Q1.1.1

Gaussian filter - This removes noise from the image and blurs it. It is a low pass filter which removes the high frequency components from the image. It detects broader features.

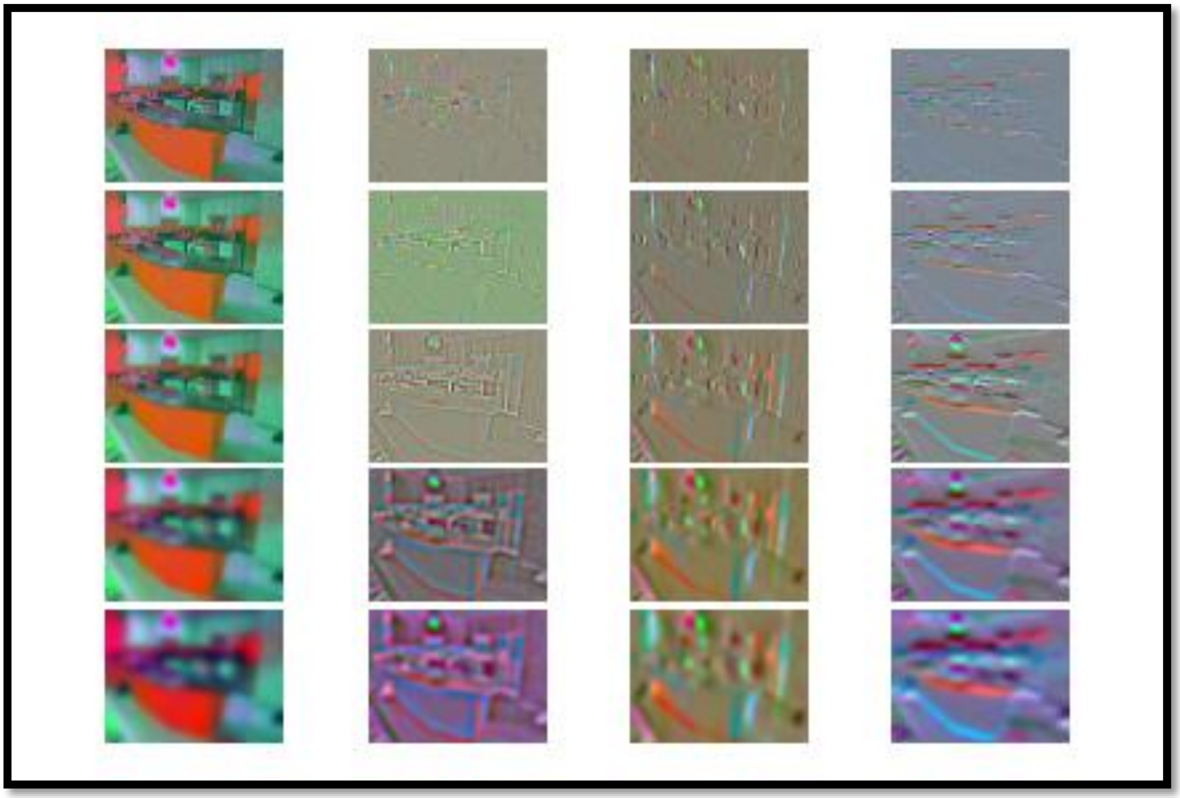
Laplacian of Gaussian - It is used to detect horizontal and vertical edges (shapes) in the image by determining rapid intensity change areas in the image. Since the derivative filter is very sensitive to intensity changes, a Gaussian is used to remove the noise and smooth it.

Derivative of Gaussian in x direction - It is used to detect vertical edges or changes in intensity in the x direction of the image.

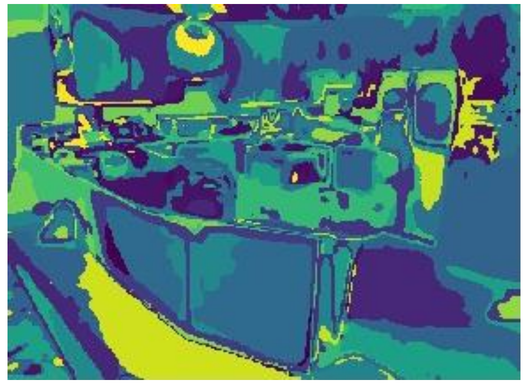
Derivative of Gaussian in y direction - It is used to detect horizontal edges or changes in intensity in the y direction of the image.

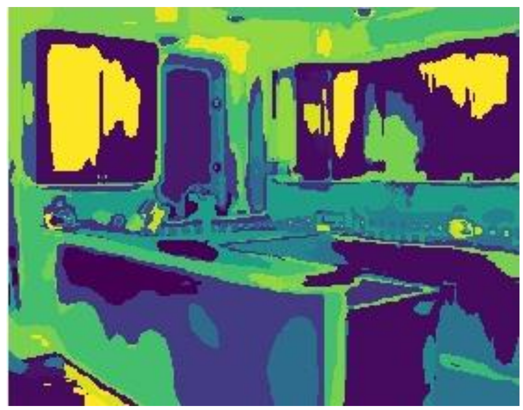
Different scales of filters correspond to different aspects of feature detection. Small scale filters account for features with high intensity changes but for broader feature detection, large scale filters are needed.

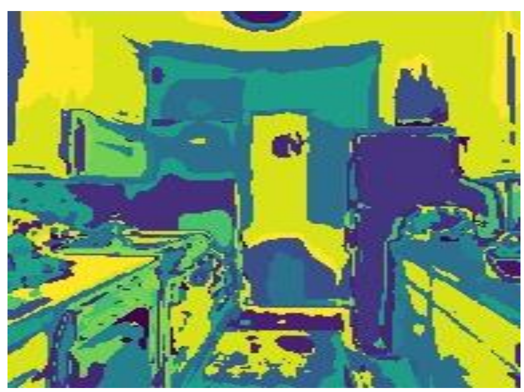
Q1.1.2



Q1.3

The word boundaries are effective at segregating different fixtures in the kitchen. They help distinguish between color changes, variation in intensity and thus are efficient in discerning between cabinets of various surface textures and shades, appliances like refrigerators, ovens and stoves as well as floors and windows in the three images.

Q2.5 alpha = 250 k =200

Confusion matrix is printed below with an accuracy of 53.726 percent

```
[[28.  2.  7.  4.  1.  6.  7.  6.]
 [ 1. 46.  3.  3.  2.  5.  5.  2.]
 [ 4.  3. 57. 10.  5.  2.  1.  5.]
 [ 0.  2.  7. 38.  0.  8.  3. 14.]
 [ 5.  1.  5.  4. 32. 10.  2.  1.]
 [ 6. 11.  2.  6. 17. 37.  0.  2.]
 [ 5. 11.  9.  2.  6.  9. 35.  6.]
 [ 2.  4.  6. 10.  1.  6.  0. 37.]]
53.7261698440208
```

Q2.6

The highway is often misclassified as the dessert (10 times). This is possible due to the fact that they have similar features and long winding pathways approaching the horizon that resemble each other. They also tend to have similar intensities as both are set outdoors. Another set of classes that is wrongly classified is the kitchen and the laundromat. The machines in the laundromat have the tendency to resemble cabinets and appliances present in the kitchen. The program is unable to discern these finer differences with the given set of parameters and they end up being incorrectly classified.

3. Extra credit

Attempts were made to improve the classifier by changing values of alpha and k which is cluster size. Theoretically speaking alpha must be increased to improved accuracy as must the number of clusters. But a tradeoff needs to be attained when choosing the value of cluster size as too large a dictionary might overfit the features of each image whereas too small a size may not capture the necessary features causing a case of underfitting.

Alpha = 200, K= 75

```
[[36.  1.  3.  4.  2.  6.  4.  5.]
 [ 3. 44.  1.  3.  3.  3.  8.  2.]
 [ 0.  2. 58.  7.  8.  3.  0.  9.]
 [ 0. 10.  6. 43.  1.  1.  1. 10.]
 [ 7.  2.  1.  1. 31. 16.  1.  1.]
 [ 1. 15.  1.  4. 18. 37.  4.  1.]
 [ 1. 17. 10.  3.  4.  9. 35.  4.]
 [ 1.  8.  7. 11.  4.  2.  2. 31.]]
54.59272097053726
```

Alpha = 250 K = 75

```
[[35.  2.  8.  6.  2.  4.  2.  2.]
 [ 1. 46.  1.  3.  4.  3.  5.  4.]
 [ 3.  3. 56. 14.  1.  4.  4.  2.]
 [ 1.  2. 10. 41.  1.  6.  3.  8.]
 [ 8.  3.  3.  1. 28. 12.  4.  1.]
 [ 6. 12.  4.  5. 18. 32.  3.  1.]
 [ 7.  7.  7.  5.  7. 14. 29.  7.]
 [ 3.  3.  7. 13.  2.  2.  4. 32.]]
51.81975736568457
```

Alpha = 250, k = 200

```
[[28.  2.  7.  4.  1.  6.  7.  6.]
 [ 1. 46.  3.  3.  2.  5.  5.  2.]
 [ 4.  3. 57. 10.  5.  2.  1.  5.]
 [ 0.  2.  7. 38.  0.  8.  3. 14.]
 [ 5.  1.  5.  4. 32. 10.  2.  1.]
 [ 6. 11.  2.  6. 17. 37.  0.  2.]
 [ 5. 11.  9.  2.  6.  9. 35.  6.]
 [ 2.  4.  6. 10.  1.  6.  0. 37.]]
53.7261698440208
```

Going out on a hunch, I decided to replace the kmeans approach for creating the dictionary with **gaussian mixture models** computed through **expectation maximization**. Theoretically speaking the cluster computed by k means would be spherical in nature which could not necessarily represent the feature distribution. However Gaussian mixture models would allow for ellipses or similar shapes with different covariances thus allowing for higher accuracy. It took a very long time to compute. I have included this function in custom.py.

I also attempted to **resize the images** using the method proposed by Torralba, Fergus and Freeman [1]. I resized the image to a small resolution image (16 by 16). With zero mean and unit length. High frequency content is neglected and it gives fairly poor accuracy. Though this might work better on larger datasets with greater variation, it fails in this particular case. I have also change the distance metric of comparison between the 2 histograms to account for **euclidean, cosine and jaccard distance**. To account for each of these, just change the metric to 'cosine' etc in line 74 of visual words.py. With the cosine metric the accuracy increased upto 56 percent with alpha 250 and clusters 150

```
[[49.  3.  1.  2.  1.  2.  2.  1.]
 [ 2. 50.  2.  1.  2.  2.  5.  3.]
 [ 2.  4. 50.  5.  8.  5.  3. 10.]
 [ 2.  9.  7. 34.  2.  1.  5. 12.]
 [ 7.  2.  2.  0. 42.  3.  1.  3.]
 [ 6. 12.  1.  3. 25. 29.  5.  0.]
 [ 8. 17.  4.  3.  7.  2. 38.  4.]
 [ 5.  7.  7. 10.  2.  0.  3. 32.]]
56.1525129982669
```

References:

Torralba, Antonio, Rob Fergus, and William T. Freeman. "80 million tiny images: A large data set for nonparametric object and scene recognition." *IEEE transactions on pattern analysis and machine intelligence* 30.11 (2008): 1958-1970.