16720 Computer Vision Assignment1

Yu Fang Chang, Robotics Institute
yufangc@andrew.cmu.edu
September 12, 2015

1 Representing the World with Visual Words

A1.0: Each row represents a type of filter with different scale. Following are properties of each filter with order.

1. *Gaussian*: A Gaussian filter is a low-pass filter, which surpresses high frequency components and thus results in blurring an image. It often applies in smoothing the image to reduce the noise and image details, as visualized in Fig. 1 (1)∼(3).

2. *Laplacian of Gaussian (LoG)*: The LoG filter computes the second-derivative of intensity in an image. It is a band-pass filter, which surpresses both high and low frequency components and responses in rapid change in intensity, and hense often applies in edge detection. If the input image is too noisy, the resulting image will contain many but meaningless small edges. We exploit LoG to extract the edge information that describe the object shapes in a scene. Fig. 1 (4)∼(6) are the image applying LoG filter with $\sigma = 1$ on L, a, b respecitvely.

3. *Derivative of Gaussian in X (DoG)*: A filter that extracts maximal derivative magnitude difference horizontally. It responses the intensity change in an image in x direction, as visualized in Fig. 1 (7)∼(9).

4. *Derivative of Y*: A filter that is similar to DoG in x direction discussed above but extracts vertical intensity difference, as visualized in Fig. 1 (10)∼(12).

A1.1: Extracting Filter Response

Implementation please refer to extractFilterResponses.m, and the response demostration refer to demo1_1.m.

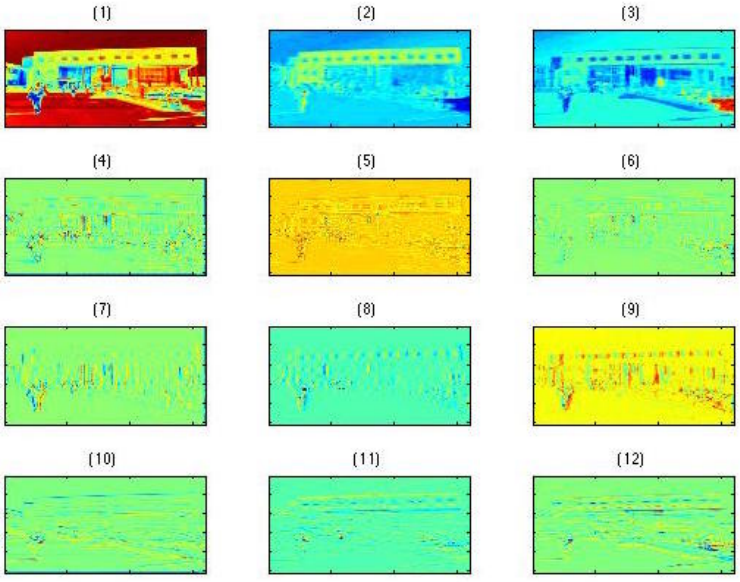

Figure 1: Filter responses for the image sun_abslhphpiejdjmpz.jpg with imagesc. Each column with $\sigma = 1$ on L,a,b channel respectively; **Gaussian:**(1)∼(3) , **LoG:**(4)∼(6) , **x-derivative:**(7)∼(9) , **y-derivative:**(10)∼(12)

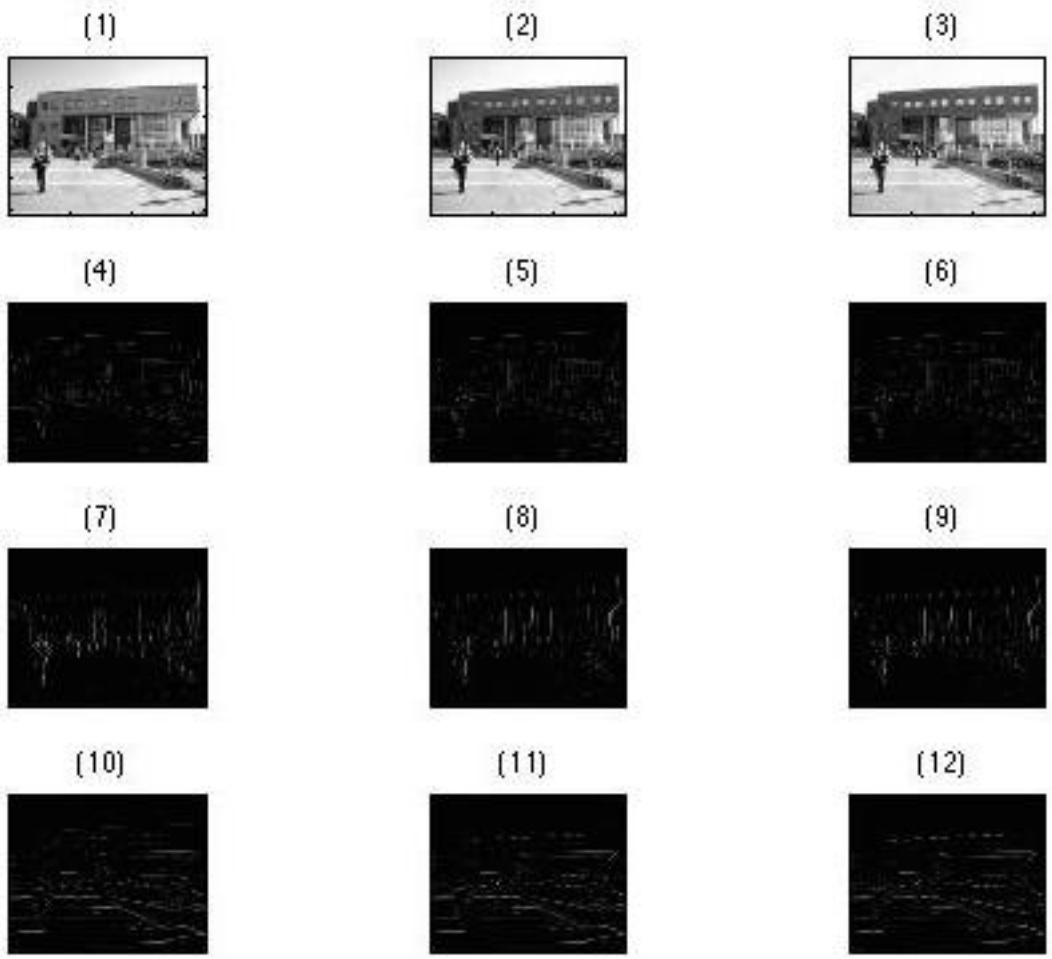

Figure 2: Filter responses for the image sun_abslhphpiejdjmpz.jpg with image in RGB channel. Each column with $\sigma = 1$ on R,G,B channel respectively; **Gaussian:**(1)∼(3) , **LoG:**(4)∼(6) , **x-derivative:**(7)∼(9) , **y-derivative:**(10)∼(12)

A1.2: Creating Visual Words

For implementation, please refer to getFilterBankAndDictionary.m and computeDictionary.m to produce a .mat file named dictionary.mat that contains the filter bank as well as the dictionary of visual words.

A1.3: Computing Visual Words

For mapping each pixel in the image to its closest word in the dictionary, please refer to getVisualWords.m. For a demo please run demo1_3.m. Fig.3 shows exemplar pairs of an input image and its resulting wordMap from each category.

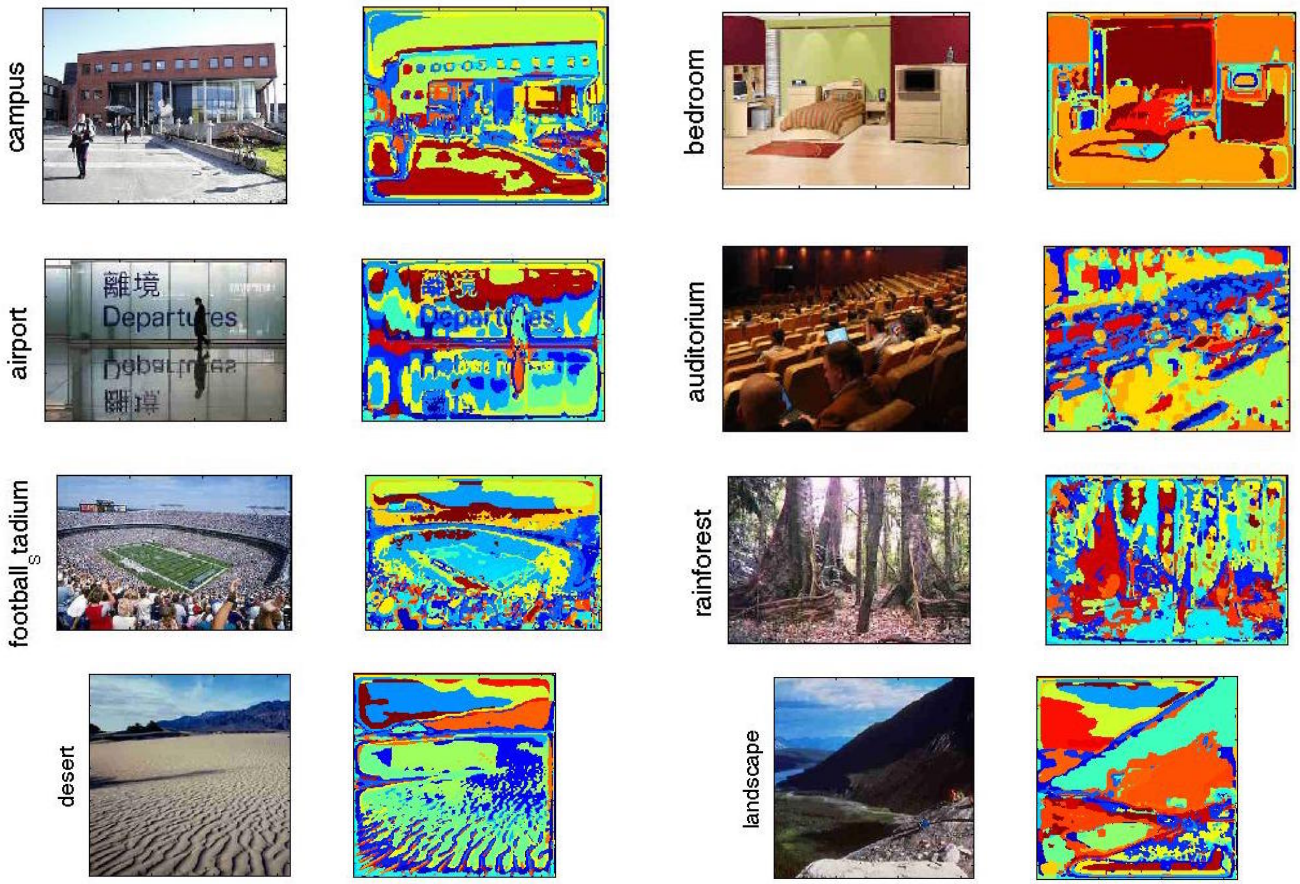

Figure 3: Above show different categories of an input image (left) and its resulting wordMap (right).

2 Building a Recognition System

A2.1: Extracting Features

getImageFeatures.m extracts the histogram of visual words within the given image.

A2.2: Multi-resolution: Spatial Pyramid Matching

Since Bag of Words discards information about the spatial structure of the image, we apply spatial pyramid matching to alleviate the problem. For efficiency, I computer the histograms of the finest layer (Level=2), and aggregated the finer ones to the coarser ones. getImageFeaturesSPM.m forms a multi-resolution representation of the given image.

A2.3: Comparing images

To find the nearest training image, we use the histogram intersection similarity and find the largest one. For implementation, please refoer to distanceToSet.m and demo2_3.m for visualize the nearest training image.

A2.4: Building A Model of the Visual World

Plese refer to buildRecognitionSystem.m for saving vision.mat and guessImage.m for getting a prediction based on the histogram.

A2.5: Quantitative Evaluation

To quantify the accuracy, we compute a confusion matrix C: given a classification problem, the entry C(i,j) of a confusion matrix counts the number of instances of class i that were predicted as class j. For implementation, please refer to evaluateRecognitionSystem.m.

The accuracy depends greatly on how we create the filter bank, how large we set the dictionary and how many random pixel we extract from filter response. As we expected, the baseline performance is poor due to naive feature representation and classification method. Following is the confusion matrix of the baseline approach with dictionary size=100, $\alpha = 50$ random pixels. Accuracy = 50.62%

$$C = \begin{pmatrix} 12 & 1 & 0 & 2 & 0 & 0 & 2 & 3 \\ 5 & 11 & 2 & 0 & 0 & 0 & 2 & 0 \\ 6 & 3 & 9 & 1 & 1 & 0 & 0 & 0 \\ 3 & 2 & 0 & 4 & 2 & 1 & 7 & 1 \\ 0 & 1 & 2 & 3 & 10 & 1 & 3 & 0 \\ 5 & 0 & 0 & 4 & 1 & 6 & 3 & 1 \\ 2 & 1 & 0 & 1 & 3 & 0 & 12 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 17 \end{pmatrix}$$

A2.6: Find out the failed cases

From above confusion matrix and accuracy for each category:

label	1	2	3	4	5	6	7	8
category	airport	auditorium	bedroom	campus	desert	football stadium	landscape	rainforest
accuracy	0.6000	0.5500	0.4500	0.2000	0.5000	0.3000	0.6000	0.8500

We can observe that there are lots of campus wrongly predicted as landscape. From visualizing the mis-prediction, I found that most of them are buildings with blue sky and some trees and may result in classifing as landscape. Additionally, we can found that if there are crowds of people in the picture, it might be hard to predict since football stadium, airport and auditorium all have some features on crowds of people. Fig.4 shows several examples that were wrongly classified by the baseline approach.


(a) landscape(campus)


(b) landscape(campus)


(c) rainforest(campus)


(d) airport(football)

Figure 4: Failure cases: X (Y) indicates the image was classified as X but should be Y.

I have tested some different parameters in baseline approach.
(α: numbers of random pixel picked, K: generate a visual words dictionary with K words):

α	K	accuracy
50	100	50.62%
60	100	50.00%
100	100	48.75%
100	200	43.75%

References

[1] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):15821596, 2010.*

[2] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. *ICCV, 2005.*

[3] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR, volume 2, pages 21692178, 2006.*