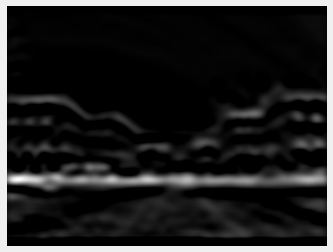
# Project 4 Scene recognition with bag of words

## Q1.1 Extract Filter Responses



Original campus image from the dataset







3 filter responses

I noticed these artifacts: color reversing, blurring, sharpening, horizontal and vertical edge extraction.

CIE Lab color space expresses color using L (lightness), a (from green, negative to red, positive) and b (from blue, negative to yellow, positive). Lab color space is a better approximation to human vision than RGB.

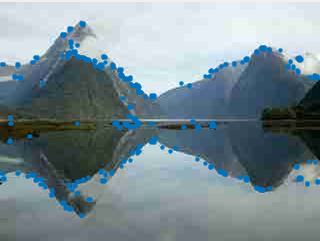
## Q1.2 Collect sample of points from image



Random points on campus image

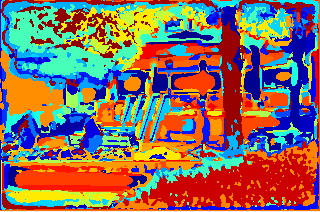
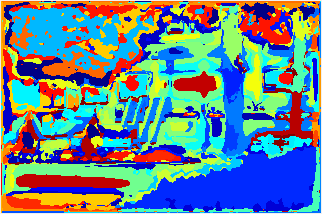


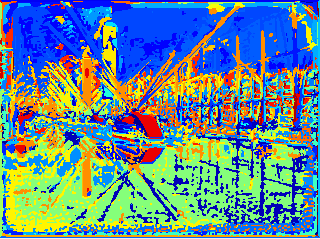
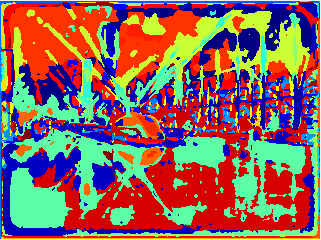


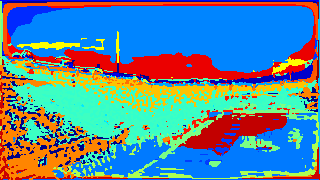
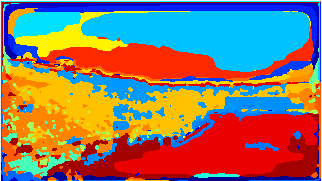


The results of the corner detector on 3 different images

## Q2.1 Convert image to word map





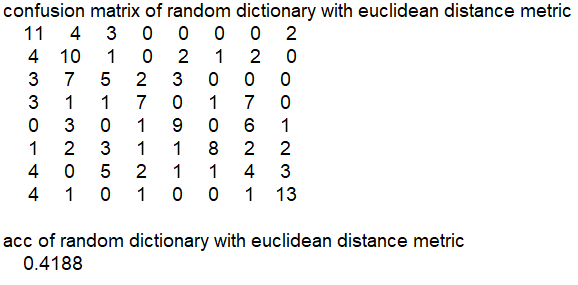


Original Image(left), Word map using Random Dict(middle), Word map using Harris Dict(right)

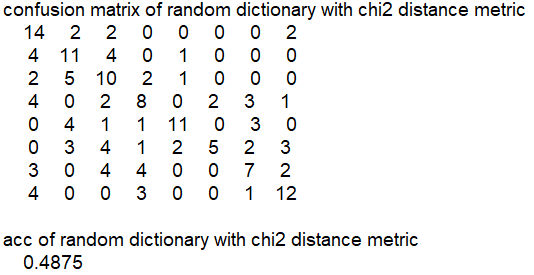
As we can see from the images above, these two visual words does capture the semantic meanings. The Harris dictionary seems better than Random dictionary, because Harris dictionary can clearly show the differences in the details in the picture, but the Random dictionary can only capture the obvious different parts of the images.

## Q3.2 Evaluate Recognition System - NN and kNN

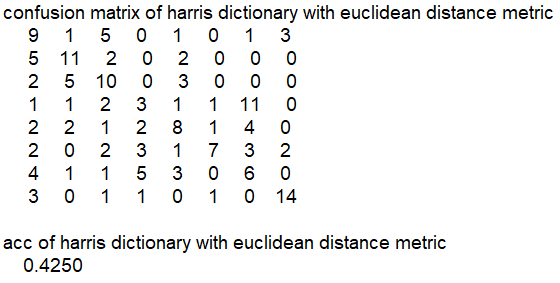
The output of the confusion matrix of using random dictionary with Euclidean distance metric shows below, and the accuracy of this combination is 0.4188



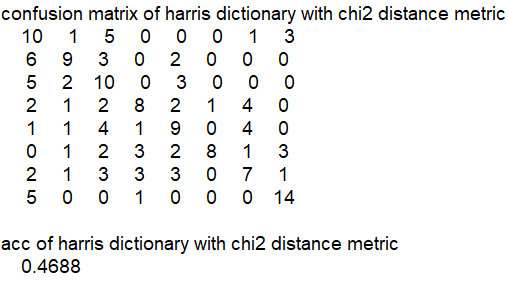
The output of the confusion matrix of using random dictionary with chi2 distance metric shows below, and the accuracy of this combination is 0.4875



The output of the confusion matrix of using harris dictionary with Euclidean distance metric shows below, and the accuracy of this combination is 0.4250



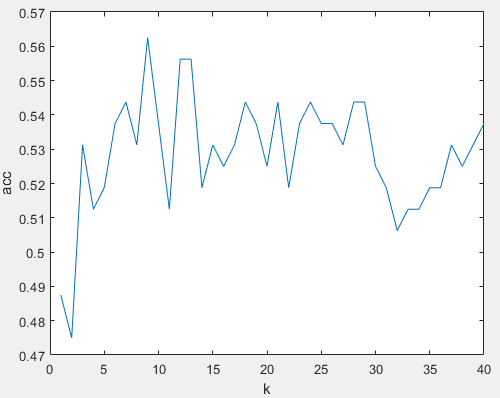
The output of the confusion matrix of using harris dictionary with chi2 distance metric shows below, and the accuracy of this combination is 0.4688



It is surprising that the performance of random dictionary is slightly higher than the harris dictionary.

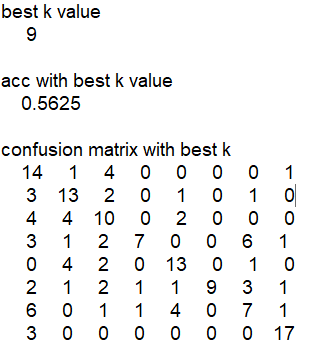
And the chi2 distance metrics performed much better than Euclidean distance metrics. I think it is because chi2 distance punishes relative difference instead of absolute, and chi2 is more sensitive to outliers.

The best combination of dictionary and distance metric is using random dictionary with chi2 distance metric. And here is the plot of evaluate recognition system with knn using different k values.



As is shown on the image below, the best value of k is 9 with acc 0.5625. A larger k doesn’t always work better, because when k is too large, a lot of irrelevant points may be taken into consideration of the classification task.

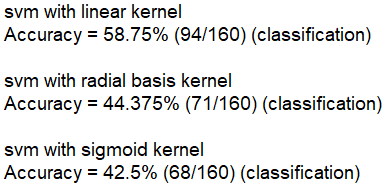
To break the tie of even votes, we can simply choose an odd k that near the best performance k to avoid this situation.



## Q4.1 Evaluate Recognition System - Support Vector Machine

I tried three types of kernels in LIBSVM. The result is shown on the image below. The accuracies for different kernels are 58.75% for linear kernel, 44.375% for radial basis kernel and 42.5% for sigmoid kernel.

The performances for the SVMs are not always better than nearest neighbor. With linear kernel, the svm do beat the nearest neighbor because SVM solves the problem on a higher dimension and is more reliable. But for radial basis kernel and sigmoid kernel, they are equally or even slightly worse than nearest neighbor. And the performance for different kernels vary a lot as is shown on the result.



## Q4.2 Inverse Document Frequency

I used SVM as my classifier. After applying IDF, the accuracy which is 54.375% as is shown on the image below, is slightly lower than before which was 58.75%. This does make sense because features with high frequency may doesn’t mean the same thing in natural language processing and computer vision. It might be the cause that in images those high frequency features do represent the image.



## Q4.3 Better pixel features

I did experiment with Gabor filter and HOG feature extraction using different K values and evaluating using SVM.

### Result of HOG feature extraction

In the experiment of HOG feature extraction I used built-in Matlab function extractHOGFeatures().

For HOG feature extraction, a larger K value cannot always provide better result. I chose K value with 1000, and after fine-tuning the parameter of SVM, I got accuracy 63.75% as is shown on the image below with sigmoid kernel SVM.



### Result of Gabor filter

In the experiment of Gabor filter, I used Gabor filter bank and Gabor feature extraction by Mohammad Haghighat (2020). Gabor Feature Extraction (https://www.github.com/mhaghighat/gabor), GitHub. Retrieved March 14, 2020.

For Gabor filter, I find that the accuracy is better when using larger K value. In the experiment I chose K value with 86400 without normalization. And the result shows on the image below.

The accuracy of using Gabor filter with radial basis kernel SVM is 43.75%.



The accuracy of using Gabor filter with sigmoid kernel SVM is 41.25%.



The overall performance of Gabor filter is slightly worse than the original filter bank, and it runs much slower too because to get a better result I have to choose larger K value.