EE 569: Homework #4

Issued: 3/12/2024   Due: 3/29/2024, 11:59PM

General Instructions:

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

Problem 1: Texture Analysis (35%)

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 (15%)

48 images of four types of 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.

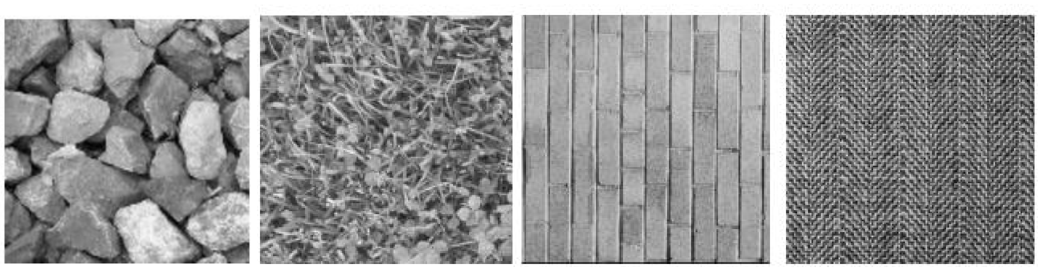

Figure 1: Examples of Stone, Grass, Brick and Blanket

Please follow the steps below to extract features for all texture images provided and do an analysis:

1. **Filter bank response computation:** Use the twenty-five 5x5 Laws Filters in Table 1 to extract the response vectors from each pixel in the image (use appropriate boundary extensions).
2. **Energy feature averaging:** Compute the energy feature of each element of the response vector. Average the energy 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. **Feature dimension reduction via PCA:** Reduce the feature dimension from 25 to 3 using the principal component analysis (PCA). Plot the reduced 3-D feature vectors in the 3-D feature space.

Please conduct texture classification using the nearest neighbor rule based on the Mahalanobis distance. Report test error rate and compare your results with observations (by eyes).

Note: Built-in PCA function can be used.

b) Advanced Texture Classification --- Classifier Exploration (20%)

Based on the 25-D and 3-D feature vectors obtained above, conduct both unsupervised and supervised learning. Please follow the steps below.

1. **Unsupervised:** K-means clustering is a kind of unsupervised learning algorithm which separates the textures into different categories without the help of ground truth labels. It will not directly tell the class for each image but will group similar images.
   a. Apply the K-means algorithm for test images based on the 25-D feature and the reduced 3-D features via PCA, respectively. Set the hyperparameter *K* (number of clusters) equal to the number of possible classes in the dataset (e.g., *K=4*).
   b. Use the test labels to evaluate the purity of each cluster. Specifically, classify the images in each cluster as the majority class of that cluster. Report the error rate for both methods. Discuss the effectiveness of the feature dimension reduction over the K-means clustering.

2. **Supervised:** Use the 25-D features, and the reduced 3-D features of training images to train the Support Vector Machine (SVM) respectively. Then predict the test set labels and report the error rate. Discuss the results from two feature sets. Compared with the unsupervised method, discuss your observation.

Note: Built-in K-means function, SVM can be used.

Problem 2: Texture Segmentation (30%)

a) Basic Texture Segmentation (20%)

Segment the texture mosaic *Mosaic.raw* in Fig. 2 by following the steps below:

1. **Filter bank response computation:** Use the twenty-five 5x5 Laws Filters in Table 1 to extract the response vectors from each pixel in the image (use appropriate boundary extensions).
2. **Energy feature computation:** Use a window approach to compute the energy measure for each center pixel based on the results from step 1. You may try a couple of different window sizes. After this step, you will obtain a 25-D energy feature vector for each pixel.
3. **Energy feature normalization:** All filter responses have a zero-mean except for the $L5^TL5$. Actually, the feature extracted by filter $L5^T L5$ is not useful for texture classification and segmentation. Use its energy to normalize all other features at each pixel.
4. **Segmentation:** Discard the feature associated with $L5^T L5$. Use the K-means algorithm to segment the composite texture image given in Fig. 2 based on the 24-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, 63, 127, 191, 255) to denote five segmented regions in the output for five textures.

b) Advanced Texture Segmentation (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. Use 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 only on the texture properties in these two regions.

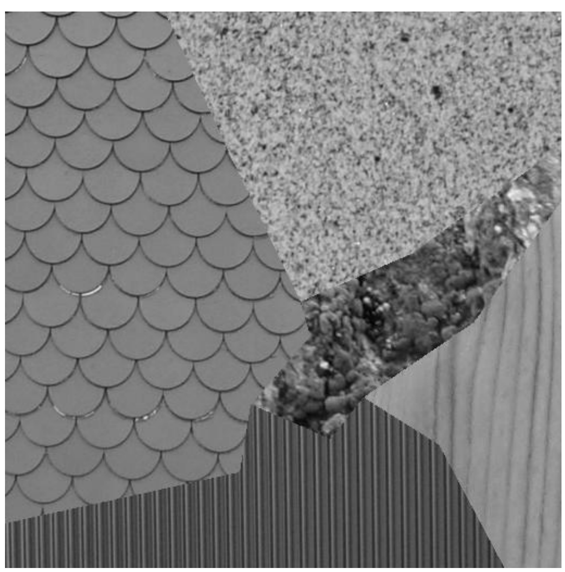

Figure 2. Composite Texture Images[1].

Problem 3: SIFT and Image Matching (35%)

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

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

SIFT is an effective tool to extract salient points in an image. Read the paper in [2] and answer the following questions.

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

(b) Image Matching (Basic: 15%)

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

1. Find key points of the Cat_1 and Cat_3 images in Fig. 3. Pick the key point with the largest scale in Cat_1 and find its closest neighboring key point in Cat_3. You can search the nearest neighbor in the searching database for the query image which is represented as a SIFT-extracted feature vector. Discuss your results, especially the orientation of each key point. Show the corresponding SIFT pairs between Cat_1 and Cat_3.
2. Show the corresponding SIFT pairs between the Cat_1 and Cat_3 in Fig. 3.

Perform the same job with the following three image pairs: 1) Cat_3 and Cat_2, 2) Dog_1 and Cat_3, 3) Cat_1 and Dog_1. The matching may not work well between different objects and against the same object but with a large viewing angle difference. Show and comment on the matching results. Explain why it works or fails in some cases.

You can use an open-source library (OpenCV or VLFeat) to extract features.

(a) Cat_1

(b) Cat_2

(c) Cat_3

(d) Dog_1

Figure 3: Images for image matching. [3]

(c) Bag of Words (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 Cat_3's codewords with other images. Show the results and discuss your observations.

Appendix:

Problem 1: Texture Analysis

48 texture images .raw	128x128	8-bit	grayscale
(include train and test)			

Problem 2: Texture Segmentation

Composite.png	512x512	grayscale

Problem 3: Image Feature Extractors

Cat_1.png	270x180	Color (RGB)
Cat_2.png	270x180	Color (RGB)
Cat_3.png	270x180	Color (RGB)
Dog_1.png	270x180	Color (RGB)

Reference Images

Images in this homework are taken from SIPI Image Database [3], and Google images [4].

References

[1] https://mosaic.utia.cas.cz/
[2]David G. Lowe, "Distinctive image features from scale-invariant key points," International Journal of Computer Vision, 60(2), 91- 110, 2004
[3]https://sipi.usc.edu/database/
[4][Online]http://images.google.com/