

**EE 569: Homework #4**

Issued: 03/04/2020 Due: 11:59PM, 03/22/2020

**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. 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 and Segmentation (60%)**

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.

**Table 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]

**(a) Texture Classification --- Feature Extraction (Basic: 15%)**

48 images of four types textures are given for the texture classification task. They are split into two sets, 36 training samples and 12 testing samples. The ground truth labels of the 36 training samples are known, while the testing samples' categories are waiting for you to explore. Samples of these images are shown in Fig. 1.

**Figure 1:** Four types of textures: rice, grass, brick, blanket

Please follow steps below to extract features for all texture images provided and do analysis.

1. **Feature Extraction:** Use the twenty-five 5x5 Laws Filters in Table 1 to extract feature vectors from 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. Average feature vectors from pairs such as  $L5E5/E5L5$  to get a 15-D feature vector. Which feature dimension has the strongest discriminant power? Which has the weakest? Please justify your answer.
3. **Feature Reduction:** Reduce the feature dimension from 15 to 3 using the principle component analysis (PCA). Plot the reduced 3-D feature vector in the feature space. (Note that you should do PCA on train and test separately.)

Built-in PCA function can be used. For those who implement PCA correctly can obtain 5% bonus. Put it in your report!

**(b) Advanced Texture Classification --- Classifier Explore (Basic: 15%)**

Based on the 15-D and 3-D feature vectors obtained above, let's do both unsupervised and supervised learning. Please follow steps below.

1. **Unsupervised:** K-means clustering is kind of unsupervised classifier which categorize the textures without the help of ground truth labels. Use the K-means algorithm for test image clustering based on the 15-D and 3-D feature. Discuss the effectiveness of feature dimension reduction over K-means. Report and compare your results with observations (by eyes), give error rate.
2. **Supervised:** Use the 3-D feature of training images to train Random Forest (RF) and Support Vector Machine (SVM) respectively. Then predict the test set labels and give error rate. Compare the two kinds of classification.

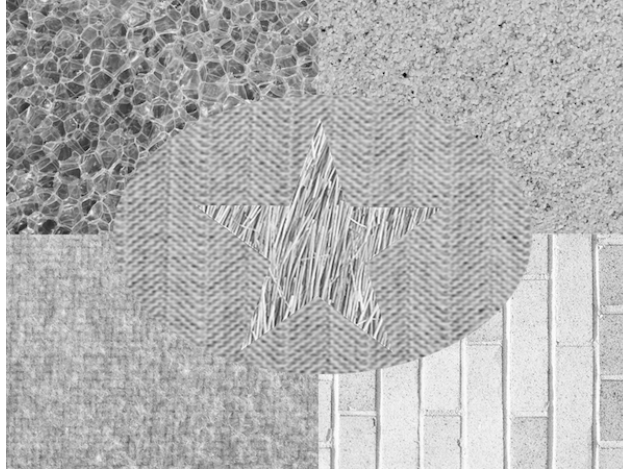
Built-in K-means function, RF and SVM can be used. For those who implement K-means algorithm correctly can obtain 5% bonus. Show your work!

**(c) Texture Segmentation (Basic: 20%)**

Segment the textures in Fig. 2 by applying the 15 5x5 Laws Filters. Please follow steps below to complete this problem.

1. **Feature extraction:** Apply Laws filters to the input image and get 15 gray-scale images.
2. **Energy feature computation:** Use a window approach to compute 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 15-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 Fig. 2 based on the 14-D energy feature vectors.

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, you can use (0, 51, 102, 153, 204, 255) to denote six segmented regions in the output for six textures.



**Figure 2:** Composite texture images

**(d) Advanced Texture Segmentation (Advanced: 10%)**

You may not get good segmentation results for the complicated texture mosaic image in Fig. 2. Please develop some techniques to improve your segmentation result. Several ideas are sketched below.

1. Adopt the PCA for feature reduction. Use the dimension reduced features to do texture segmentation of Fig. 2.
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.

Brainstorms are welcome!

**Problem 2: Image Feature Extractors (40%)**

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

**(a) Salient Point Descriptor (Basic: 10%)**

SIFT are effective tools to extract salient points in an image. Read the paper in [1] and answer the following questions.

1. From the paper abstract, the SIFT is robust to what geometric modifications?
2. How does SIFT achieves its robustness to each of them?
3. How does SIFT enhances its robustness to illumination change?
4. What the advantages that SIFT uses Difference of Gaussians (DoG) instead of Laplacian of Gaussians (LoG)?
5. What is the SIFT's output vector size in its original paper?

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

You can apply SIFT to object matching. Extract and show SIFT features.

1. Find key-points of the two Husky images in Fig. 3. Pick the key-point with the largest scale in Husky\_3 and find its closest neighboring key-point in Husky\_1. You can do nearest neighbor search in the searching database for the query image which is represented as a SIFT extracted feature vector. Discuss your results, esp. the orientation of each key-point.
2. Show the corresponding SIFT pairs between the Husky\_1 and Husky\_3 in Fig. 3. The matching may not work well between different objects and against the same object but with a large viewing angle difference. Perform the same job with the following three image pairs: 1) Husky\_3 and Husky\_2, 2) Husky\_3 and Puppy\_1, 3) Husky\_1 and Puppy\_1. Show and comment on the matching results. Explain why it works or fails in some cases.

You are allowed to use open source library (OpenCV or VLFeat) to extract features.



(a) Husky\_1



(b) Husky\_2



(c) Husky\_3



(d) Puppy\_1

**Figure 3:** Dog images for image matching

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

Apply the K-means clustering to extracted SIFT features to form a codebook. The codebook contains 8 bins, where each bin is characterized by the centroid of the SIFT feature vector. In other words, each image can be represented as histogram of SIFT feature vectors. This representation is called the Bag of Words (BoW). Create codewords for all four images and match Husky\_3's codewords with other images. Show the results and discuss your observations.

## Appendix:

### Problem 1: Texture Analysis and Segmentation

Texture1 to 48.raw	128x128	8-bit	Gray
Comp.raw	600x450	8-bit	Gray

### Problem 2: Salient Point Descriptors and Image Matching

Husky_1.jpg	640x420	24-bit	Color(RGB)
Husky_2.jpg	640x420	24-bit	Color(RGB)
Husky_3.jpg	640x420	24-bit	Color(RGB)
Puppy_1.jpg	640x420	24-bit	Color(RGB)

## Reference Images

Images in this homework are taken from Google images [2], the USC-SIPI image database [3].

## References

- [1] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60(2), 91-110, 2004.
- [2] [Online] <http://images.google.com/>
- [3] [Online] <http://sipi.usc.edu/database/>