EE 569: Homework #4

Issued: 03/04/2019 Due: 11:59PM, 03/19/2019

General Instructions:

1. Read Homework Guidelines and MATLAB Function Guidelines for the information about homework programming, write-up and submission. If you make any assumptions about a problem, please clearly state them in your report.
2. Do not copy sentences directly from any reference or online source. Written reports and source codes are subject to verification for any plagiarism. You need to understand the USC policy on academic integrity and penalties for cheating and plagiarism. These rules will be strictly enforced.

Problem 1: Texture Analysis (50%)

In this problem, you will implement texture analysis and segmentation algorithms based on the 5x5 Laws filters constructed by the tensor product of the five 1D kernels in Table 1.1:

Tabel 1.1. 1D Kernel for 5x5 Laws Filters

Name	Kernel
L5 (Level)	[1 4 6 4 1]
E5 (Edge)	[-1 -2 0 2 1]
S5 (Spot)	[-1 0 2 0 -1]
W5(Wave)	[-1 2 0 -2 1]
R5 (Ripple)	[1 -4 6 -4 1]

1(a) Texture Classification (Basic: 15%)

Twelve texture images [1], from texture1.raw to texture12.raw (size 128x128) are provided for the texture classification task in this part. Samples of these images are shown in Figure 1.1.

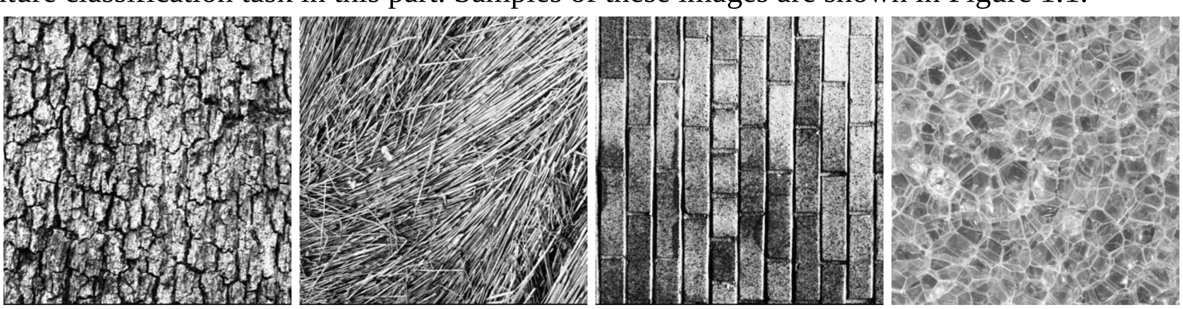

Figure 1.1: Four types of textures: bark, straw, brick, and bubbles

Please cluster them into four texture types with the following steps below to complete this problem.

1. Feature Extraction: Use the twenty-five 5x5 Laws Filters to extract feature vectors from each pixel in the image (use appropriate boundary extensions).
2. Feature Averaging: Average the feature vectors of all image pixels, leading to a 25-D feature vector for each image. Which feature dimension has the strongest discriminant power? Which has the weakest? Please justify your answer.

3. Reduce the feature dimension from 25 to 3 using the principal component analysis (PCA). Plot the reduced 3-D feature vector in the feature space. (You may use built-in C++/Matlab functions of PCA.)
4. Clustering: Use the K-means algorithm for image clustering based on the 25-D and 3-D obtained in Steps 2 and 3, respectively. Discuss the effectiveness of feature dimension reduction over K-means. Report your results and compare them with the reality (by eyes).

1(b) Texture Segmentation (Basic: 20%)

In this part, apply the twenty-five 5x5 Laws Filters to texture segmentation for the image shown in Figure 1.2 with the following steps.

1. Laws feature extraction: Apply all 25 Laws filters to the input image and get 25 gray–scale images.
2. Energy feature computation: Use a window approach to computer the energy measure for each pixel based on the results from Step 1. You may try a couple of different window sizes. After this step, you will obtain 25-D energy feature vector for each pixel.
3. Energy feature normalization: All kernels have a zero-mean except for $L5^T\, L5$. Actually, the feature extracted by the filter $L5^T\, L5$ is not a useful feature for texture classification and segmentation. Use its energy to normal all other features at each pixel.
4. Segmentation: Use the k–means algorithm to perform segmentation on the composite texture images given in Figure 1.2 based on the 25-D energy feature vectors.

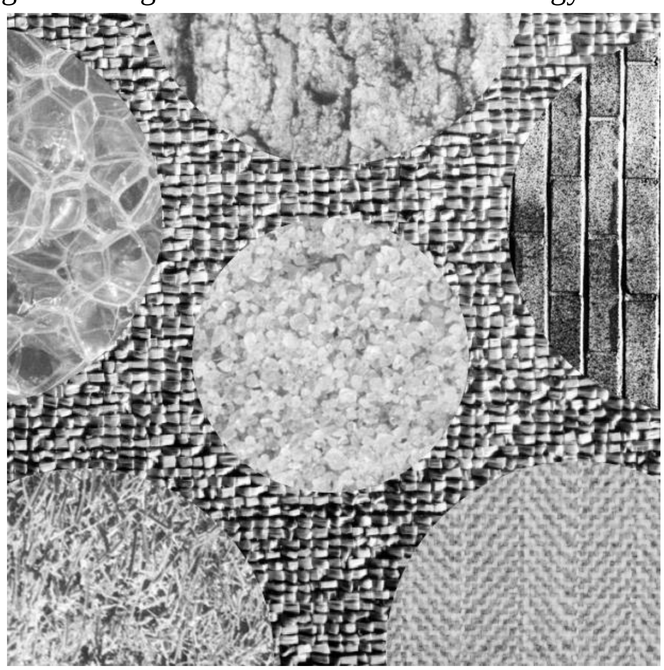

Figure 1.2: Composite texture image (comb.raw)

In order to denote segmented regions, if there are K textures in the image, your output image will be of K gray levels, with each level represents one type of texture. For example, there are 7 types of texture in Figure 1.2 you can use 7 gray levels (0, 42, 84, 126, 168, 210, 255) to do denotation.

1(c) Advanced Texture Segmentation Techniques (Advanced 15%)

You may not get good texture segmentation results for the complicated image in Figure 1.2. Please develop various techniques to enhance your segmentation result. Several ideas are sketched below.

1. Adopt the PCA for feature reduction and, thus, cleaning.

2. Develop a post-processing technique to merge small holes.

3. Enhance the boundary of two adjacent regions by focusing on the texture properties in these two regions only.

Problem 2: Image Feature Extractor (50%)

Image feature extractors are useful for representing the image information in a low dimensional form.

2(a) SIFT (Basic: 20%)

In this problem, you are asked to read the original SIFT paper in [2] and answer the following questions.

I. From the paper abstract, the SIFT is robust to what geometric modifications?
II. How does SIFT achieves its robustness to each of them?
III. How does SIFT enhances its robustness to illumination change?
IV. What are the advantages that SIFT uses difference of Gaussians (DoG) instead of Laplacian of Gaussians (LoG)?
V. What is the SIFT's output vector size in its original paper?

2(b) Image Matching (Basic: 20%)

One implementation of SIFT is image retrieval. You can do nearest neighbor search in the searching database for the query image which is represented as a SIFT extracted feature vector.

Figure 2.1 River images

Use open source SIFT tool (OpenCV, VLFeat, etc.) to find the key-points in the two river images shown above. Pick the key-point with the largest scale in river image 1 and find its closest neighboring key-point in river image 2. Discuss your results, esp. the orientation of each key-points.

2(c) Bag of Words (Advanced: 10%)

You are given samples of zeros and ones from the MNIST [3] dataset as shown below

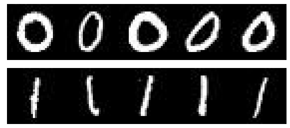

Figure 2.2 Images of zeros and ones

Use these images as your training dataset to form a codebook with bin size of 2 (two clusters for k-means clustering). Then extract the SIFT feature vector for the image of eight and show your histogram of the Bag of Words for this image. Discuss your observation.

Figure 2.3 Image of eight

Appendix:

Problem 1: Texture Analysis

texture1~12.raw	128x128	8-bit	gray
comb. raw	510x510	8-bit	gray

Problem 2: Image Matching

river1.raw	1024x768x3	24-bit	color
river2.raw	1024x768x3	24-bit	color
zero_1~5.raw	28x28	8-bit	gray
one_1~5.raw	28x28	8-bit	gray
eight. raw	28x28	8-bit	gray

Reference:

[1] http://sipi.usc.edu/database/database.php?volume=textures&image=19#top

[2] Lowe, David G. "Object recognition from local scale-invariant features." iccv. Ieee, 1999.

[3] http://yann.lecun.com/exdb/mnist/