16-720 Homework 1

Chendi Lin

September 25, 2018

Problem 1.1.1. What properties do each of the filter functions pick up? You should group the filters into broad categories (e.g. all the Gaussians). Also, why do we need multiple scales of filter responses?

Solution . Gaussian filter is a low-pass filter that remove the high-frequency noises. It can smooth and blur the figures and thus we do not need to worry about the unnecessary details.

Laplacian of Gaussian filter detects the sudden change of the figures. Thus, it picks up the edges in the graphs.

The derivative of Gaussian in the x direction detects the change in the x-axis. Therefore, it picks up the vertical edges in the graphs.

The derivative of Gaussian in the y direction detects the change in the y-axis. Therefore, it picks up the horizontal edges in the graphs.

The scales determine the amounts of details in the responses. The smaller the scale is, the more details are included in the responses, and more sharp the filter responses images are.

Problem 1.1.2. Extract filter responses.

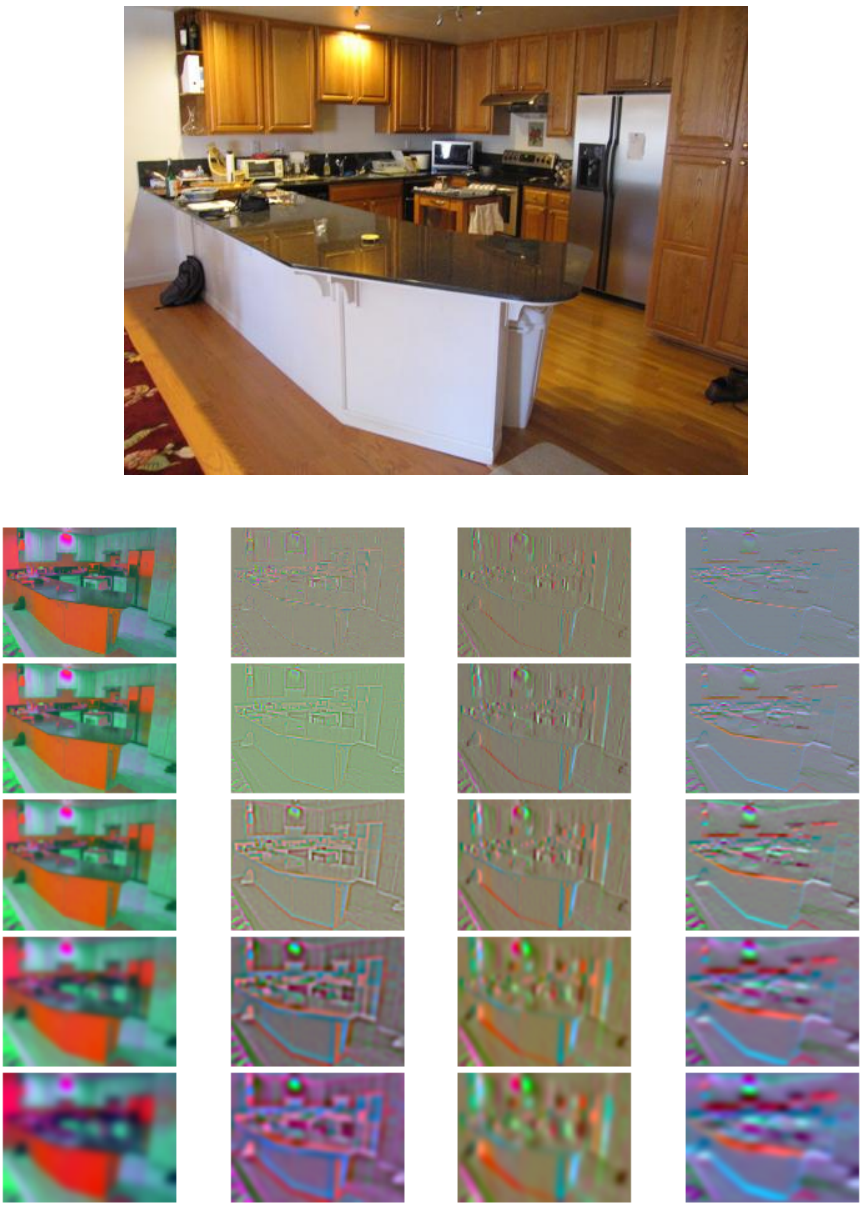

Figure 1: Twenty filter responses of a kitchen picture

Problem 1.3. Computing Visual Words

To generate the dictionary, K was chosen to be 200, and α was chosen to be 250. Several times of tuning in parameters have been done, but this pair worked appropriately when evaluating the recognition system. Images of kitchen, baseball field, and windmill and their wordmaps are attached below. It can be told from the examples below that, for well-structured images, like the kitchen and windmill images, the wordmaps indicate the features more clearly. However, for the images like baseball field, less characteristics can be differentiated from the wordmap.

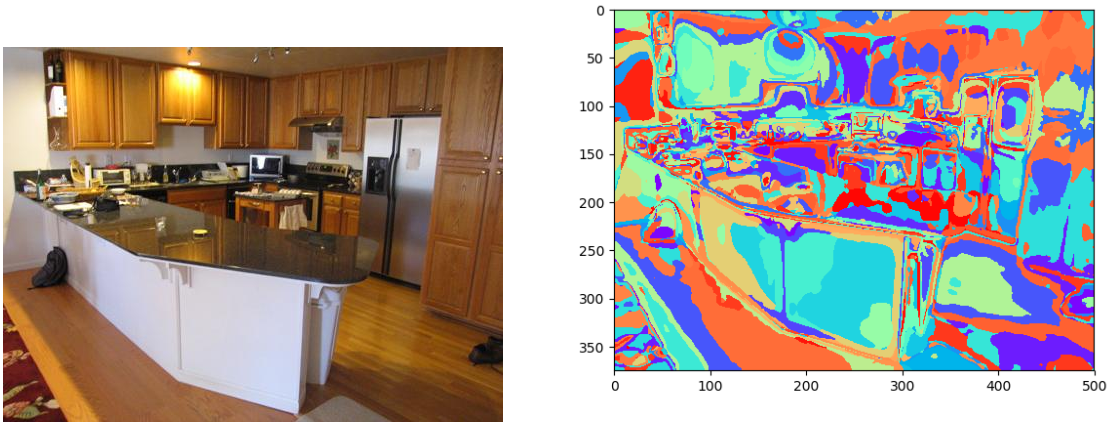

Figure 2: Kitchen and Wordmap

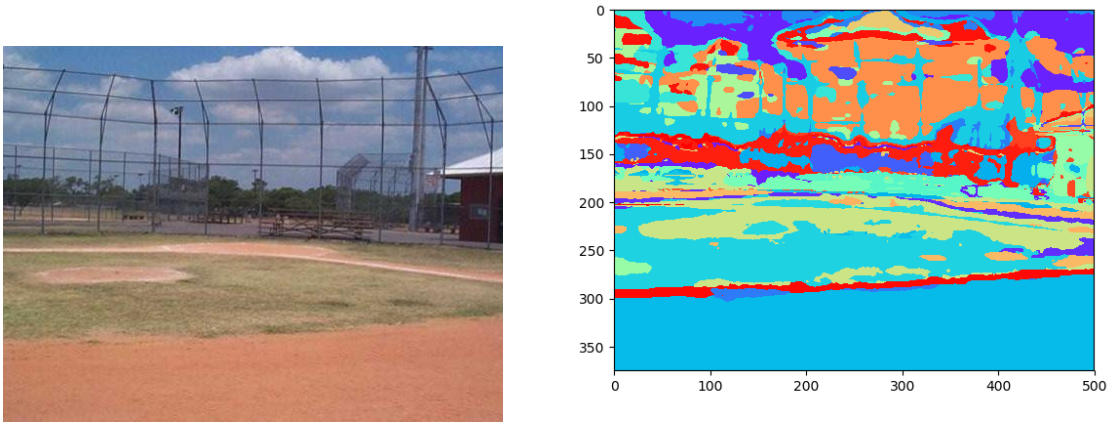

Figure 3: Baseball Field and Wordmap

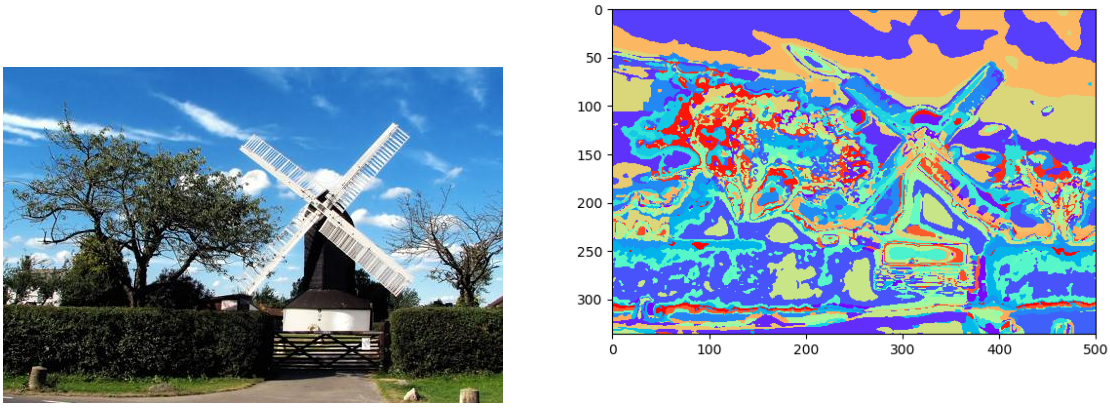

Figure 4: Windmill and Wordmap

Problem 2.5. Quantitative Evaluation of BOW Visual Recognition

Solution . The implementations are included in the codes. The confusion matrix is:

$$\begin{bmatrix} 10 & 0 & 3 & 1 & 2 & 1 & 3 & 0 \\ 0 & 9 & 3 & 0 & 1 & 0 & 0 & 7 \\ 1 & 1 & 14 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 14 & 0 & 0 & 1 & 5 \\ 1 & 0 & 2 & 0 & 11 & 2 & 3 & 1 \\ 4 & 0 & 1 & 0 & 7 & 7 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 2 & 13 & 2 \\ 0 & 2 & 1 & 6 & 0 & 0 & 1 & 10 \end{bmatrix}$$

The accuracy is 55%

Problem 2.6. Find the failed cases.

Solution . From the confusion matrix it can be told that, class "laundromat", "baseball field", "windmill" are the top three classes with the highest error probability. Three examples that were classified incorrectly are shown below.

"Laundromat" is the class with the largest error percentage. Because of the similarity, it is easy to be confused with the class "Kitchen". From Fig. 5 we can see that, though the features were nicely and neatly extracted, since the setup was close to that of class "kitchen", this one was still classified into "kitchen" incorrectly.

"Baseball field" is the class with the second largest error percentage, for which most images were classified as "windmill" mistakenly. In Fig. 6 we can see that, since the image is so empty barely any features were extracted, which made the visual recognition complicated.

"Windmill" is the class with the third largest error percentage. Around 50% of images were classified correctly, which was around the average level of the whole BOW visual recognition system. Figure 7 is an example that was classified falsely. It is suspected that it was because the characteristic of windmills were hidden behind the castle, which made the recognition harder.


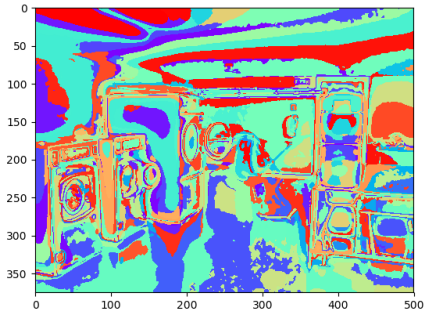

Figure 5: Wrong Laundromat Image and Wordmap


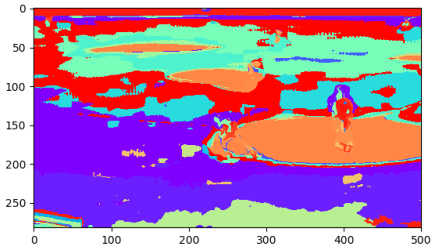

Figure 6: Wrong Baseball Field Image and Wordmap


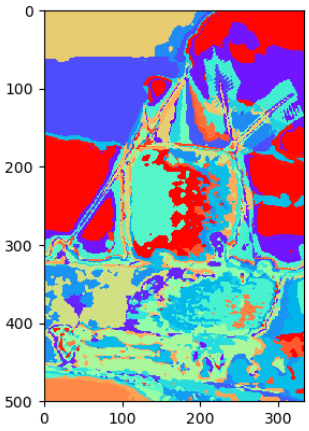

Figure 7: Wrong Windmill Image and Wordmap

Solution 3.1. To verify that the implementations are correct, an arbitrary image was inputted and features were extracted using both "network_layer.extract_deep_feature", and "pytorch". The 2-norm of the difference of the two features is 39.09, which is relatively small considering that it is a vector with 4096 entries. Also, two random images were loaded into the evaluation system using "network_layer.extract_deep_feature", and they are correctly classified, which proved that the implementations work well.

Problem 3.2. Quantitative Evaluation of VGG Network

Solution . The implementations are include in the code.

The confusion matrix of the result of VGG nets is:

$$\begin{bmatrix} 19 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 19 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 19 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 19 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 18 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 19 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 & 0 & 16 \end{bmatrix}$$

The accuracy is 95.625%