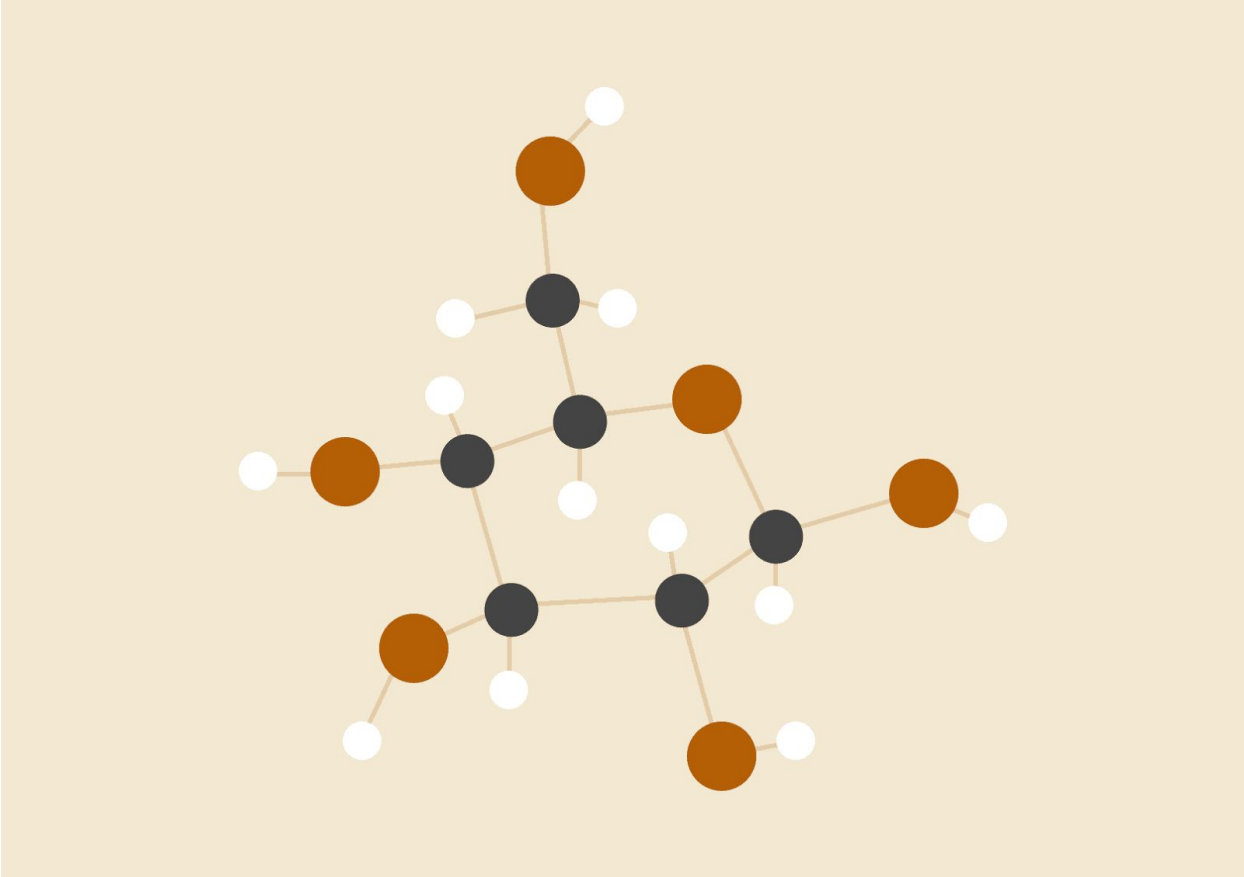
HOMEWORK 1 REPORT

Spatial Pyramid Matching for Scene Classification

Computer Vision and Image Processing CSE 573



Jayant Solanki

UBIT Name: jayantso
Person Number: 50246821
Fall 2017 CSE

INTRODUCTION

This report talks about the results of the application of the **Statistical Pyramid Matching** (SPM) on the **bag-of-words** (BoW) approach. BoW approach is used for object recognition and scene classification. In this report we will talk about how we implemented. We have been provided with training and testing case of images. From the directions given in the Section of the **hw1.pdf** 60 features has been generated for each image. From those 60 features, visual words has been created using **K-Means clustering,** which has been used as dictionary. Dictionary is then used for generating visual words vector for every images. Section 2 helps us to implement SPM approach on the on the histograms generated from the visual words vector of different images. We trained our algorithm based upon the **3-layered** histograms matching. Finally we checked for the accuracy of the trained algorithm in classifying 160 test images

IMPLEMENTATION

Section 1.0: This section talks about the application of Gaussian filters on the image. Gaussian Filter is a spatial filter whose main purpose is for smoothing 'blurring' the image on which it is applied. It has a kernel which is represents the gaussian distribution of a random variable (Bell Shape).

The formula for univariate Gaussian distribution has the following form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

The formula for a bivariate Gaussian distribution has the following form:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The Gaussian filter is the only circular symmetric filter. It is applied using the convolution method on the 2-D image. Degree of the blurriness is found by the standard deviation of the Gaussian kernel. It outputs a weighted average of each pixel's

neighbouring which centers around the central pixels. One of main advantage of using Gaussian filter is its strict frequency response boundaries. It weakens the high frequency components of the image and shows no oscillations compared to the mean filter. **This property largely identifies Gaussian filter in edge detection.** Other properties include Gaussian noise reduction.

There are two broad categories of filters used in the **createFilterBank** matlab script.

- Gaussian
- Log of Gaussian that is Laplacian of Gaussian Filter

From MATLAB documentation, `h = fspecial('gaussian',hsize,sigma)`, returns a rotationally symmetric Gaussian lowpass filter of size hsize with standard deviation sigma, where **hsize** represents the size of the filter, **hsize** can be scalar or vector. In the script it is a vector. **hsize** represents the size of the filter matrix, in the script, there are 5 matrices for each 4 filters, their sizes are: **[7×7 double], [11×11 double], [21×21 double], [41×41 double], [59×59 double]**. Sigma represents the standard deviation to be maintained in the each filter matrices. Similarly `h = fspecial('log',hsize,sigma),` returns the laplacian of gaussian filter of hsize and sigma standard deviation.

First Gaussian filter declares 5 filter matrices of sizes **[7×7 double], [11×11 double], [21×21 double], [41×41 double], [59×59 double]** with elements having standard deviation of **1, 2, 4, 8, and 11.3137**.

Seconds Filter is a Laplacian of Gaussian filter, declaring 5 filter matrices of sizes **[7×7 double], [11×11 double], [21×21 double], [41×41 double], [59×59 double]** with elements having standard deviation of **1, 2, 4, 8, and 11.3137.**

Fourth Gaussian filter declares 5 filter matrices of sizes **[7×7 double], [11×11 double], [21×21 double], [41×41 double], [59×59 double]** with elements having standard deviation of **1, 2, 4, 8, and 11.3137**. But with the property where every element in the middle column of each filter matrix is zero, and column right of the middle column are negative mirror image of columns left of the middle column.

Third Gaussian filter declares 5 filter matrices of sizes **[7×7 double], [11×11 double], [21×21 double], [41×41 double], [59×59 double]** with elements having standard deviation of **1, 2, 4, 8, and 11.3137**. But with the property where every element in the middle row of each filter matrix is zero, and rows below the middle row are negative mirror image of rows above the middle row.

There are 20 filter responses , so each image will have 60 filter responses after

considering the 3 channels in each image.

Section 1.1: Montage of image **"ice_skating\sun_advbapyfkehgemjf.jpg",** Figure 1.1

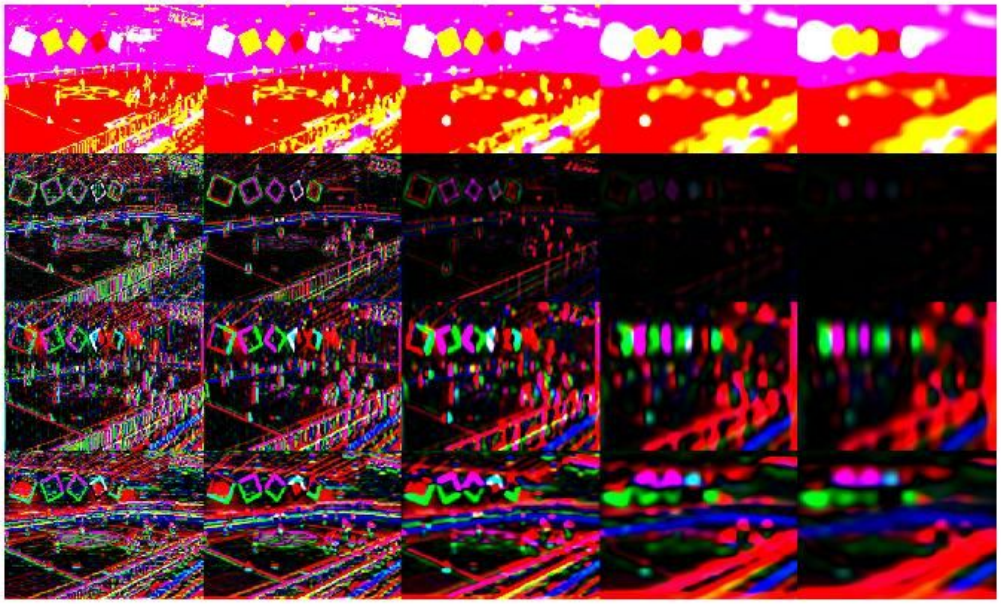
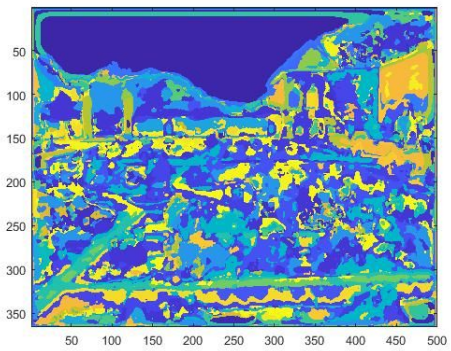


Figure 1.1

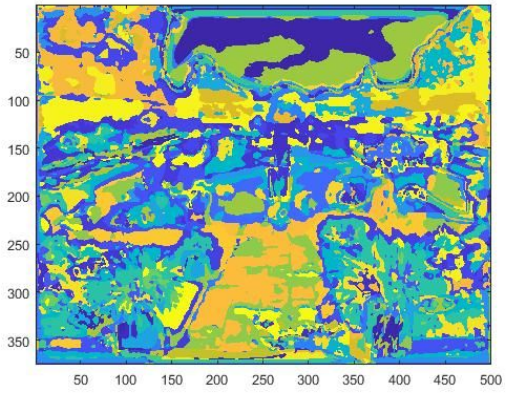
Section 1.2: dictionary.mat is generated using k-means clustering for **K** clusters on the 60 filter responses of every image in the training set.

Section 1.3: Visualisation of three wordMaps from Garden category

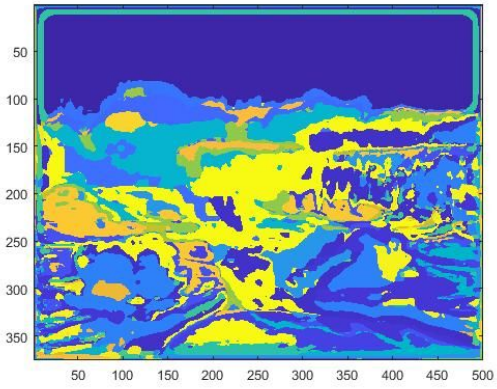
Original: sun_bbzrfihsevbsmiyz.jpg



wordMap: sun_bbzrfihsevbsmiyz-wordMap.jpg



Original: sun_bcgmbjdhmotrstqr.jpg wordMap: sun_bcgmbjdhmotrstqr-wordMap.jpg



Original: sun_bdcfingxwkzaozgs.jpg wordMap: sun_bdcfingxwkzaozgs-wordMap.jpg

Each wordMap has been generated for alpha value 125 and clusters value 200. More is the cluster value more diverse can be the visual depiction of the wordMap. More clusters results in more discrete grouping of the pixels having similar features.

Section 2.1: Normalised histogram generated as directed in the Section 2.1 hw1.pdf.

Section 2.2: Using SPM till layer 3, histogram of each image generated. Do note that generation of layer 0 and layer 1 histogram from the layer 3 histogram negatively affected the accuracy, so separate histograms were calculated for layer 0 and 1 which were later concatenated to the layer 3 histograms. Layer 0 has one histogram , of Kx1 dimension, Layer 1 has 4 histograms each of Kx1 dimension, Layer 3 had 16 histograms, each having Kx1 dimensions. K is the dictionary size. So after concatenation net histogram has 21Kx1 dimension.

Section 2.3: Histograms were compared using normal histogram intersection matching technique.

Section 2.4: vision.mat was generated after following section 2.4.

Section 2.5: Accuracy/Confusion matrix for the testing images is mentioned below: Please note that accuracy could have been higher, but the appropriate cluster and alpha have yet to be determined. Below data was found after different permutations of clusters sizes and alpha.

C is the confusion matrix, for alpha=150 and clusters size=200, using L*a*b* colorspace, with the accuracy of 58.75 using SPM, highest till now found:

Without SPM

C =

10	1	0	3	3	0	1	2
1	11	1	2	5	0	0	0
0	0	18	1	0	1	0	0
1	2	1	15	0	1	0	0
4	4	0	2	9	1	0	0
0	1	2	2	2	10	2	1
1	1	0	0	0	4	14	0
4	5	3	0	2	1	1	4

order =

1
2
3
4
5
6
7
8

accuracy =

0.5687

With SPM

C =

```
    7    1    0    2    7    0    2    1
    1   12    0    1    5    1    0    0
    0    1   19    0    0    0    0    0
    0    2    1   15    2    0    0    0
    4    5    0    0   11    0    0    0
    0    1    2    1    1   10    5    0
    1    0    0    0    1    2   16    0
    1    3    3    5    2    2    0    4
```

order =

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

accuracy =

```
    0.5875
```

Class	Correct Recognition	Accuracy
Garden	19	0.95
Ocean	16	0.8
Ice-skating	15	0.75
Computer Room	12	0.6
Library	11	0.55
Mountain	10	0.5
Art Gallery	7	0.35
Tennis-court	4	0.2
Total	**94**	**0.5875**

Table depicting the accuracy for each test class (SPM)

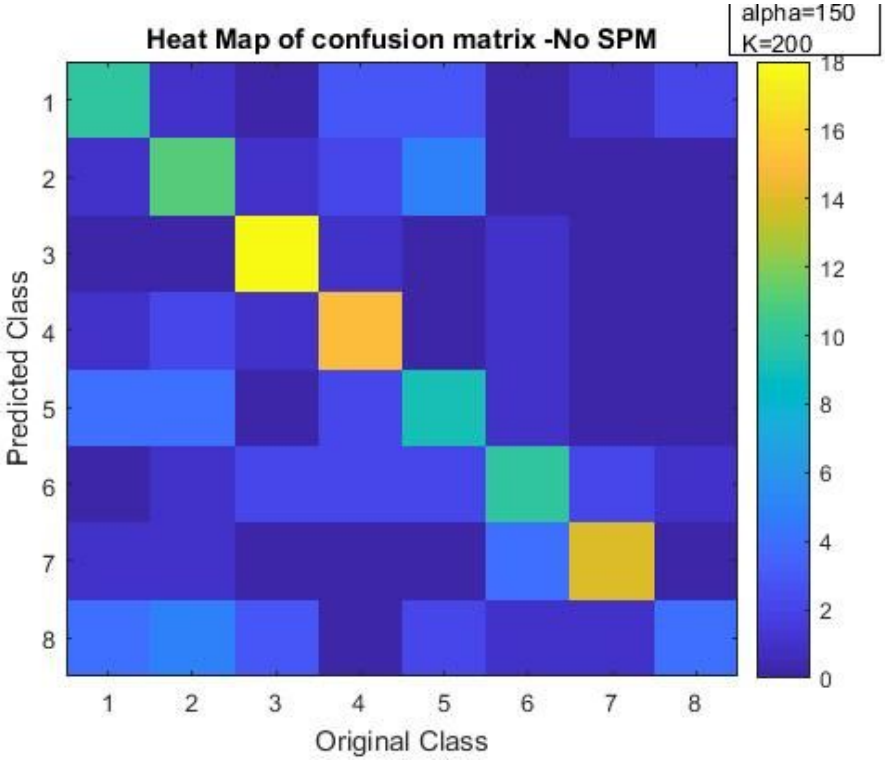


Figure 2.5.1

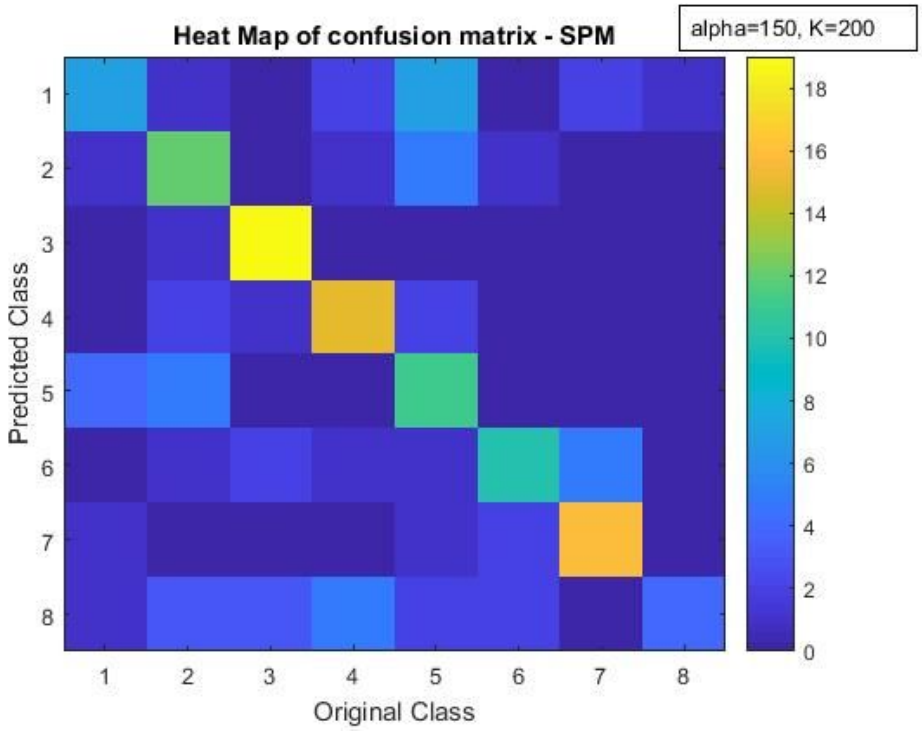


Figure 2.5.2

Section 2.6: From the above accuracy matrix and accuracy table we can note following observations:

- Garden class is the most easily recognisable class. The main reason for easy classification is the color green and most abundant presence of different types of edges.
- Most least and most confused class is Tennis-court. Main reason for its misclassification in nearly all other classes is because of minimal presence of edges. Lack of edges is making it misclassifying with ice-skating class, which also has minimal edges but more prominence of white color. Secondly if the scene recognition is done without using SPM, we notice that accuracy of Tennis court class doesn't change, but the distribution of misclassification gets localised into few classes only, which are namely art-gallery, computer-rooms, and garden. This can be explained due to the fact that, presence of straight edges in the tennis-court classes is not broken in the histograms which are not using the SPM. In fact the use of SPM is making the misclassification go up in the test-images.
- Mountain is sometimes confused with Ocean, due to the presence of no edges in the sky water and the color blue. It is also getting misclassified as garden owing to the presence of trees and green color abundance. Similarly ocean is sometimes misclassified as mountain.
- Library and computer rooms are getting misclassified with each other mostly due the presence of rectangular edges and tables.
- Art gallery is mostly getting confused with the library, due to the presence of discrete rectangular shapes in both classes. We can notice that the misclassification decreases if SPM is not used. In the absence of SPM it is misclassified as ice-skating class which can be explained owing to the fact that there is abundance of white color in both classes.
- Ice-skating is easily recognizable owing to the presence of white color. It is sometimes misclassified as computer-rooms and library which can be explained with the fact the there are straight edges in the ice-skating class which makes it match with later two classes.

In short SPM makes the discrete polygonal objects more prominent which are hampering with the correct recognition of those classes which have similar polygonal structures, such as art-gallery, tennis-court, library and computer-room. Whereas SPM is considerably improving the results where the pattern of edges in parts of image is more vital in image recognition.

Section 2.7: I have included the custom folder where I have implemented the scene recognition using the HSV color space. Though the results are not better than L*a*b* color space. However they are better in few combinations of cluster and alpha values. In those cases scene recognition using SPM in L*a*b* color space performed poorly than Non SPM in L*a*b* color space. Datasheet for the accuracy comparison has been included below.

Overall if the application is human-vision centered, such as scene recognition, using L*a*b* color space is recommended. However in some cases we cases we need better representation of colors, where HSV color space comes in handy. HSV is better suited in those cases where the edges detection is required more.[1]

I also observed that the reconstruction of layer 0 and layer 1 histograms from the layer 3 histograms was not totally similar to layer 0 histogram calculated directly. This in turn was hampering the accuracy of the scene recognition. So in the code, both in the matlab folder and custom folder, I have used the direct calculation of the histograms.

Computation of dictionary and wordMap can be optimised by directly taking the alpha pixels from each image and taking the filter response of them. Currently I am not using the this step and the code is taking few hours to generate the wordMap and dictionary time to generate also varies with different combinations of alpha and clusters. I have added ***parfor*** in the **batchToVisualWords.m** script to parallelise the wordMap generation.

Code can be further optimised using SVM for training sets. Histogram Intersection kernel can be replaced with Generalised Histogram Intersection kernel[2].

[1]: The Effect of Color Space Selection on Detectability and Discriminability of Colored Objects

[2]: GHM

Datasheet for the accuracy comparison

Id.	Alpha	Clusters	ImageFormat	Accuracy(SPM)(%)	Accuracy(NoSPM)(%)	Change in accuracy(%)	Remark
1	70	150	LAB	51.88	49.38	2.5	Reconstruction of histogram was taken
2	70	150	LAB	54.37	49.38	4.99	
3	70	150	HSV	54.37	49.38	4.99	
4	100	150	LAB	45	47.5	-2.5	
5	100	150	HSV	51.88	47.5	4.38	
6	125	200	LAB	57.5	48.75	8.75	
7	125	200	HSV	51.88	50.62	1.26	
8	150	200	HSV	58.13	48.13	10	
9	150	200	LAB	58.75	56.87	1.88	
10	150	150	HSV	50	47.5	2.5	
11	150	150	LAB	48.75	43.13	5.62	
12	180	180	HSV	48.13	50.62	-2.49	
13	180	180	LAB	56.87	45.62	11.25	
14	200	300	LAB	51.25	41.88	9.37	Reconstruction of histogram was taken
15	200	300	LAB	54.37	41.88	12.49	
16	200	200	LAB	54.37	46.88	7.49	
17	200	125	LAB	55	47.5	7.5	
18	200	200	HSV	47.5	46.25	1.25	
19	200	300	HSV	56.25	50.62	5.63	
20	200	125	HSV	50	41.88	8.12	

DATASHEET

1. Accuracy table: https://goo.gl/HAJwq1
2. Confusion matrix Dump: https://goo.gl/ZwgmCZ

SOFTWARE/HARDWARE USED

1. MATLAB R2017a on personal system.
2. Intel i3, 3GB Ram, Sublime Text 3 IDE.
3. Training also done on timberlake and Styx Systems where MATLAB version was 2015..

REFERENCES

1. MATLAB Documentation
2. Hw1.pdf
3. Grauman_darrell_iccv2005.pdf
4. cvpr06b.pdf