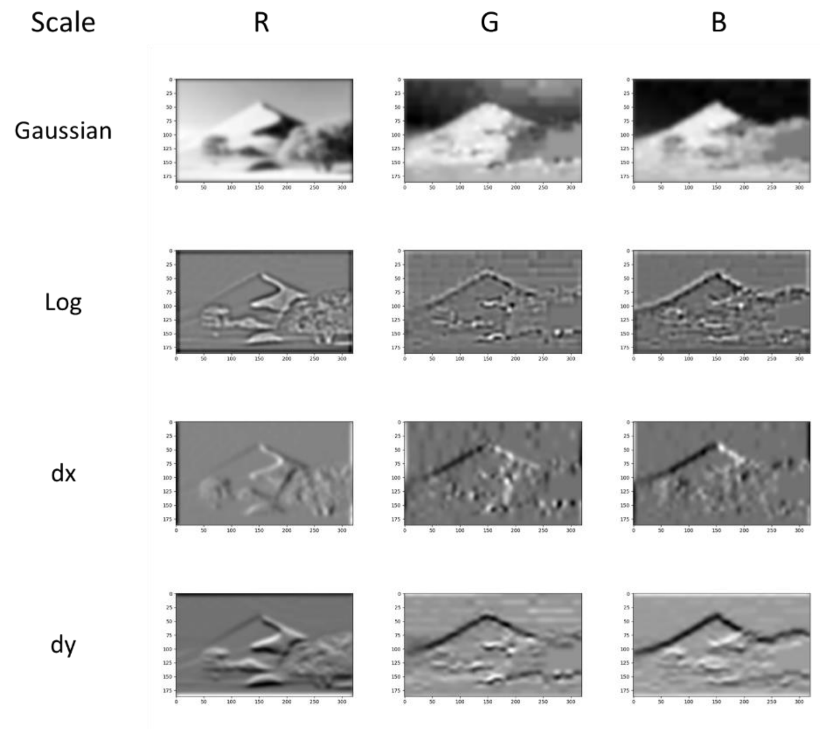
Name: Soo Ah Lee
Andrew ID: sooahl

HOMEWORK ASSIGNMENT 2
Scene recognition with bag of words

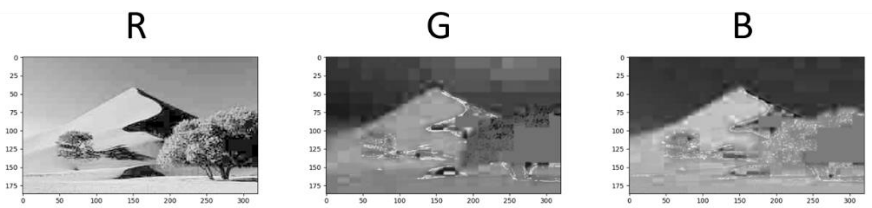
DUE : Mon October 7, 2019 11:59 PM

1. Writeup Summary

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.



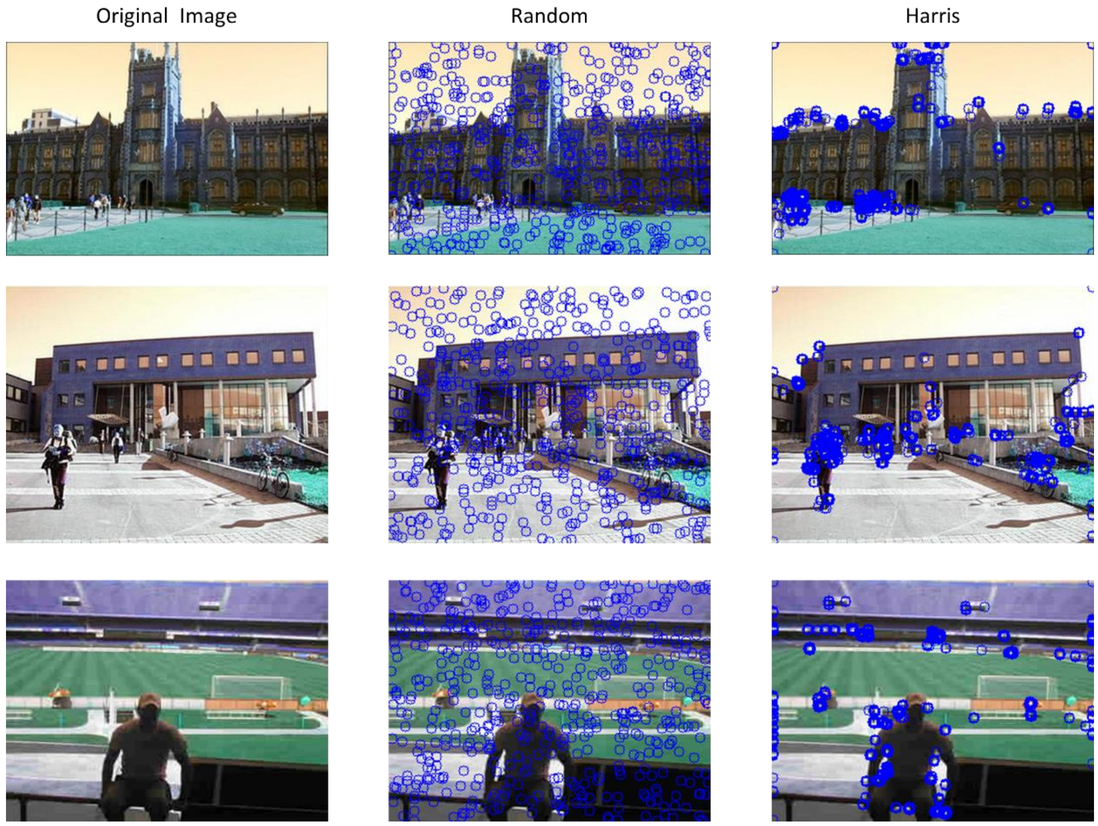
At each channel, we extract each color maps. They have similar brightness in all channels, but the red dress is brighter in the red channel than in the others, and how the green part of the picture is shown brighter in the green channel. We apply each filter to each channel. The below figure represents each color maps.



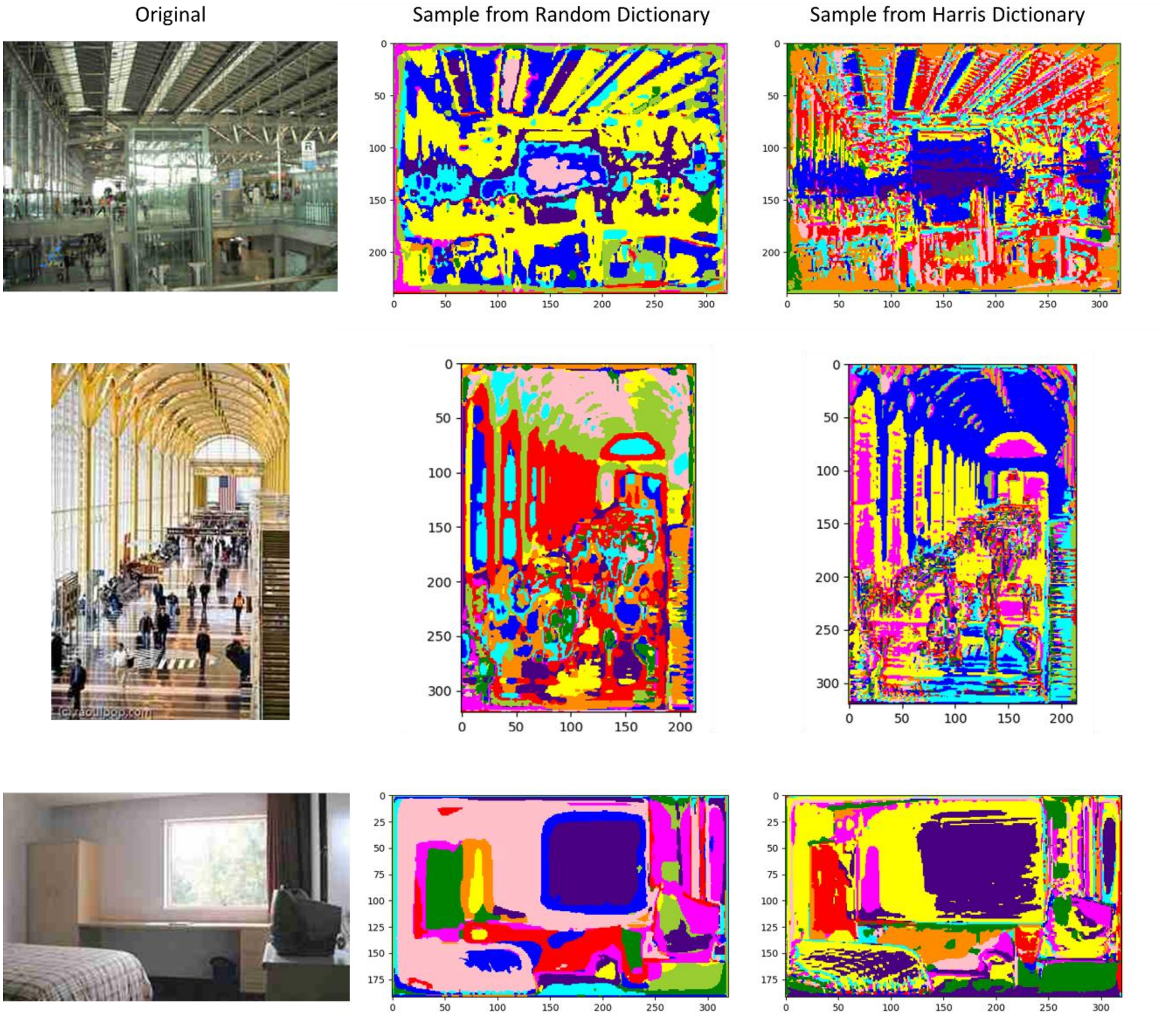
If we apply gaussian filter, the result is blurred. As you see above, each filter blurs each color maps. Especially, you can check blurring at red colormap. The result is more blurring, and the edge is nor clear.
The log filter catches edge of images. It can find areas in the image. So we can see the result of edges. At R colormap, the desert has shadow. So the shadow edge is detected especially at R map. And at B colormap, the difference between mountain and sky is distinct, then the edge of mountain is strong.
The sobel filter only returns the x and y edge responses. I use sobel filter for x gradient as [1 0 -1; 2 0 -2 ; 1 0 -1] and for y gradient as [1 2 1; 0 0 0 ; -1 -2 -1]. Sobel x gradient filter captures change of brightness on horizontal and y gradient filter captures on vertical. It catches distinct in R colormap. At dx, the difference is distinct in shadow and desert, the value is high at shadow of desert. And it is also big value at dy.

CIELAB color space is one type of color space. It expresses color as three values. Brightness for L is from black to white, A is from green to red, B is from blue to yellow. It is designed so that the same amount of numerical change in these values corresponds to roughly the same amount of visual perceived change. We use CIE LAB for object specific color. The LAB space is larger than the gamut of computer displays and printers and because the visual stepwidths are relatively different to the color area.

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



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?



The visual word captures semantic meanings like edge or corners and background or front-ground. The edge is captured, you can see it on second image, especially harris dictionary. The edge of people is distinct. And the background characteristic is captured. In my opinion, the harris capture more characteristics more in common case. But some of images, the random case is better. I think it is because, the brightness is too low, so the corner detection does not work well. But in most case, harris case is better.

Q3.2 Comment on whole system:

i) Include the output of 'evaluateRecognitionSystem.NN.py' (4 confusion matrices and accuracies).

1) $\alpha = 50, K = 100$

Random points & Euclidean distance

```
random euclidean metric
confusion matrix :
[[12.  3.  1.  1.  0.  0.  1.  2.]
 [ 6.  8.  3.  0.  1.  2.  0.  0.]
 [ 6.  7.  5.  1.  1.  0.  0.  0.]
 [ 2.  1.  2.  9.  0.  3.  2.  1.]
 [ 2.  2.  3.  1.  7.  0.  5.  0.]
 [ 2.  3.  3.  0.  1.  6.  3.  2.]
 [ 6.  1.  1.  6.  1.  0.  4.  1.]
 [ 1.  1.  1.  2.  0.  0.  1. 14.]]
random point & euclidean accuracy : 0.406250
```

Random points & Chi distance

```
[random chi metric]
confusion matrix :
[[14.  1.  3.  0.  0.  0.  0.  2.]
 [ 2. 13.  3.  0.  1.  1.  0.  0.]
 [ 4.  4. 11.  1.  0.  0.  0.  0.]
 [ 3.  0.  2.  7.  1.  5.  1.  1.]
 [ 0.  1.  3.  1. 11.  0.  4.  0.]
 [ 1.  2.  4.  0.  0.  6.  5.  2.]
 [ 3.  1.  2.  5.  2.  0.  6.  1.]
 [ 3.  0.  0.  3.  0.  0.  1. 13.]]
random point & chi accuracy : 0.506250
```

Harris points & Euclidean distance

```
[harris euclidean metric]
confusion matrix :
[[ 8.  4.  3.  2.  0.  0.  0.  3.]
 [ 5. 11.  3.  0.  1.  0.  0.  0.]
 [ 7.  2.  9.  0.  1.  1.  0.  0.]
 [ 4.  1.  0.  6.  3.  4.  1.  1.]
 [ 2.  2.  2.  1.  6.  0.  7.  0.]
 [ 1.  2.  2.  4.  1.  4.  4.  2.]
 [ 1.  1.  4.  4.  0.  0.  8.  2.]
 [ 4.  0.  1.  2.  0.  0.  0. 13.]]
harris point & euclidean accuracy : 0.406250
```

Harris points & Chi distance

```
[harris chi metric]
confusion matrix :
[[14.  1.  3.  0.  0.  0.  0.  2.]
 [ 5. 10.  3.  0.  1.  1.  0.  0.]
 [ 4.  3. 11.  1.  1.  0.  0.  0.]
 [ 3.  1.  1.  6.  1.  2.  6.  0.]
 [ 1.  3.  3.  0.  9.  0.  4.  0.]
 [ 0.  4.  3.  1.  1.  6.  3.  2.]
 [ 1.  3.  2.  4.  2.  0.  6.  2.]
 [ 3.  0.  1.  3.  0.  0.  0. 13.]]
harris point & chi accuracy : 0.468750
```

2) $\alpha = 200, K = 500$

Random points & Euclidean distance

```
random euclidean metric
confusion matrix :
[[10.  3.  4.  1.  0.  0.  0.  2.]
 [ 4. 12.  2.  0.  1.  1.  0.  0.]
 [ 4.  6.  7.  1.  1.  1.  0.  0.]
 [ 5.  2.  1.  7.  0.  1.  3.  1.]
 [ 0.  3.  7.  1.  5.  0.  3.  1.]
 [ 3.  3.  0.  5.  2.  4.  2.  1.]
 [ 5.  3.  2.  4.  0.  1.  4.  1.]
 [ 3.  1.  0.  2.  0.  0.  1. 13.]]
random points & euclidean distance : 0.387500
```

Random points & Chi distance

```
[random chi metric]
confusion matrix :
[[15.  2.  1.  0.  0.  0.  0.  2.]
 [ 4. 12.  2.  0.  1.  1.  0.  0.]
 [ 2.  3. 15.  0.  0.  0.  0.  0.]
 [ 4.  1.  0.  8.  1.  1.  4.  1.]
 [ 0.  1.  2.  1. 11.  0.  5.  0.]
 [ 2.  2.  3.  3.  0.  5.  3.  2.]
 [ 4.  2.  2.  2.  1.  1.  7.  1.]
 [ 4.  0.  0.  1.  0.  0.  1. 14.]]
random points & chi distance : 0.543750
```

Harris points & Euclidean distance	Harris points & Chi distance

```
[harris euclidean metric]
confusion matrix :
[[ 8.  2.  3.  1.  0.  1.  1.  4.]
 [ 4. 12.  1.  0.  2.  1.  0.  0.]
 [ 5.  3. 10.  0.  1.  0.  0.  1.]
 [ 1.  3.  0.  9.  0.  3.  3.  1.]
 [ 1.  3.  2.  1.  7.  1.  4.  1.]
 [ 3.  3.  0.  4.  1.  5.  1.  3.]
 [ 3.  2.  3.  5.  2.  0.  3.  2.]
 [ 1.  0.  0.  1.  0.  0.  0. 18.]]
harris points & euclidean distance : 0.450000
```

```
[harris chi metric]
confusion matrix :
[[15.  2.  1.  0.  0.  0.  0.  2.]
 [ 3. 13.  2.  0.  1.  1.  0.  0.]
 [ 1.  5. 13.  0.  1.  0.  0.  0.]
 [ 1.  1.  3.  9.  0.  2.  4.  0.]
 [ 1.  3.  1.  0. 13.  0.  2.  0.]
 [ 3.  1.  2.  1.  1.  8.  2.  2.]
 [ 2.  1.  0.  5.  2.  0.  9.  1.]
 [ 2.  0.  0.  2.  0.  0.  0. 16.]]
harris points & chi distance : 0.600000
```

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

When we use Euclidean distance, the result is same. So we didn't say which one is better. But when we use chi distance, the random point case has higher accuracy. In my thought, harris corner case may be better than random point case. Because it choose points based on corner and edge points. But the result says not. So it is so surprising.

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

In both case, as random point or harris point, when we use chi distance case is better than the other. It is because chi distance used weighting, each term has a weight that is the inverse of its frequency. The weighting can give more accurate distance between histogram.

iv) Also include output of 'evaluateRecognitionSystem_kNN.py (plot and confusion matrix). Comment on the best value of k. Is a larger k always better? Why or Why not? How did you choose to resolve this?

1) $\alpha = 50, K = 100$

Random points & Euclidean distance

```
[random euclidean metric]
confusion matrix :
[[12.  5.  2.  0.  0.  0.  0.  1.]
 [ 4. 16.  0.  0.  0.  0.  0.  0.]
 [ 5.  4. 10.  0.  0.  1.  0.  0.]
 [ 4.  1.  1. 11.  0.  1.  2.  0.]
 [ 5.  5.  2.  2.  4.  1.  1.  0.]
 [ 3.  3.  2.  3.  0.  6.  1.  2.]
 [ 7.  0.  1.  5.  0.  1.  5.  1.]
 [ 2.  1.  0.  0.  0.  0.  0. 17.]]
25 th knn is best - acc : 0.506250
```

Random points & Chi distance

```
[random chi metric]
confusion matrix :
[[14.  2.  2.  1.  0.  0.  0.  1.]
 [ 6. 13.  0.  0.  0.  0.  1.  0.]
 [ 4.  3. 13.  0.  0.  0.  0.  0.]
 [ 3.  1.  1. 11.  0.  0.  3.  1.]
 [ 0.  5.  2.  1. 12.  0.  0.  0.]
 [ 3.  1.  4.  1.  1.  7.  1.  2.]
 [ 4.  0.  2.  5.  2.  0.  6.  1.]
 [ 4.  0.  0.  0.  0.  0.  0. 16.]]
7 th knn is best - acc : 0.575000
```

Harris points & Euclidean distance

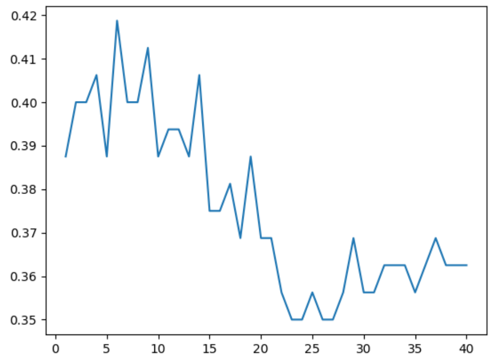
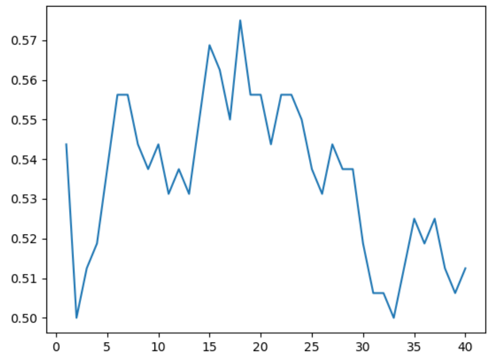
Harris points & Chi distance

[harris euclidean metric] confusion matrix : [[15. 2. 2. 0. 0. 0. 0. 1.] [2. 15. 1. 0. 1. 0. 1. 0.] [2. 5. 13. 0. 0. 0. 0. 0.] [3. 2. 1. 7. 1. 0. 5. 1.] [1. 1. 4. 3. 7. 0. 4. 0.] [4. 3. 2. 4. 1. 3. 2. 1.] [3. 1. 1. 6. 2. 0. 6. 1.] [3. 1. 0. 1. 0. 0. 1. 14.]] 18 th knn is best - acc : 0.500000	[harris chi metric] confusion matrix : [[14. 2. 2. 1. 0. 0. 0. 1.] [3. 13. 2. 0. 1. 0. 1. 0.] [2. 5. 13. 0. 0. 0. 0. 0.] [4. 1. 0. 7. 1. 0. 6. 1.] [3. 2. 3. 2. 6. 0. 4. 0.] [2. 4. 2. 3. 1. 3. 2. 3.] [3. 1. 0. 6. 3. 0. 6. 1.] [4. 1. 0. 1. 0. 0. 1. 13.]] 14 th knn is best - acc : 0.587500

2) $\alpha = 200,\ \mathrm{K} = 500$

Random points & Euclidean distance	Random points & Chi distance
[random euclidean metric] confusion matrix : [[15. 3. 0. 0. 0. 0. 0. 2.] [8. 12. 0. 0. 0. 0. 0. 0.] [5. 4. 9. 2. 0. 0. 0. 0.] [9. 0. 0. 8. 0. 1. 1. 1.] [1. 4. 6. 2. 3. 0. 4. 0.] [6. 4. 0. 3. 1. 3. 0. 3.] [7. 0. 3. 5. 1. 2. 2. 0.] [4. 1. 0. 0. 0. 0. 0. 15.]] 6 th knn is best - acc : 0.418750	[random chi metric] confusion matrix : [[15. 2. 1. 0. 0. 0. 0. 2.] [5. 13. 1. 0. 0. 0. 1. 0.] [4. 3. 13. 0. 0. 0. 0. 0.] [3. 1. 1. 10. 0. 1. 2. 2.] [1. 3. 6. 0. 9. 0. 1. 0.] [2. 2. 3. 2. 1. 8. 1. 1.] [6. 1. 1. 3. 2. 0. 7. 0.] [3. 0. 0. 0. 0. 0. 0. 17.]] 18 th knn is best - acc : 0.575000
Harris points & Euclidean distance	**Harris points & Chi distance**
[harris euclidean metric] confusion matrix : [[8. 2. 3. 1. 0. 1. 1. 4.] [4. 12. 1. 0. 2. 1. 0. 0.] [5. 3. 10. 0. 1. 0. 0. 1.] [1. 3. 0. 9. 0. 3. 3. 1.] [1. 3. 2. 1. 7. 1. 4. 1.] [3. 3. 0. 4. 1. 5. 1. 3.] [3. 2. 3. 5. 2. 0. 3. 2.] [1. 0. 0. 1. 0. 0. 0. 18.]] 1 th knn is best - acc : 0.450000	[harris chi metric] confusion matrix : [[8. 2. 3. 1. 0. 1. 1. 4.] [4. 12. 1. 0. 2. 1. 0. 0.] [5. 3. 10. 0. 1. 0. 0. 1.] [1. 3. 0. 9. 0. 3. 3. 1.] [1. 3. 2. 1. 7. 1. 4. 1.] [3. 3. 0. 4. 1. 5. 1. 3.] [3. 2. 3. 5. 2. 0. 3. 2.] [1. 0. 0. 1. 0. 0. 0. 18.]] 1 th knn is best - acc : 0.600000

For all cases, the best knn case is different. I try all of knn case like below graph. It is case of $\alpha = 200,\ \mathrm{K} = 500$. As n increases, accuracy tends to decrease. Especially, when harris case, the k=1 case has best result. If k increases, test label needs to see many k labels. It will contain noise labels. For reducing this, we just check few knn results.

Random points & Euclidean distance	Random points & Chi distance
	
Harris points & Euclidean distance	**Harris points & Chi distance**