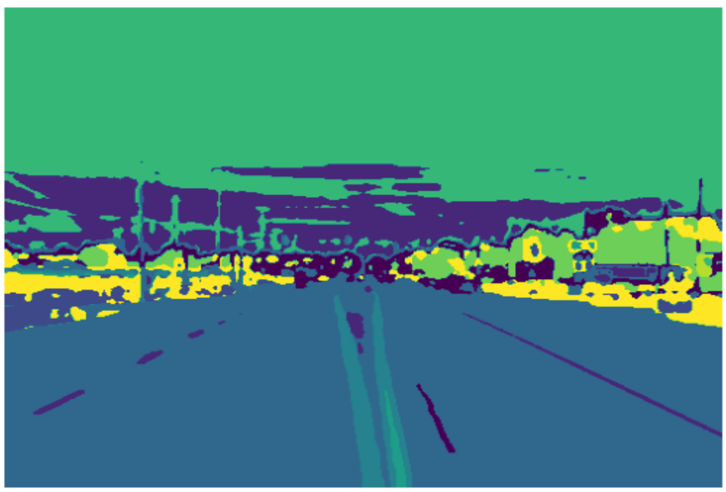

Fig3. Wordmap of ***‘highway/sun_aagkjhignpmigxkv.jpg’***

Fig4. Wordmap of **‘aquarium/sun_aavhedlomnpnwhbf.jpg’**

Q2.5)

Confusion Matrix:

```
[[32.  0.  1.  2.  3.  2.  6.  4.]
 [ 1. 26.  4.  3.  1.  1.  0.  6.]
 [ 2.  4. 33.  3.  0.  4.  1.  4.]
 [ 3.  8.  0. 32. 13.  0.  1.  0.]
 [ 1.  5.  1.  7. 25.  3.  5.  2.]
 [ 3.  0.  1.  1.  3. 31. 12.  5.]
 [ 4.  2.  1.  1.  3.  4. 22.  6.]
 [ 4.  5.  9.  1.  2.  5.  3. 23.]]
```

Accuracy Value: 0.56

Note that ***conf[i][j]*** represents the number of test images that are predicted to be **ith** class but the true label is ‘**j**’

Q2.6)
The confusion matrix obtained using default parameters is as follows:

```
[[32.  0.  1.  2.  3.  2.  6.  4.]
 [ 1. 26.  4.  3.  1.  1.  0.  6.]
 [ 2.  4. 33.  3.  0.  4.  1.  4.]
 [ 3.  8.  0. 32. 13.  0.  1.  0.]
 [ 1.  5.  1.  7. 25.  3.  5.  2.]
 [ 3.  0.  1.  1.  3. 31. 12.  5.]
 [ 4.  2.  1.  1.  3.  4. 22.  6.]
 [ 4.  5.  9.  1.  2.  5.  3. 23.]]
```

Observe that the values [3][4] and [4][3] are high, implying that the model fails to distinguish accurately between the classes ‘Kitchen’ and ‘Laundromat’. Also, [5][6] is high, indicating that waterfall images are being misclassified as parks.

Laundromat kitchen

Observe that the setting in both of these scenarios is where similar and that is the reason the model is confused between the two.

The water present in the waterfall images leads to misclassification as a park.

Q3.1)

Filters	Alpha	K	L	Metric (wordmap)	Test Accuracy
[1,2]	25	10	1	Euclidean	56
[1,3,7,10,20]	25	10	2	Euclidean	56
[1,3,7,10,20]	50	50	3	Euclidean	63.5
[1,2,3,4,5]	50	50	3	Euclidean	63.75
[1,2,3,4,5]	125	50	3	Euclidean	67

Final Achieved Accuracy: 67%

My intuition for changing the parameters is as follows:

Alpha:

Alpha signifies the points we choose for generating the features. Hence using a higher alpha should improve the performance since now more features will be generated for each image. More features should result in better recognition. This behavior can be actually seen in the above table.

L (number of layers = L+1):

By increasing the number of layers, the histograms in a small region will also be computed. This gives an estimate of the finer features present in the image thus improving accuracy. However, note that increasing the layers too much can lead to reduced generalizability, i.e. when the SPM_histogram_features are coming from an image that has a somewhat different distribution than the trained distribution, the model may fail to classify correctly. Reducing L leads to more generalization however at the cost of underfitting.

Number of Words (K):

Using a higher K means that more features in the image will be recognized. Using more words will lead to a better description of the image and hence will improve the accuracy in classifying the images. However, it can also lead to overfitting, i.e. let's say there are people present in the classes “aquarium”, “ highway” and “park”. Now recognition of this feature can lead to misclassification among these classes. Therefore, a higher K will overfit the training data and a lower K will underfit, however, it will be more generalizable.

Filters:

Using larger filters will result in more generalisability of the model. This can be a potential cause for misclassification of kitchen and laundromat because both classes consist of very similar features. Using lower filters can extract smaller details too, which can be useful in scenarios such as kitchens & laundromats, and waterfalls and parks.

Acknowledgements:
For implementing the features for SPM, the idea of recursion has been developed from Kavin Kailash Ravie. However, the code has been entirely written by myself.