Report for HW1

Q1.0

The first one is Gaussian. The low-frequency signal can be filtered from the whole image by it, putting away high-frequency signal, which could be noise or in a way work as noise for our process.

The second one is Laplacian of the Gaussian. It picks up edges of images.

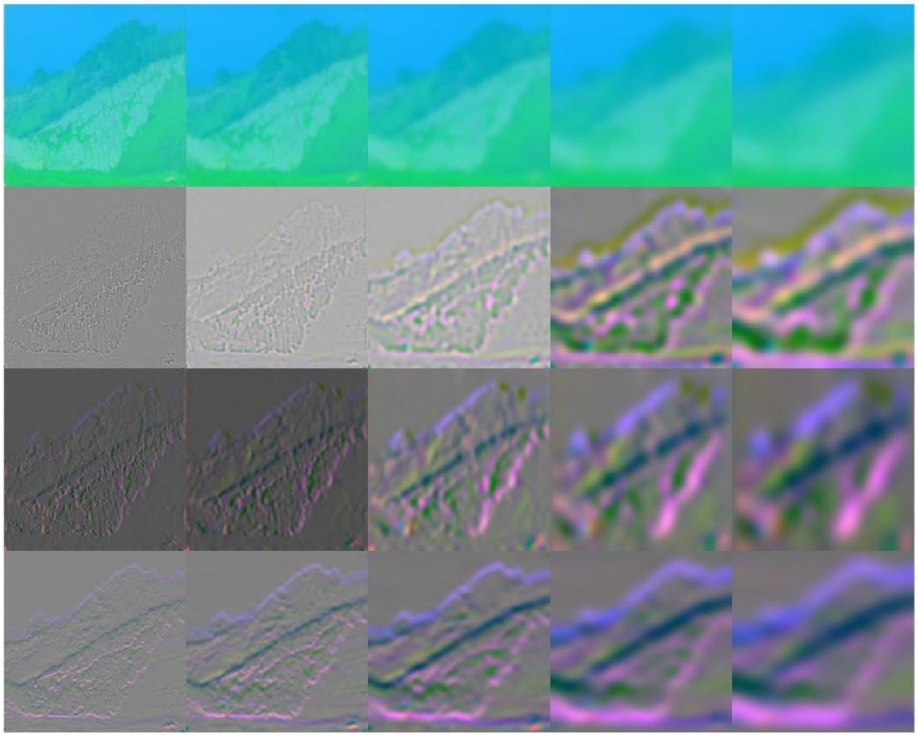
The third one is Linear of the Gaussian. It picks up vertical edges of images.

The third one is also Linear of the Gaussian. It picks up horizontal edges of images.

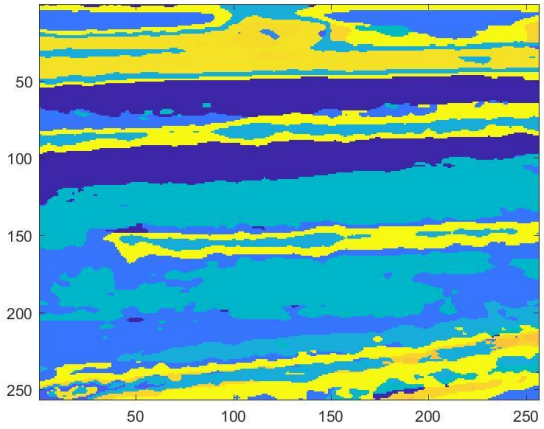
Q1.1

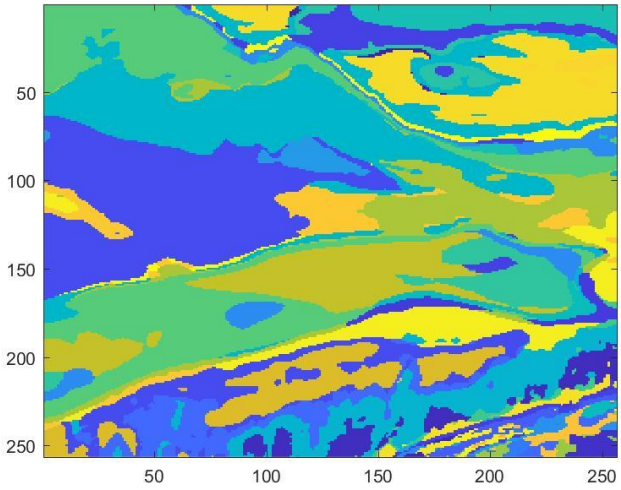
The picture below is for easier observation. The little parts are based on different color map.

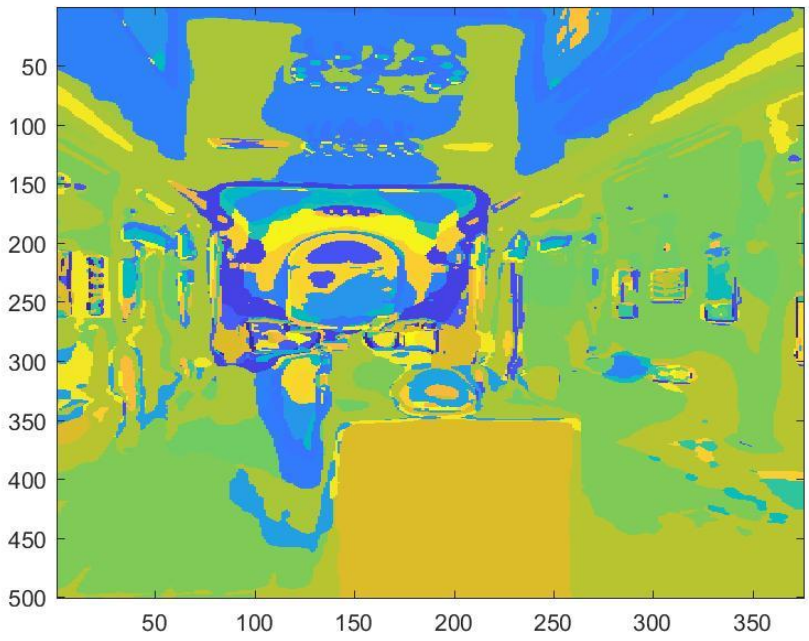
The picture on top is the answer for this question.

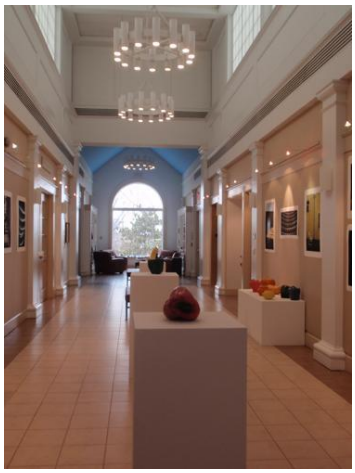


Q1.3




50
100
150
200
250
50
100
150
200
250


50
100
150
200
250
300
350
400
450
500
50
100
150
200
250
300
350



Imagesc() changed sizes of pictures, but still it's obvious that wordmaps match original images. Different colors represent points with different features.

Q2.5

Confuse Matrix

Label	1	2	3	4	5	6	7	8
1	12	2			6			
2	1	14		2	3			
3			20					
4		1		16	3			
5	3	1			16			
6			2		1	17		
7						3	17	
8	2	1	1	4				12

Accuracy=77.5%

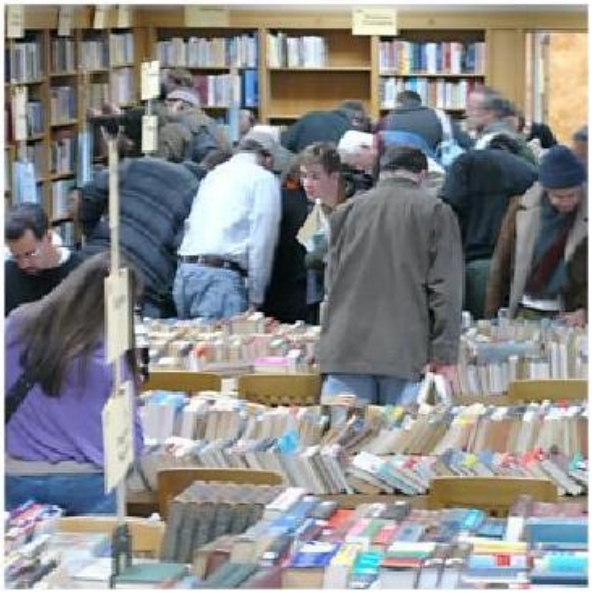
Q2.6

I think there are several causes for wrong prediction.

First, the pictures of indoor scenario are just similar, sharing similar objects like windows, lights and walls. And some things have same shape like shelves and frames both have many squares. Though there are features like computers, books that we can use to differentiate, we yet can't recognize those features as certain object but just gather information of edges and colors.

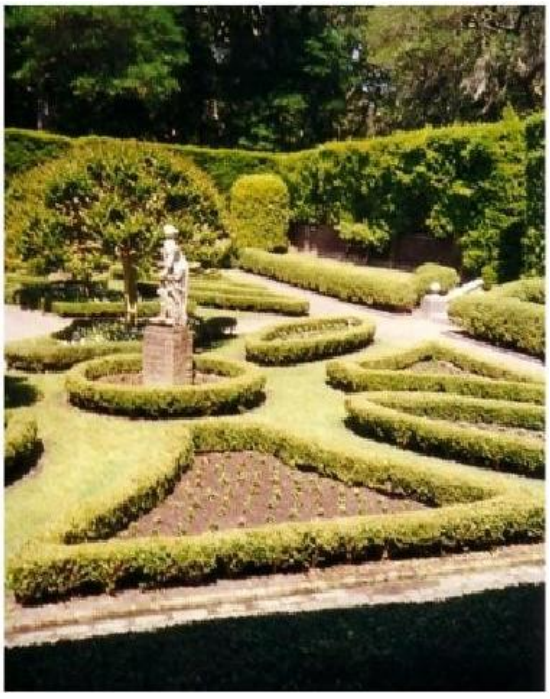
e.g. a picture of library was matched to a picture of art gallery

Second, some extra objects, mainly people, interference prediction.
e.g. a picture of ice skating was matched to a picture of library



Third, some pictures share same structures.

e.g. a picture of tennis court was matched to a picture of garden



Q2.7

There are two ways I tried that successfully boosted the accuracy of the system.

The first one is to adjust weight of each filters. I do this because I found that the first filter take too much weight and the others have weak affection.

The second one is to use several, instead of only one, train pictures that are closest to a test picture to predict. I do this because I think this can avoid some extreme cases by weaken power of certain one train picture that is really similar to a test picture, which have different labels.

The effect of these two ways affect each other, so individual effect isn't clear. But I can say they both work and together boost accuracy by about 20%(from about 55% to about 75%), and this can be improved by adjusting their parameters, which isn't easy due to long running time.