

Assignment 4

Vishal Batvia

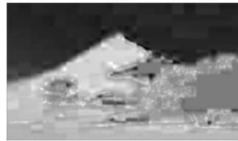
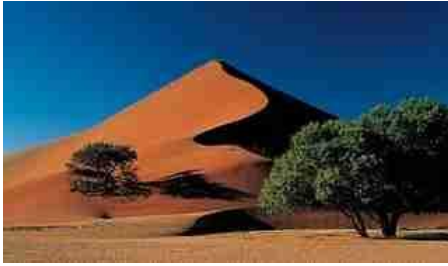
CMPT 762 Computer Vision

Author Note

Assignment 4

1.1. Extract Feature Responses

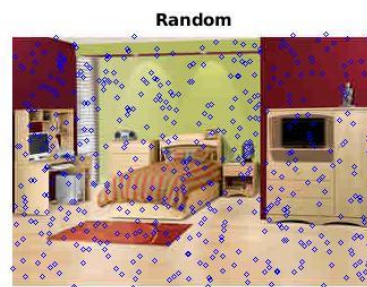
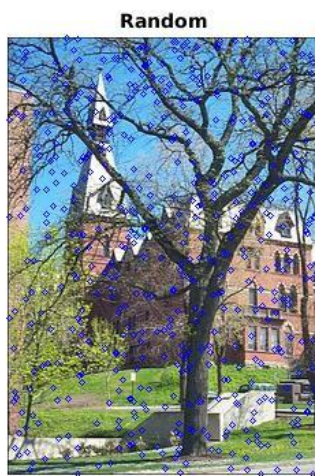
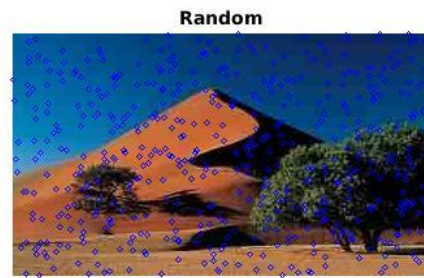
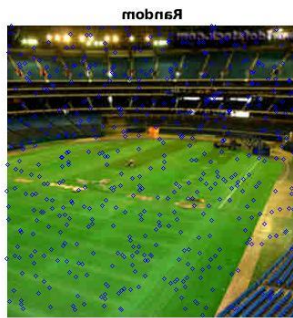




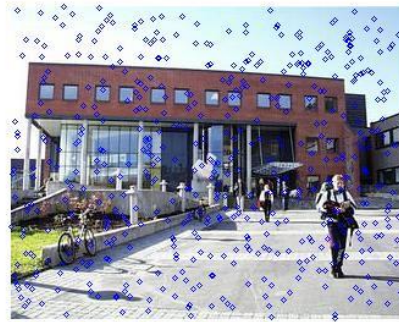
CIE Lab Color space is defined by International Commission on Illumination and it consists of L, a, b. L is the measure of lightness, a is a measure of green to red and b is a measure of blue to yellow. In the task in current assignment it will be useful to have a sense of lightness of a pixel to find the details about its corners.

1.2. Collect sample of points from image

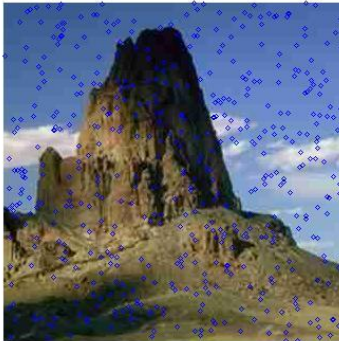
Random



Random



Random



Harris

Harris



Harris

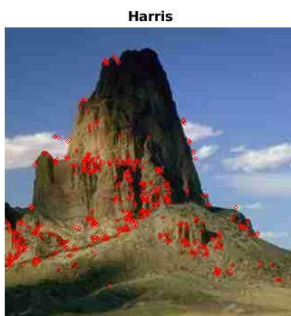
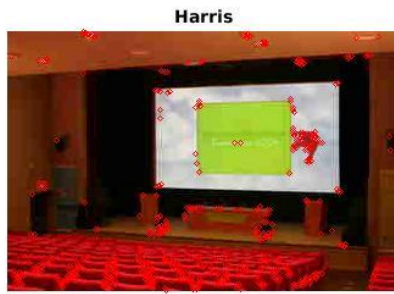


Harris



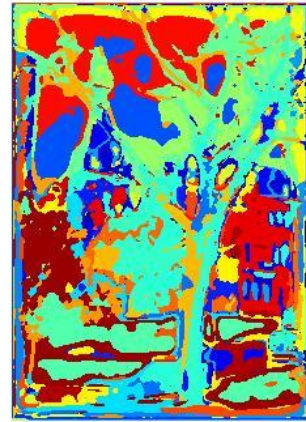
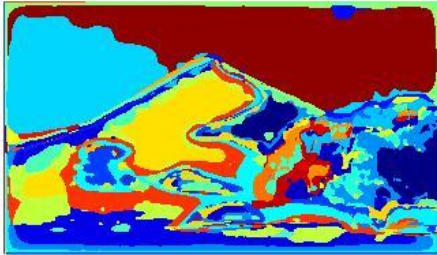
Harris

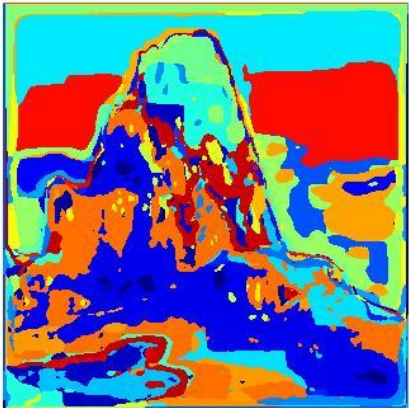
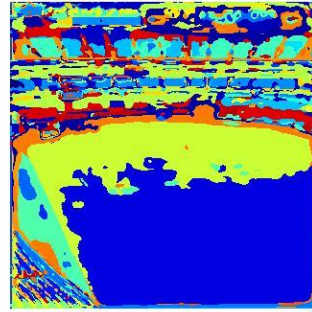
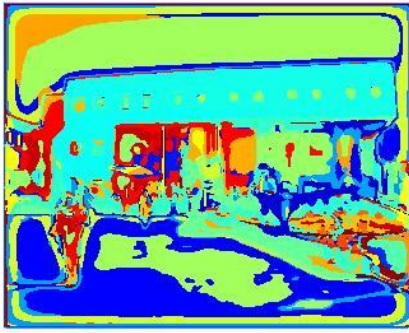




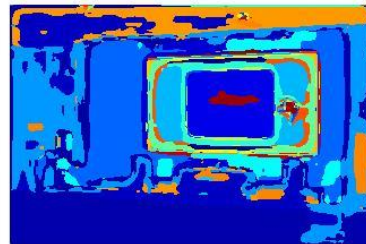
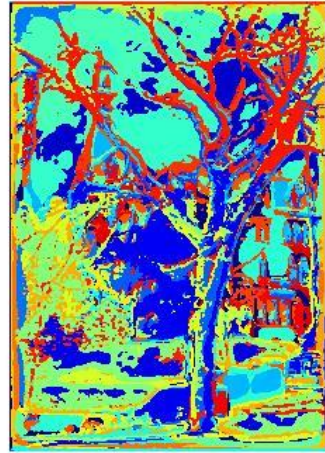
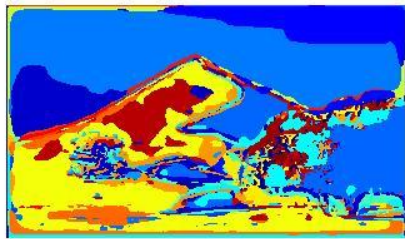
2.1. Convert image to word map

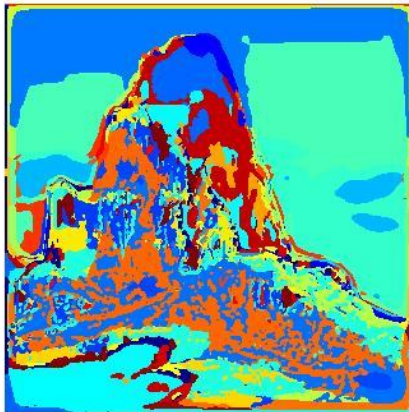
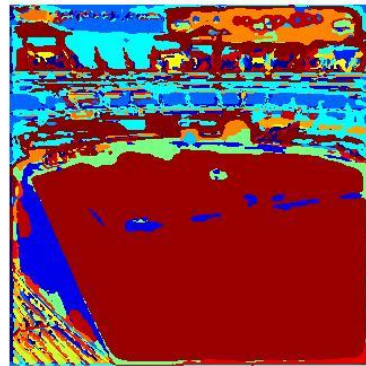
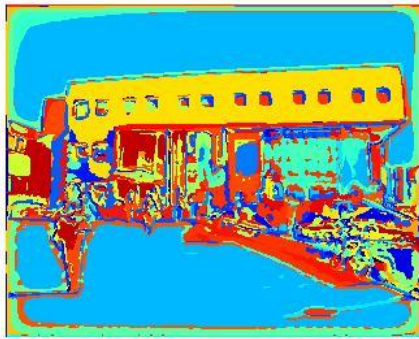
Random





Harris

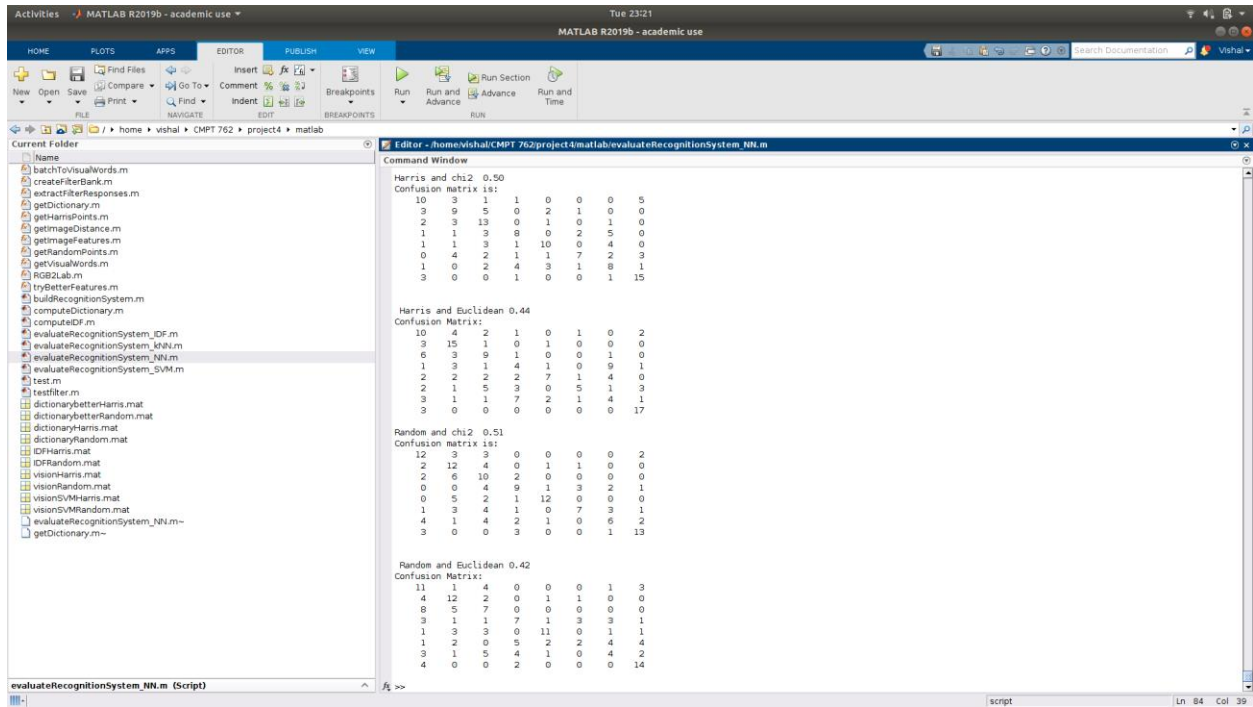




Harris looks better in the images as it gives some semantic meanings as well

3.2. Evaluate Recognition System (NN and KNN)

Accuracies and confusion matrix with NN



As Chi2 produced better results in previous I have given the image of Harris and chi2 below

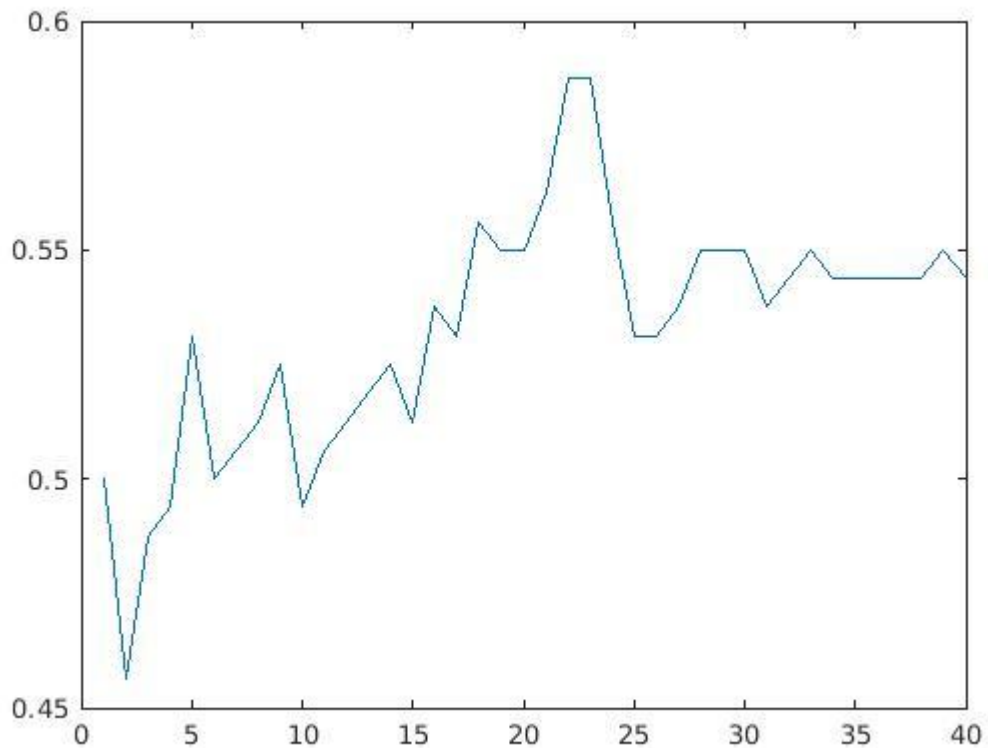
KNN Harris and Chi2

```
>> evaluateRecognitionSystem_kNN
Running kNN
Harris and chi2 Best accuracy is    0.5875

at K    22

    15     2     2     0     0     0     0     1
     4    14     1     0     0     0     1     0
     4     5    11     0     0     0     0     0
     2     2     0    12     0     0     4     0
     0     4     4     0    10     0     2     0
     1     3     4     3     0     6     2     1
     3     0     2     4     3     0     8     0
     1     0     1     0     0     0     0    18

Warning: MATLAB has disabled some advanced graphics rendering features t
^ f_x >>
```



4.1. Evaluate Recognition System (SVM)

SVM is tested with gaussian and linear kernel and gaussian works better.

Editor - home\visual\CMPT 762\project4\matlab\evaluateRecognitionSystem_SVM.m

```

40 - batchToVisualWords.m = buildRecognitionSystem.m; evaluateRecognitionSystem_NN.m; evaluateRecognitionSystem_KNN.m; evaluateRecognitionSystem_SVM.m
41 -
42 -
43 - size_dict=size(dictionary,1);
44 - load('visionRandom.mat');
45 -
46 - t=templateSVM('kernelFunction','linear');
47 -
48 - classifier = fitcecoc(train_features,train_labels,'Learners',t);
49 -
50 - save('visionSVMRandom.mat','classifier');
51 -
52 - correct=0;
53 - incorrect=0;
54 - C=Randomzeros(8,8);
55 -
56 - for i=1:len_test
57 -     wordMap=load(['../data/', sprintf(test_imagenames(i),'.jpg','.r.mat')], 'wordMap');
58 -     test_features=getImageFeatures(wordMap,wordMap, size_dict);
59 -     outputLabel=predict(classifier,test_features);
60 -     C=Random(test_labels(i),outputLabel)+C;
61 - end

```

Command Window

```

>> evaluateRecognitionSystem_SVM
Accuracy with Harris 0.4988

14 3 2 0 0 0 0 1
5 13 1 0 0 0 1 0
3 6 11 0 0 0 0 0
3 1 1 5 0 10 0
0 4 2 0 9 0 4 1
6 2 4 2 0 1 4 1
6 0 1 0 5 0 8 0
5 1 0 0 0 0 0 14

Accuracy with Random 0.4988

14 4 1 0 0 0 0 1
5 13 0 0 1 0 1 0
3 6 10 0 1 0 0 0
3 1 1 4 0 0 11 0
0 4 4 0 7 0 5 0
5 3 2 2 1 1 5 1
6 0 0 1 5 0 8 0
4 1 0 0 0 0 1 14

```

Editor - home\visual\CMPT 762\project4\matlab\evaluateRecognitionSystem_SVM.m

```

34 - accuracy=correct/total;
35 - fprintf('Accuracy with Harris')
36 - disp(accuracy)
37 - C=disp(C);
38 - N=Random;
39 -
40 - load('dictionaryRandom.mat');
41 -
42 - size_dict=size(dictionary,1);
43 - load('visionRandom.mat');
44 -
45 - t=templateSVM('kernelFunction','gaussian');
46 -
47 - classifier = fitcecoc(train_features,train_labels,'Learners',t);
48 -
49 - save('visionSVMRandom.mat','classifier');
50 -
51 - correct=0;
52 - incorrect=0;
53 - C=Randomzeros(8,8);
54 -
55 -

```

Command Window

```

>> evaluateRecognitionSystem_SVM
Accuracy with Harris 0.5000

13 5 1 0 0 0 0 1
5 12 0 0 2 0 1 0
3 6 11 0 0 0 0 0
2 1 1 8 0 0 8 0
1 3 2 0 6 0 5 0
4 2 4 2 0 3 4 1
2 0 1 1 5 0 11 0
5 1 0 0 0 0 0 14

Accuracy with Random 0.5000

14 4 1 0 0 0 0 1
5 13 0 0 1 0 1 0
2 7 11 0 0 0 0 0
3 1 1 9 0 1 5 0
0 3 2 1 10 0 4 0
2 4 2 3 1 4 3 1
5 0 2 3 2 0 8 0
3 1 0 0 0 0 1 15

```

4.2. Inverse Document Frequency

Matrix is present in the code please check it.

4.3. Better Pixel Features

I tried gobar filters.

Transformation is here and code is present in the folder



Experiments and performance