

Q1.1 Show an image from the data set and 3 of its filter responses. Explain any artifacts you may notice. Also briefly describe the CIE Lab color space, and why we would like to use it. We did not cover the CIE Lab color space in class, so you will need to look it up online.

"In this project, we use the CIELab color, CIELab color space refer to three color channels, The L(lightness) channel is ranging from 0 to 100, a (green to red) and b (blue to yellow) channels are ranging from -128 to 127. This color space is more comfortable for human vision, at the same time, the color space would be broader than RGB space."(<https://danielniu.wordpress.com/2018/06/07/bag-of-words/>)

The color space is border so the different dimension of color vector of 1 pixel may contain more meaning.

The follow images are normalized to be visible

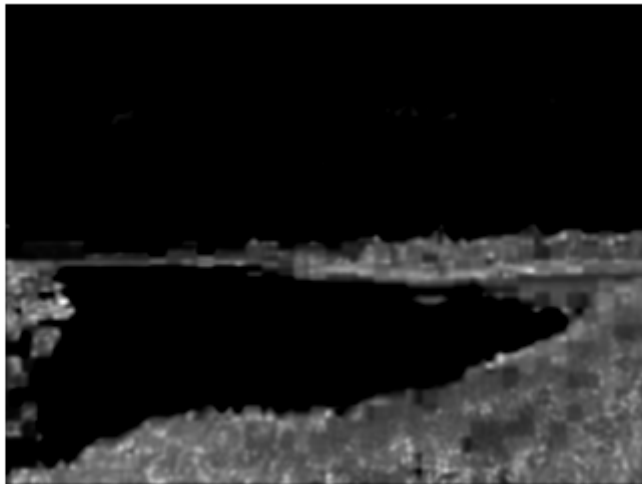
gaussian of L space, looks normal



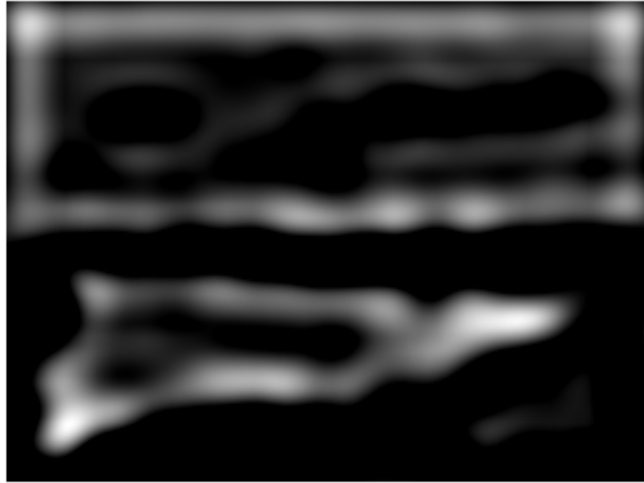
gaussian of a space, looks like some grid, but a apce looks like grid as well



gaussian of b space, can only seen the grass which are green (blue to yellow)



dx of L space

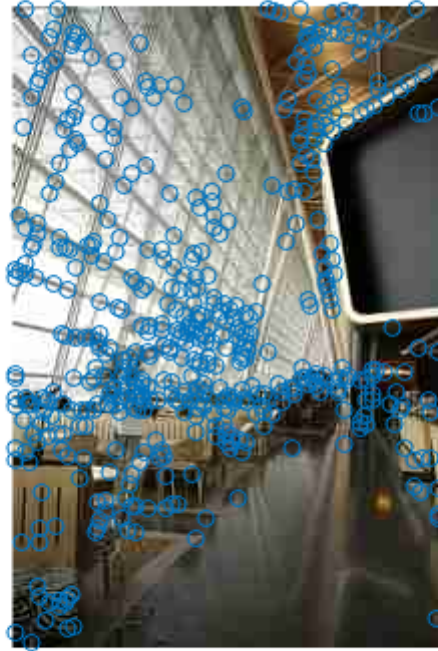


Q1.2 Show the results of your corner detector on 3 different images

500 points are shown here.

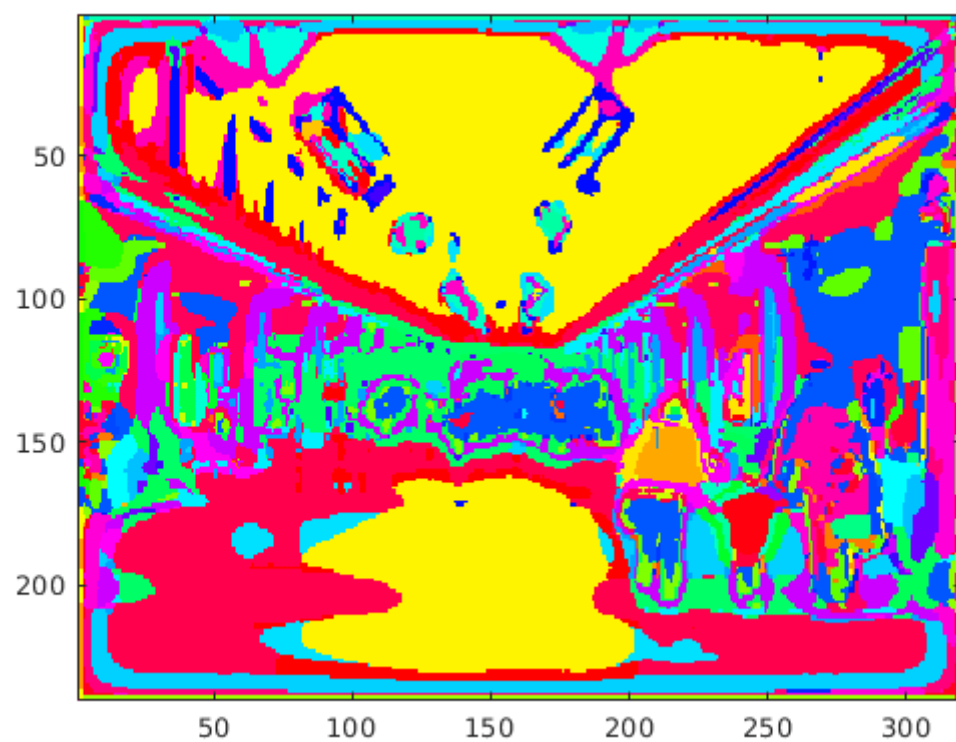
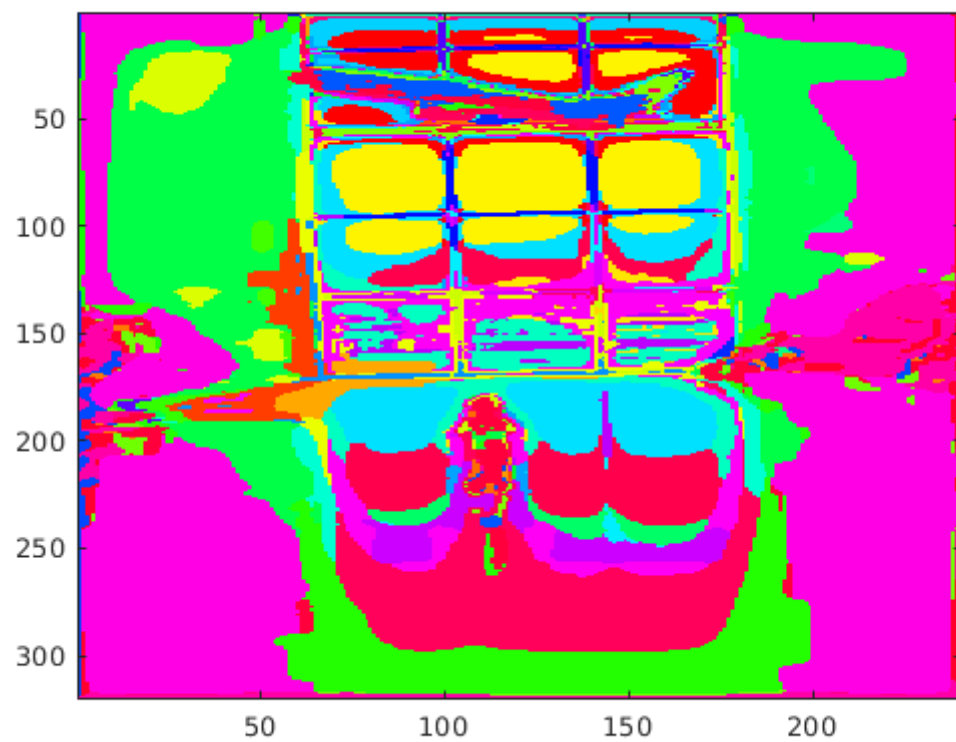
used 50 points per image for kmean

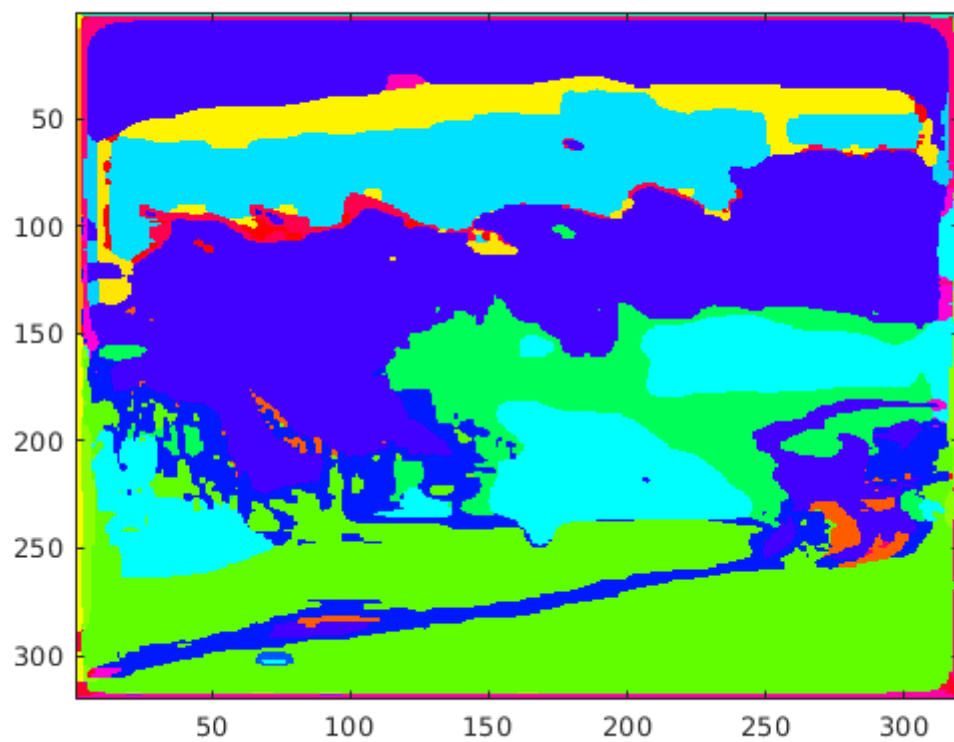




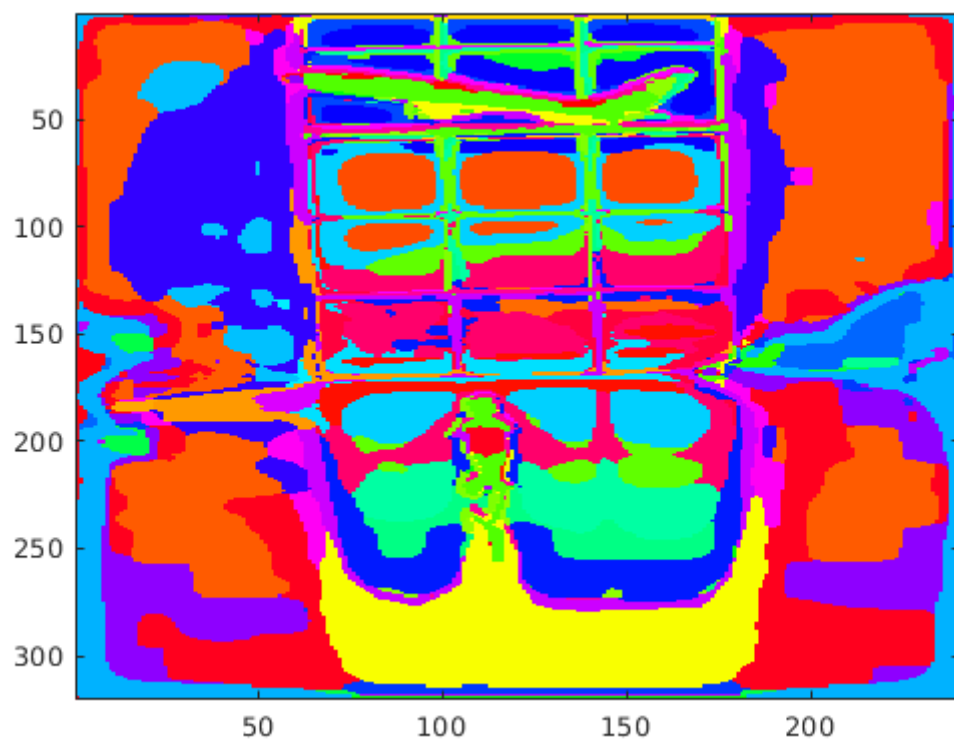
Q2.1 Show the word maps for 3 different images from two different classes (6 images total). Do this for each of the two dictionary types (random and Harris). Are the visual words capturing semantic meanings? Which dictionary seems to be better in your opinion? Why?

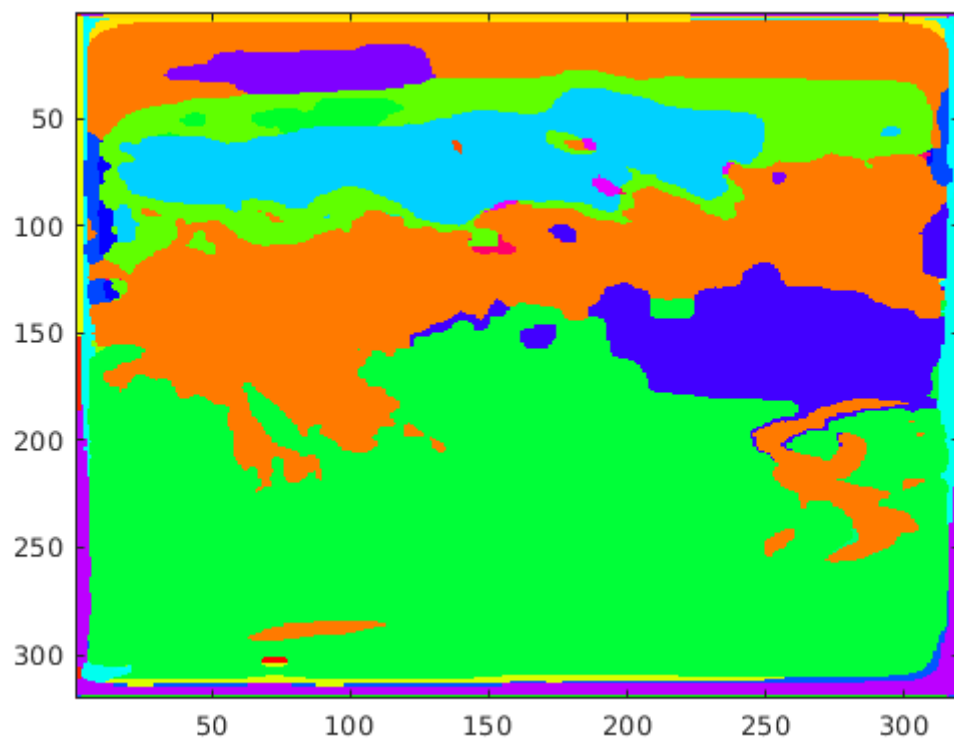
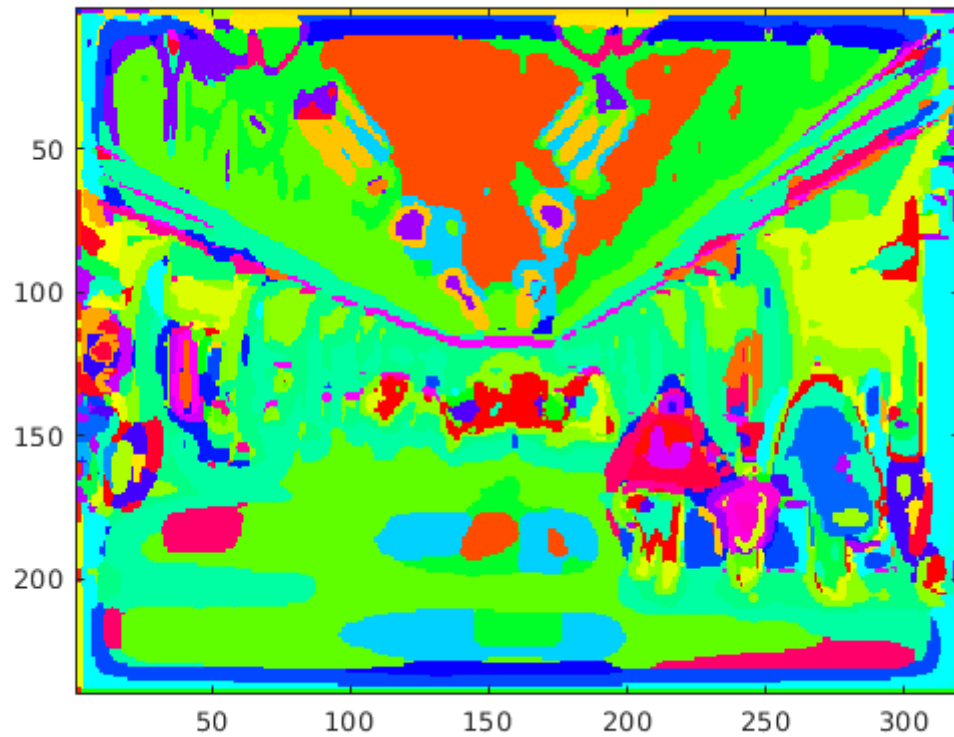
Harris:





Random:





The random ones seems better as a bog model as large region share the same tone, while the harris changes on corner despite if they are the same object.

Q3.2 Comment on whole system:

1NN:

Harris chi2, correct rate: 0.4875 T=1

1	11	3	4					2
2	4	12	2		1	1		
3	4	5	8		3			
4	1	1	3	8	1	3	3	
5		3	3		10		4	
6	1	2	4	1	1	7	2	2
7	1		3	5	3		6	2
8	2			2				16
	1	2	3	4	5	6	7	8

Predicted Class

Harris euclidan, correct rate: 0.43125 T=1

1	8	4	4	2			1	1
2	4	9	3		2	1	1	
3	6	5	8		1			
4	2	1	1	7	1	4	3	1
5	2	2	1	2	10	1	2	
6	1	1		4	2	7	4	1
7	3	1	1	5	3		6	1
8	5			1				14
	1	2	3	4	5	6	7	8

Predicted Class

Random chi2, correct rate: 0.53125 T=1

1	16		3					1
2	2	13	3		1	1		
3	3	3	14					
4	2		2	7		4	5	
5	1	2	1	2	11		3	
6	2	2	3	3		5	2	3
7	2	2	1	3	2	1	6	3
8	3			2			2	13
	1	2	3	4	5	6	7	8

True Class

Predicted Class

Random euclidan, correct rate: 0.4 T=1

1	11	2	3	1			1	2
2	4	11	4		1			
3	7	4	8			1		
4	4	1	1	8	2	3	1	
5	1	4		2	8		5	
6	2	2	3	3	1	4	3	2
7	3	1	2	6	2	1	3	2
8	3			4	1		1	11
	1	2	3	4	5	6	7	8

True Class

Predicted Class

KNN:

1	15	1	2					2
2	3	12	2		2		1	
3	4	4	12					
4	1	1	1	10	1		5	1
5		2	3		14		1	
6	3	3	2	1	1	6	2	2
7	2			2	4		11	1
8	1		1					18
	1	2	3	4	5	6	7	8

1	12	4	2		1			1
2	3	15	1				1	
3	4	5	10	1				
4	3		2	12			3	
5	2	4	3	4	4		3	
6	2	4	2	3	1	4	1	3
7	5	1		5	2		6	1
8	2	1		1				16
	1	2	3	4	5	6	7	8

Random chi2, correct rate: 0.5625 T=5

1	13	2	2	1				2
2	3	14	3					
3	3	2	15					
4	2	1	2	6		2	6	1
5		3	3	1	10		3	
6	2	2	4	2		8	1	1
7	2	1	2	5	2		7	1
8	3							17
	1	2	3	4	5	6	7	8

True Class

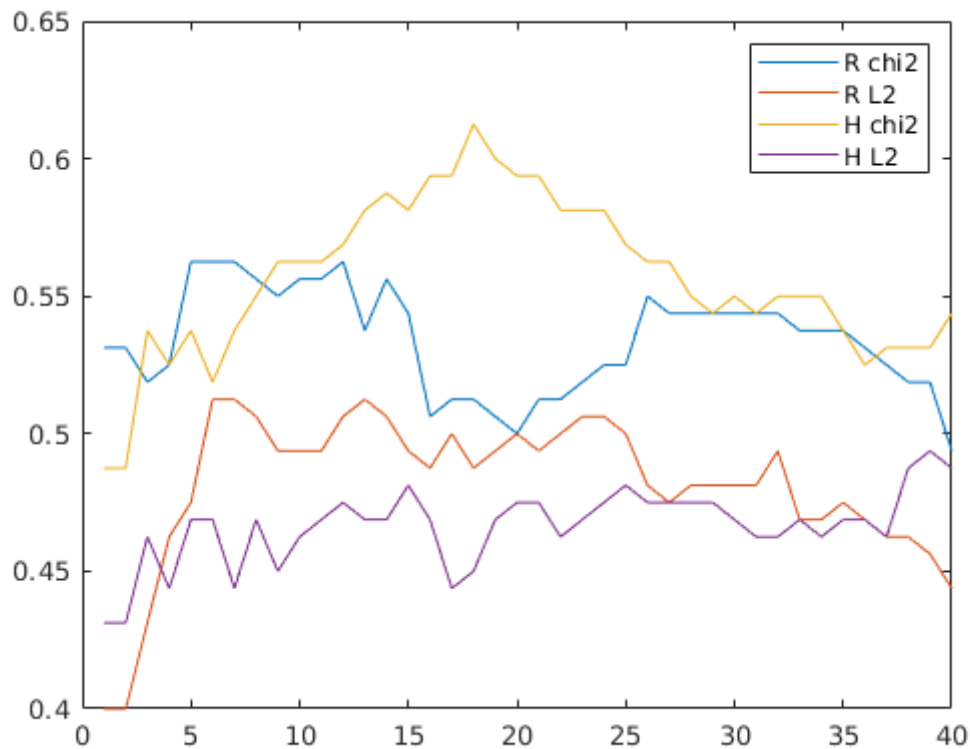
Predicted Class

Random euclidan, correct rate: 0.5125 T=6

1	13	3	2	1				1
2	5	15						
3	5	4	11					
4	3	2		11		2	2	
5	1	5	3	1	6		4	
6	3	3	2	3		4	1	4
7	5	1		6		1	6	1
8	2	1		1				16
	1	2	3	4	5	6	7	8

True Class

Predicted Class



How do the performances of the two dictionaries compare? Is this surprising?

For NN: Random chi-square > Harris chi-square > Harris euclidean > Random euclidean

For KNN: Harris chi-square > Random chi-square > Random euclidean > Harris euclidean

The result is surprising as I cannot actually see the trend of different method. Harris seems to perform better overall as it uses key point

How about the two distance metrics? Which performed better? Why do you think this is?

chi-square distance performs better in both dictionary as it account to the weight of difference.

Are the performances of the SVMs better than nearest neighbor? Why or why not? Does one kernel work better than the other? Why?

Accuracy = 56.25% (90/160) (classification)

The performances of the SVMs does not outperform. Probably because the bottleneck is the dictionary and world map, not the classification method.

How does Inverse Document Frequency affect the performance? Better or worse? Does this make sense?

This perform worse. This is probably because IDF is only used to reduce very frequent word like "The", but image don't have such problem.

What did you experiment with and how did it perform. What was the accuracy?

SURF feature is used(using matlab toolbox)

The accuracy is abit lower but i think it is ok as only ~100 feature point is used for the world map

