Andrew ID: xinghaoz
Name: Xinghao Zhou

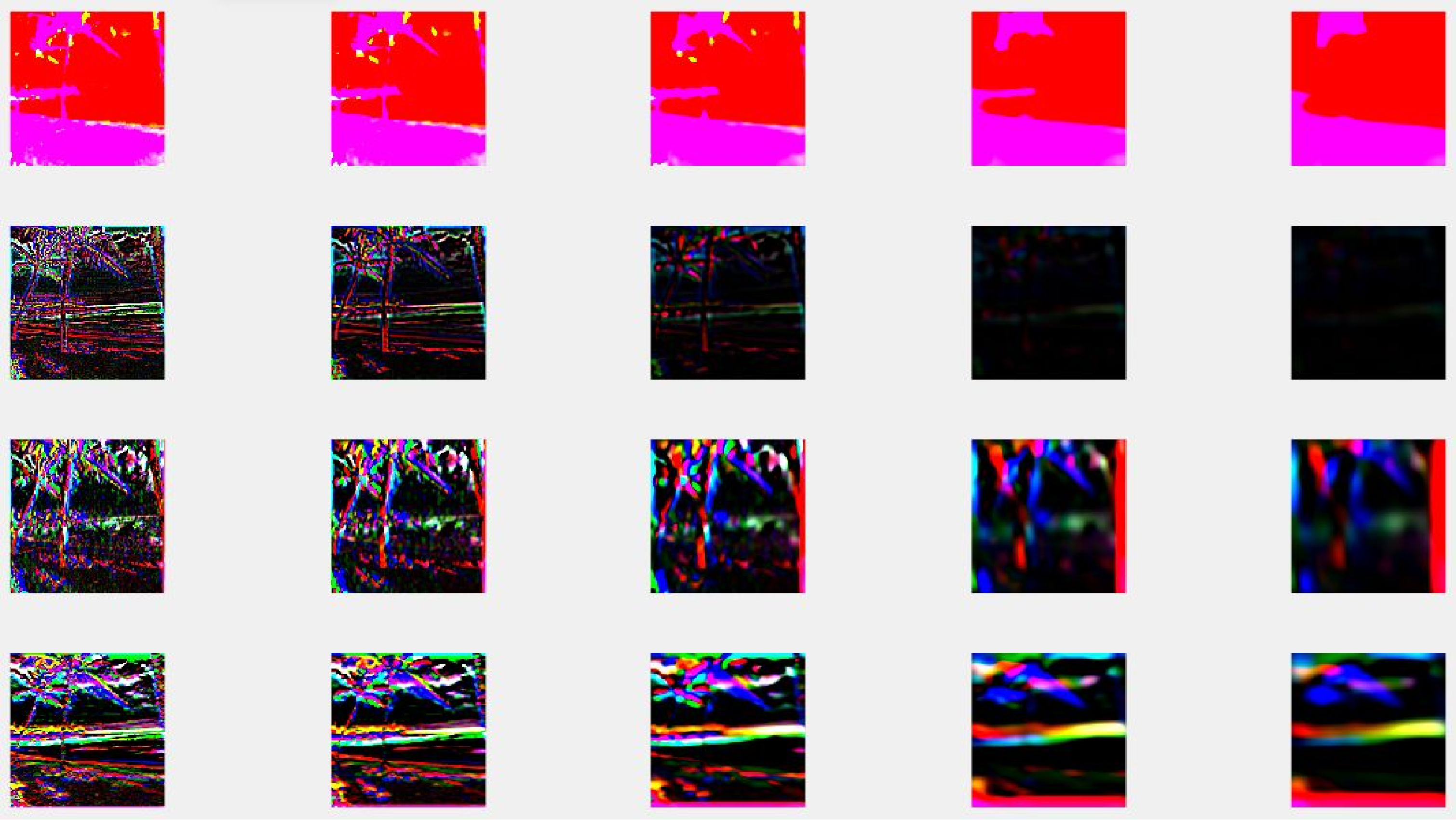
Q1.0:
Row 1 (filter bank 1 ~ 5) are Gaussian low pass filters. They pick up the regions in an image that are smooth.

Row 2 (filter bank 6 ~ 10) are Laplacian of Gaussian filters. They pick up the blobs in an image that differ in properties. Or say it highlights regions of rapid intensity change and is therefore used for edge detection.

Row 3 (filter bank 11 ~ 15) are Gaussian low pass filters derivatives at x direction. They pick up the regions in an image that have intensity change in x direction.

Row 4 (filter bank 16 ~ 20) are Gaussian low pass filters derivatives at y direction. They pick up the regions in an image that have intensity change in y direction.

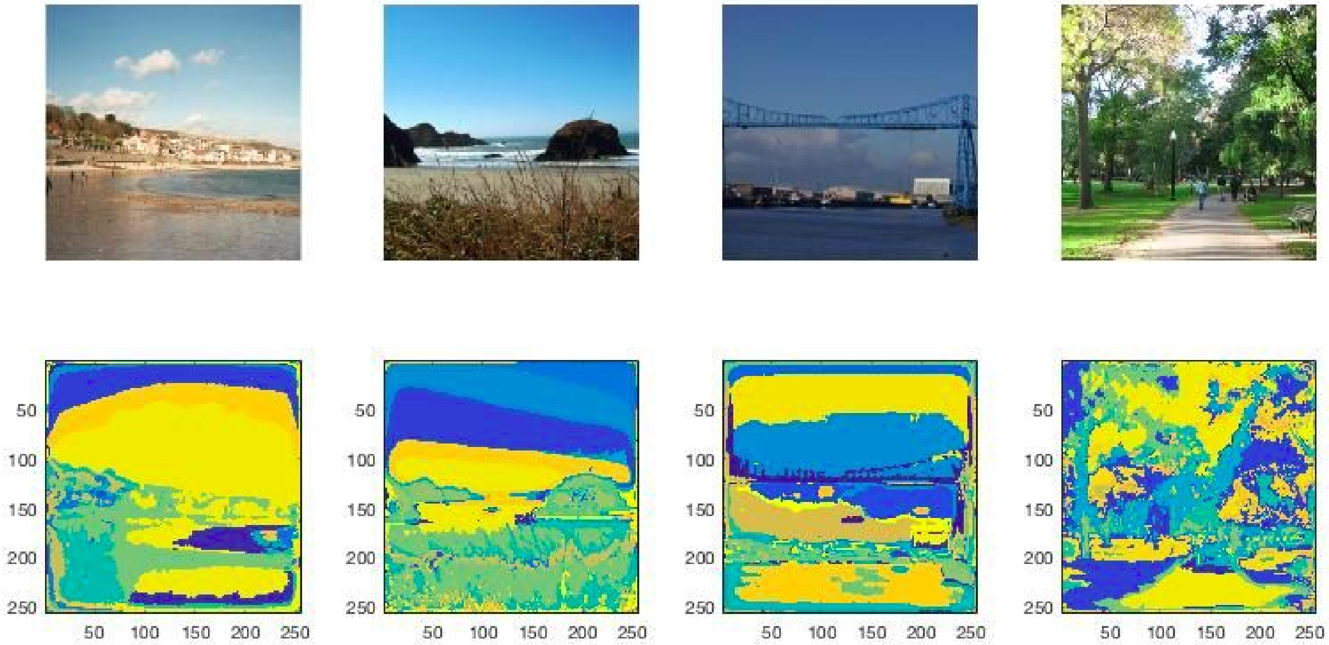
From a certain row, the scale is getting larger from left to right. Thus the one in the left contains more details but might be sensitive to the noise. The one in the right is less sensitive to the noise but it might lose precision.



Q1.3

The word map of the images tells us the important parts in the image. The more hotter (red) the color is, the more import this region is. The colder (blue) the color is, the less important this region is. Here the importance means this region plays a important role in the classification.

However, because of the version of the MatLab, there is no red region in my word map. Instead, the red region is replaced by dark blue.


50
100
150
200
250
50
100
150
200
250
50
100
150
200
250
50
100
150
200
250
50
100
150
200
250
50
100
150
200
250
50
100
150
200
250
50
100
150
200
250

Q2.5:

```
conf =

    16     0     0     1     0     0     0     0
     0    12     1     1     3     1     0     1
     0     2     9     1     0     3     0     6
     0     1     2    13     1     0     0     0
     0     3     1     1    15     2     4     1
     1     0     1     0     0    12     1     4
     3     2     4     1     1     1    14     2
     0     0     2     2     0     1     1     6

97 correct. Accuracy: 0.606250
```

Q2.6

```
../dat/park/sun_bkhifroddropoxyu.jpg
../dat/park/sun_bjmcizwdiixltjaw.jpg
../dat/parking_lot/sun_ashvdbzjuonoywgn.jpg
../dat/parking_lot/sun_aspaqbbliyqapggy.jpg
../dat/waterfall/sun_btpaxygvjiqfrsvs.jpg
../dat/waterfall/sun_bnztrbwcwwkkrnum.jpg
```

Those images are difficult to classify because they contain lots of blogs similar to the images of other categories. Their spatial pyramid have the similar pattern and thus make it hard to tell from other categories.

2.7

In the function getImageFeaturesSPM() of 2.2, instead of computing each layer one by one, I took the hint in the write up into consideration. Because for all of the three layers I need to traverse the same matrix, instead of traversing the matrix for 3 times, I can figure out the index of the three layers in the output [h] every time I visit a point. Thus I can build the histogram by only traverse the image once.

Furthermore, I tried several combination of K and alpha, and I found that the larger alpha is, the more accurate it will be because alpha denotes the number of points we are going to sample in the response, thus alpha has the positive relationship to the performance.

K, however, denotes the number of cluster in the k-means. It's hard to tell how many categories in a images, and each image is even different from each other. Thus the choice is K is tricky.

After several trials, the accuracy I have got range from very low (0.1437) to 0.606250. And I found K = 250 leads to a better performance.