

# 16-720B Homework 1 Write-up

Gu, Qiao

September 25, 2019

**Please note that:** Before running `main.py`, please make sure `data` folder are there in the same level with `code`, and please also create two empty `results` and `temp` in the same level with `code`.

## Q1.1.1

**Gaussian filters** can smooth images and thus remove noise and details. Therefore, they can capture the “averaged” intensity of images over a scale.

**Laplacian of Gaussian** yields high responses at points where there are rapid changes in intensity. Specifically, Laplacian of Gaussian can capture peaks of intensity.

**Derivatives of Gaussian in x/y direction** gives high responses at edges in x and y direction respectively, and thus they capture the edge information of images.

**Multiple scales** are needed because the information mentioned above can happen at any scale, and multiple filters of different scales can capture them in different scales.

### Q1.1.2

Please see the Figure 1 for the filter responses

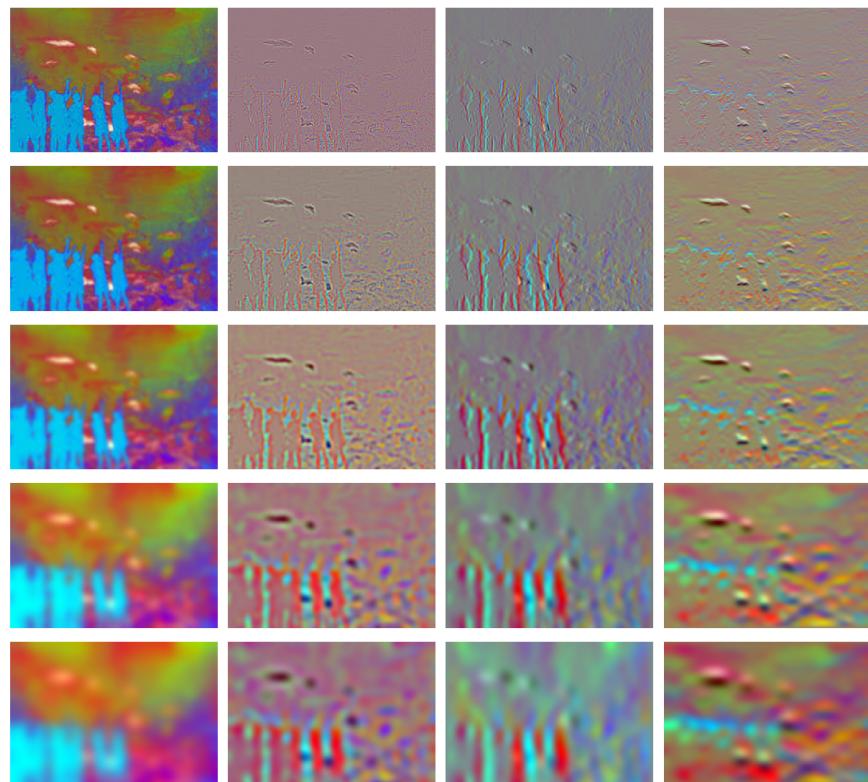


Figure 1: Filter responses on the image  
`aquarium/sun_aztvjgubyrgvirup.jpg`.

### Q1.3

Please see the Figure. 2 for the images and their wordmaps.

Yes these wordmaps indeed make sense to me. We can see that within one object, the color tends to be the same, which reflects the homogeneous texture of its appearance captured by Gaussian filters. When it goes to the boundary, there are one or more color strips on the borderline, and this reflects the different types of edges captured by derivatives and Laplacian of Gaussian filters.

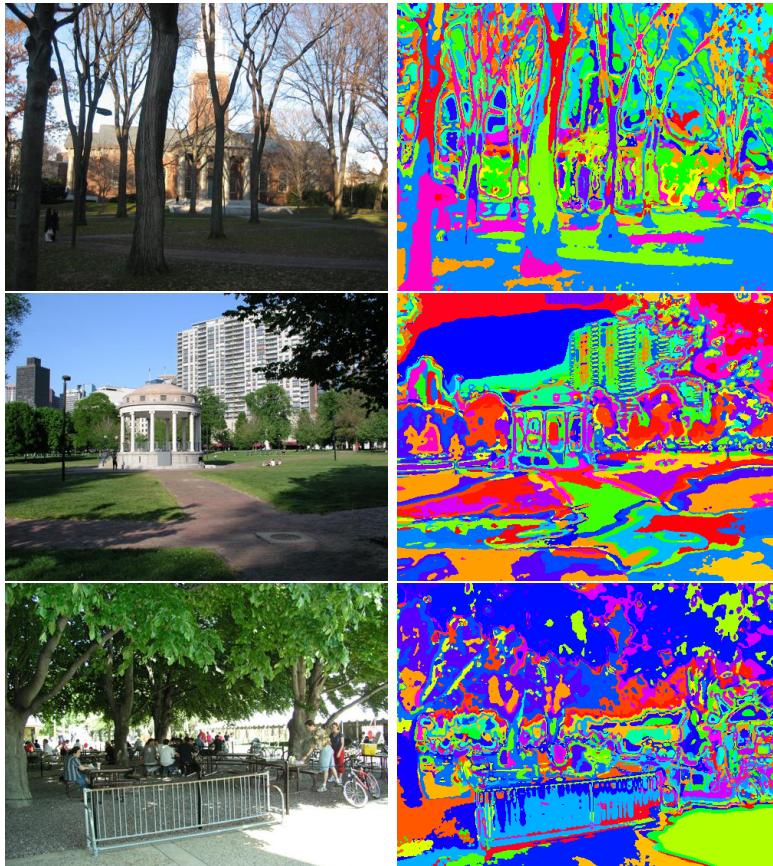


Figure 2: Images and their wordmaps.

## Q2.5

The overall accuracy is 0.64375, and the confusion matrix is as follows:

```
[[14.  0.  0.  0.  0.  0.  0.  0.]  
 [ 0. 14.  0.  1.  0.  1.  2.  0.]  
 [ 0.  0. 15.  4.  3.  0.  0.  3.]  
 [ 1.  2.  1. 17.  0.  0.  1.  4.]  
 [ 1.  1.  0.  0. 10.  1.  0.  0.]  
 [ 0.  2.  0.  1.  8. 12.  1.  0.]  
 [ 0.  5.  0.  0.  2.  3. 11.  0.]  
 [ 0.  2.  3.  2.  1.  1.  0. 10.]]
```

## Q2.6

Since our method for classification method is essentially 1 Nearest Neighbor, it is most reasonable for us to analyze the failure cases with their nearest neighbor that leads to such decisions. One example failure case together with its nearest neighbor is shown in Figure. 3. As we can see in the figure, these two images are much like each other, in terms of both scene structure and color on the RGB images and the feature distribution on the colormaps.

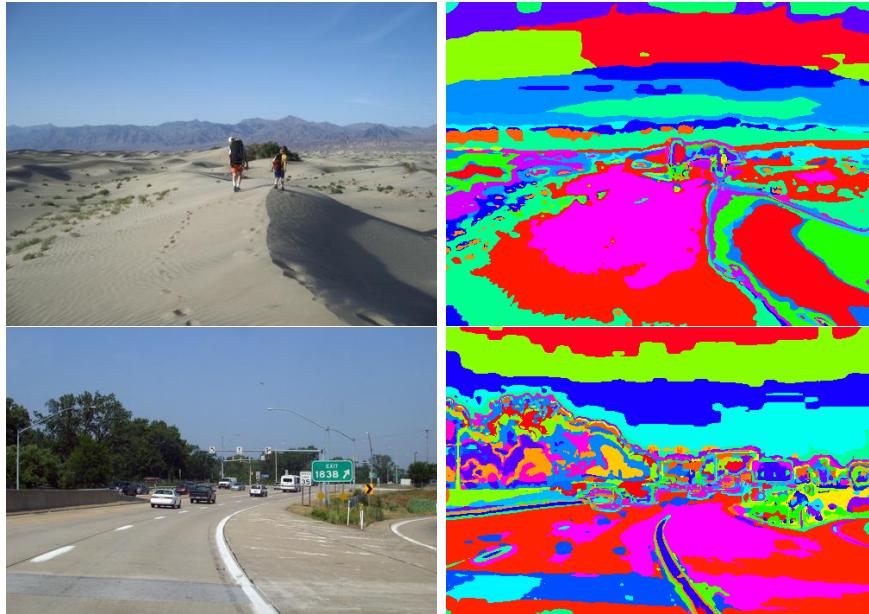


Figure 3: One failure example (the first row) that is actually in “desert” class but is misclassified to “highway” class, together its nearest neighbor (the second row) that causes such wrong decision. The left column shows their RGB image and the right column shows their wordmap.

### Q3.2

The overall accuracy is 0.975, and the confusion matrix is as follows:

```
[[14.  0.  0.  0.  0.  0.  0.  0.]  
 [ 0. 17.  0.  0.  0.  0.  0.  1.]  
 [ 0.  0. 24.  0.  0.  0.  0.  1.]  
 [ 0.  0.  0. 26.  0.  0.  0.  0.]  
 [ 0.  0.  0.  0. 12.  1.  0.  0.]  
 [ 0.  0.  0.  0.  1. 23.  0.  0.]  
 [ 0.  0.  0.  0.  0.  0. 21.  0.]  
 [ 0.  0.  0.  0.  0.  0.  0. 19.]]
```

We can see that the result accuracy and confusion matrix are much better than the those given by the classical BoW. I believe this is because since deep neural network (DNN) has deeper structure of convolutional layers, it has a much stronger capability of capturing high-level feature: DNN can synthesize the filter responses from different positions using another structured filter ratherthan simply calculate the histogram of filter responses of different scales and locations.