Computer Vision - 16720
Homework 1

[redacted]
Andrew id: [redacted]

Start date: Jan 25th
Due date: Feb 7th

Q1.0: What Properties do Each of the Filter Functions Pick Up?

Ans: **Gaussian Filter** is kind of a low-pass filter that filtered out the high frequency components. It's like a convolution(or correlation if not flip) and thus picks up the "average" value among the center pixel and the pixels around. By this the high frequency value(like noise) will be filtered out and the picture will be smoothed.
The LoG Filter(Laplacian of Gaussian Filter) is a kind of spacial derivative filter to detect the edges. So it's aiming at picking up the edge information of the input.
The dx Filter is a kind of derivative filter that pick up the edge information along x axis, to detect if there's any change in texture horizontally.
The dy Filter is also derivative filter that pick up the edge information along y axis, to detect if there's any change in texture vertically.

Q1.1: Extracting Filter Responeses

Ans: I chose an image from class art gallery in SUN dataset.
As in figure shown, is the 4*5 output of function ectractFilterResponses. At first my output was a 2-dimension matrix with w*h rows and 3*F columns because I think it might be easier to use it as an input for the later function. I changed it into a 3-dimension matrix after I saw the question on piazza but I also keep the old version in comment just for reference:)

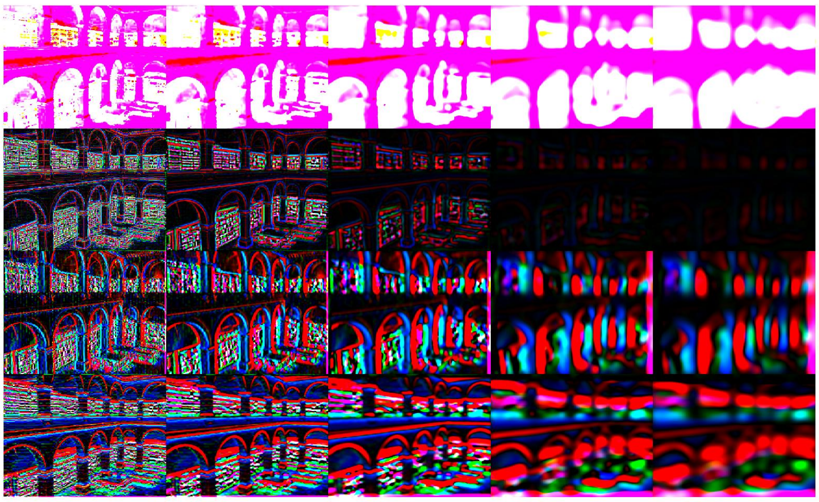


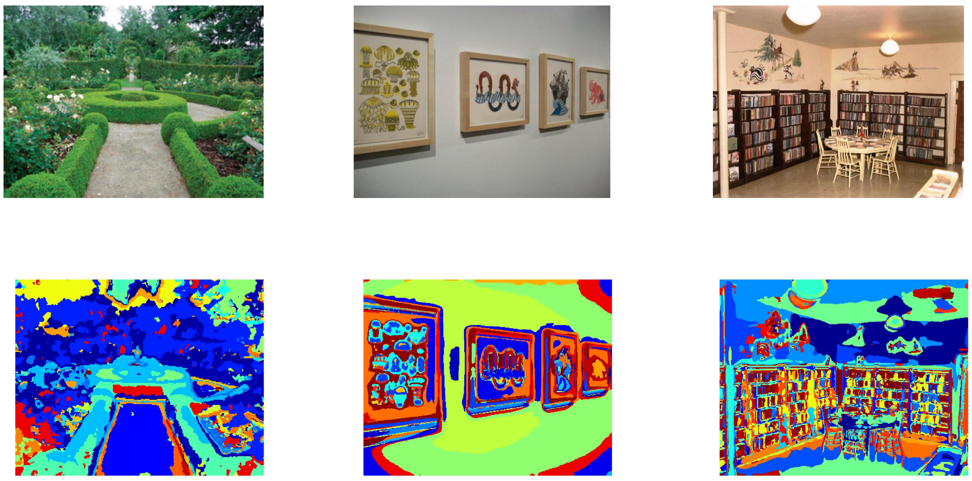
Figure 1: 4*5 filtered responses for an art gallery image

Q1.2: Creating Visual Words

Please see the code submitted on blackboard.

Q1.3: Computing Visual Words

I chose three images from class garden,art gallery and library. I calculated the distance between each filtered pixel and the dictionary and a minimum distance means that they are most similar so I could put this "word" into wordmap.At first the output was blurry and some of the wordmap is not precise enough for reader to tell which kind it belongs. I thought it might because of the value of K were small so that there was mot enough classes to refer to. So I changed k from 100 to 250 and got the images below. As you can see in the images, same "texture"in RGB images yield block of same color in wordmap, which means they belong to the same "word" in dictionary.



(a) Garden (b) Art Gallery (c) Library

Figure 3: Visual words over images

Q2.1: Extract Features

Please see the code submitted on blackboard.

Q2.2: Multi-resolution: Spacial Pyramid Matching

Please see the code submitted on blackboard.

Q2.3: Comparing Images

Please see the code submitted on blackboard.

Q2.4: Building a Model of the Visual World

Please see the code submitted on blackboard.

Q2.5: Quantitative Evaluation

With the change of alpha and k, I got groups of data in confusion matrix, accuracy and the system run-time. As shown in the tabulation, with a higher k means more "sorts" in dictionary and more alpha means more pixels were randomly picked. Theoretically higher k and alpha would have a higher accuracy in average but I think some exception might happen because the pixels were picked randomly. The confusion matrix C did have larger diagonal elements while others are relatively small which means most of test images were corrected guessed. But from all these confusion matrices I can see some classes are always incorrectly classified. See more in Q2.6 please.

k	alpha	Accuracy	System run-time (2-core)	Confusion matrix C
100	100	56.88%	42 min	$\begin{bmatrix} 7 & 0 & 1 & 4 & 6 & 0 & 0 & 2 \\ 0 & 10 & 0 & 3 & 5 & 0 & 1 & 1 \\ 0 & 0 & 18 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 17 & 0 & 0 & 0 & 1 \\ 2 & 3 & 0 & 3 & 10 & 1 & 0 & 1 \\ 0 & 0 & 2 & 0 & 2 & 11 & 4 & 1 \\ 0 & 0 & 0 & 0 & 2 & 3 & 15 & 0 \\ 1 & 2 & 4 & 2 & 2 & 2 & 4 & 3 \end{bmatrix}$
150	150	60.62%	50 min	$\begin{bmatrix} 8 & 1 & 0 & 3 & 7 & 0 & 0 & 1 \\ 0 & 10 & 0 & 2 & 6 & 0 & 0 & 2 \\ 0 & 0 & 19 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 19 & 0 & 0 & 0 & 1 \\ 3 & 5 & 0 & 2 & 9 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 & 10 & 4 & 2 \\ 0 & 1 & 0 & 0 & 1 & 3 & 15 & 0 \\ 2 & 3 & 2 & 2 & 1 & 1 & 2 & 7 \end{bmatrix}$
250	150	56.25%	67 min	$\begin{bmatrix} 6 & 0 & 0 & 6 & 6 & 0 & 1 & 1 \\ 1 & 8 & 0 & 3 & 5 & 0 & 2 & 1 \\ 0 & 0 & 19 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 17 & 1 & 0 & 0 & 0 \\ 4 & 3 & 0 & 2 & 10 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 & 9 & 5 & 2 \\ 0 & 1 & 0 & 0 & 0 & 3 & 16 & 0 \\ 2 & 2 & 2 & 2 & 2 & 2 & 3 & 5 \end{bmatrix}$

200	200	58.13%	58 min	$\begin{bmatrix} 8 & 0 & 0 & 5 & 5 & 0 & 0 & 2 \\ 1 & 9 & 0 & 2 & 7 & 1 & 0 & 0 \\ 0 & 0 & 19 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 18 & 0 & 0 & 0 & 1 \\ 4 & 5 & 0 & 1 & 9 & 1 & 0 & 0 \\ 1 & 1 & 3 & 0 & 1 & 8 & 6 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 & 17 & 0 \\ 1 & 3 & 3 & 2 & 2 & 1 & 3 & 5 \end{bmatrix}$	
250	200	56.28%	64 min	$\begin{bmatrix} 8 & 0 & 1 & 6 & 4 & 0 & 0 & 1 \\ 0 & 5 & 0 & 4 & 7 & 1 & 2 & 1 \\ 0 & 0 & 19 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 17 & 1 & 0 & 0 & 0 \\ 3 & 5 & 0 & 1 & 11 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 10 & 5 & 2 \\ 0 & 0 & 0 & 1 & 1 & 1 & 16 & 1 \\ 1 & 2 & 4 & 2 & 2 & 2 & 3 & 4 \end{bmatrix}$	
300	300	59.38%	74 min	$\begin{bmatrix} 6 & 0 & 0 & 5 & 6 & 0 & 1 & 2 \\ 2 & 8 & 0 & 4 & 5 & 1 & 0 & 0 \\ 0 & 0 & 19 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 17 & 0 & 0 & 0 & 1 \\ 3 & 3 & 0 & 1 & 12 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 & 8 & 6 & 2 \\ 1 & 1 & 0 & 0 & 0 & 1 & 17 & 0 \\ 2 & 2 & 2 & 1 & 1 & 1 & 3 & 8 \end{bmatrix}$	

Q2.6: Find out the failed cases

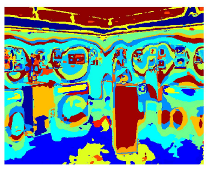
(I added some lines of code in the function evaluateRecognitionSyetem to show me the image numbers, correct labels and the wrong labels I got of all failed classes.)
From the matrices in the tabulation above, it can be seen that some class are always classified mostly correctly like the third and fourth row, the diagonal elements of them are big while others are small, even zero.These two rows are corresponding to class garden and ice skating. It's because the wordmap of these two kinds are special. Like the class garden which wordmap are always thickly dotted in different colour. manyWhile some other kinds always yield a wrong answer like the eighth row, which is corresponding to the class tennis court. Therefore, I count up the numbers of the most frequent failed classes in average and got the matrix below.

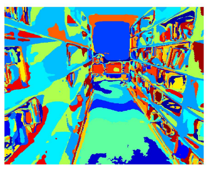
Class	Correctly classified	Most frequently classified
art gallery	34.17%	ice skating, library
computer room	40.00%	ice skating, library
garden	94.17%	mountain
ice skating	86.67%	computer room
library	50.83%	art gallery, computer room
mountain	47.50%	ocean
ocean	70.83%	mountain
tennis court	15.83%	All

Hereby are some comments with the incorrectly classified images and there wordmap. When I run the system it will show me the images numbers and the wrong label of the failed cases, I randomly picked one of each kind to illustrate.

Art gallery <-> Library

(a) RGB image of art gallery

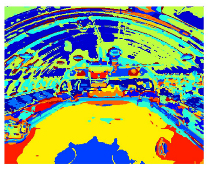

(b) wordmap of art gallery


(c) Librarywordmap sample

Computer room <-> Ice skating

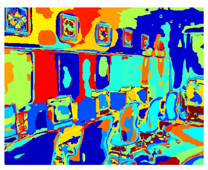
(a) RGB image of art computer room

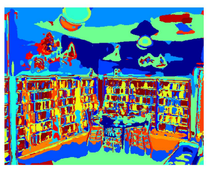
(b) Computer room wordmap


(c) Ice skating wordmap sample

Computer room <-> Library

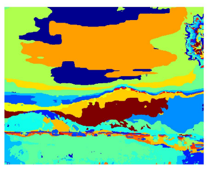
(a) RGB image of computer room

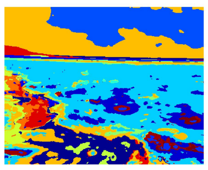

(b) Computer room wordmap


(c) Library wordmap sample

Mountain <-> Ocean

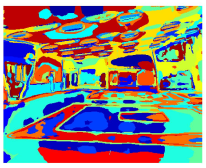
(a) RGB image of mountain

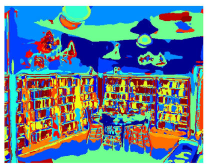

(b) Mountain wordmap


(c) Ocean wordmap sample

Tennis court <-> Library

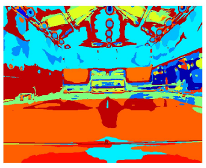
(a) RGB image of tennis court

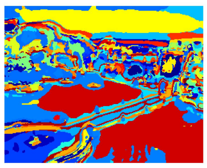

(b) Tennis court wordmap


(c) Library wordmap sample

Tennis court <-> Garden

(a) RGB image of tennis court


(b) Tennis court wordmap


(c) Garden wordmap sample

From the images and wordmaps above we can see that they were wrong labeled because their wordmaps have high similarity both in component and structure. And it means that they have similar texture. Take the art gallery and library for example. Their wordmaps were both dominated by light blue and some red or yellow blocks which means they contained the same "words" in dictionary. Thus when the system try to label them it would find their wordmaps' distance from the wrong kinds' feature from train dataset were even smaller than to their right kind. So they'd be labeled to the wrong kind.

It is normal because as we can see in the RGB images they were already alike, like ocean and sky, work of art and books or ground and wall. A similar image texture leads to similar wordmap, and thus lead to a same label. There might be some methods to fix this problem, please see more in Q2.7.

Q2.7: Improving performance

As what have been discussed above, some certain kinds are always difficult to label. The output of changing of k or alpha has been shown above and no big difference in the accuracy were found but a lot more time were consumed. So I tried some other ways.

Change the number of layer
I changed the number of layer from 3 to 4 to observe if the accuracy can be improved.It turns out that the accuracy didn't change much, just from 55% to 56.25%. In the SPM a bigger bin means less edge effect and more bin means more "new things" were counted. And bigger bin means less bins while more bins means smaller bin. It just like they're mutual exclusion. So I think if change the weight method of different layers to maximize the number of new things and minimize the edge effect might yield a better results.

Use different filter
In this assignment we use gaussian,log,dx and dy filter which orderly smooth the image, get spacial derivative, grasp texture information horizontally and vertically. Some more complicated operator like canny operator can be used to grasp a finer texture of image.