EE 569: Homework #4

Issued: 11/01/2015   Due: 11:59PM, 11/29/2015

General Instructions:

1. You are allowed to use any on-line source codes in solving Problem 2 and Problem 3 in Homework #4. However, you need to give a clear reference to the course of these codes (e.g. matlab, OpenCV, etc.)
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: Optical Character Recognition (OCR) (38%)

Develop an OCR system to read alphabetic and numeric symbols. Then, you will use the OCR system to read texts on speed limit signs. The image “Training.raw” in Figure 1 contains a set of alphabetic and numeric symbols to be encountered. Use them as the reference data and build an association between detected symbols and these reference data based on shape similarity.

Figure 1: Training.raw

(a) Decision Tree for Reference Data (14%)

Develop a decision tree classifier (from the root of the leaf) to organize symbols in Fig. 1 so that there is one reference symbol located at one leaf node using the following steps.

1. Segment each symbol into a rectangular region of interest (ROI).
2. Binarize each ROI into a black/while image.
3. Extract features from each binary image obtained in Step 2. You can use properties taught in class or described in the textbook, such as [1] Area, [2] Perimeter, [3] Euler Number, [4] Circularity, [5] Spatial Moment, [6] Symmetry, and [7] Aspect Ratio. Also, you may develop any new desired features to make the classifier more robust.
4. Build a decision tree classifier for each segmented symbol using extracted features.

Describe selected features in your OCR program, and show the decision tree result in the report.

(b) OCR Testing: Simple Cases (14%)

Apply the designed OCR to two simple test images in Fig. 2. Since each image contains multiple symbols of different fonts and sizes, you may perform a primitive segmentation step to separate each symbol in test images. (Note: Image segmentation should be done automatically using your own code.) Recognize the text shown in the two test images and show the decision tree results in the report. Please list any assumptions you make for this problem. Does your OCR system make any mistake? If so, please discuss the reason.

(a) Test_ideal1.raw (b) Test_ideal2.raw

Figure 2: Simple test images

(c) OCR Testing: Advanced Cases (10%)

The speed sign may not be well captured to offer a good input to the designed OCR system. Figure 3 shows two such examples. Please develop some pre-processing techniques to them to provide better-segmented symbols that are suitable for your OCR system. You need to recognize all symbols in the sign.

(a) Test_night.raw (b) Test_shade.raw

Figure 3: Challenging test images

Preprocessing and segmentation techniques are needed to enhance image quality, segment out each individual symbol, and convert dotted lines to solid ones. Apply all your learned image processing knowledge, including but not being limited to:

- *Histogram equalization* (homework #1) to enhance the image contrast,
- *Image denoising* (homework #1) to remove the background noise,
- *Edge detection* (homework #2) to obtain the contour of different symbols,
- *Morphological processing* (homework #3) to get rid of small blobs and lines,
- *Texture analysis & segmentation* (homework #2) to segment the region-of-interest,
- *Geometric modification* (homework #3) to adjust the orientation of the traffic sign,
- *Other techniques* that are beyond the lecture and homework problems.

Then, you will use the OCR system developed in Part (a) to recognize segmented symbols. Describe your preprocessing steps clearly in the report and discuss the robustness of the designed system.

Note: There is no expectation on perfect results, and good grades will be given for good efforts.

Problem 2: Contour Modeling (32 %)

Contour modeling is widely used in medical image analysis. Segmentation of the region of interest (ROI) from different types of medical images (MRI, CT, X-Ray, Ultrasound) using contours is its killer application. Understand the basic ideas of two state-of-the-art contour-based image segmentation algorithms (i.e. the snake algorithm [3] and the level-set algorithm [4]) and use them to solve the following four problems. Compare the performance of the segmentation results of these two methods and comment on their strengths and weaknesses.

(a) Extract the contour of the spine (8%)

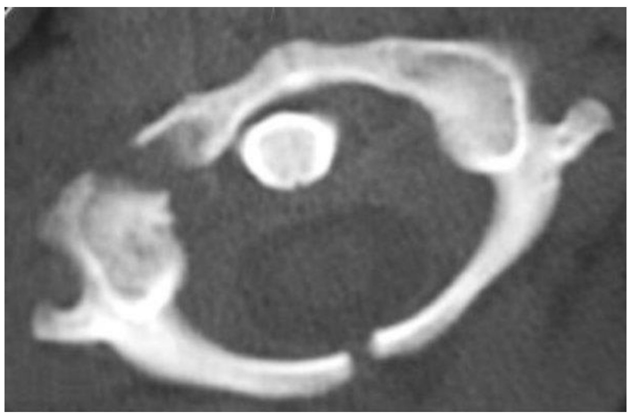

Figure 3: Spine CT image.

(b) Extract the outline of coronary from the following angiogram image (8%)

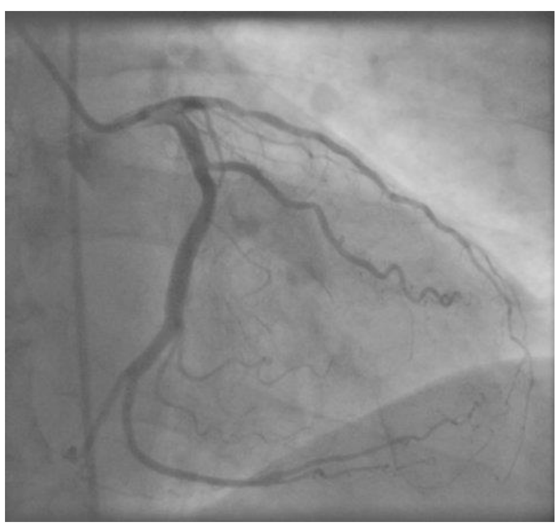

Figure 4: Coronary Angiogram.

(c) Separate red and white blood cells. (Red blood cells are brighter and white blood cells are darker.)

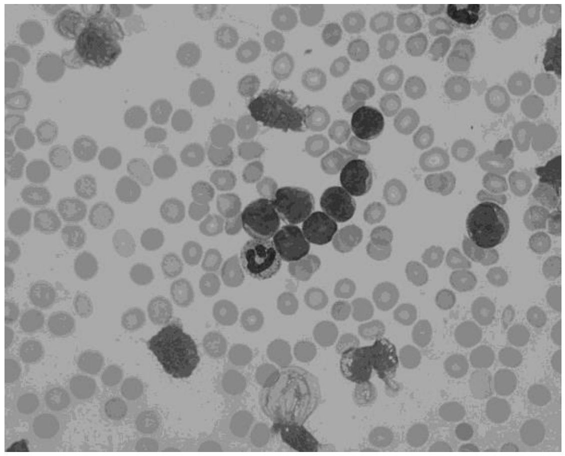

Figure 5: Blood cell image.

(d) Isolate the tumor in the brain MIR image.

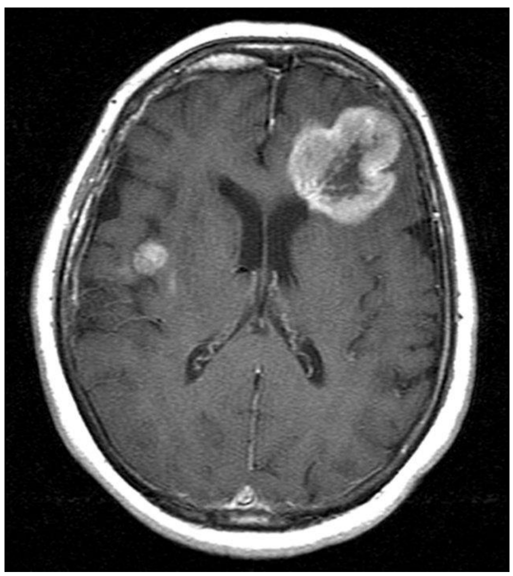

Figure 6: A patient's brain MIR image.

Problem 3: Salient Point Descriptors and Image Matching (30 %)

(a) Extraction and Description of Salient Points (10%)

SIFT [1] and SURF [2] are effective tools to extract salient points and describe their neighborhood in an image. Extract and show both SIFT and SURF features from the two vehicle images in Figure 7. Compare their results in terms of performance and efficiency. Comment on their strength and weakness.

(a) Bus (b) Sedan

Figure 7 Vehicle images

(b) Image Matching (10%)

You can apply SIFT to object matching. Extract and show SIFT features from the two school bus images in Figure 8. Then, show the corresponding SIFT pairs between the two images.

(a) School_bus1 (b) School_bus2

Figure 8 School bus images

The matching may not work well between different objects and against the same object but with a large viewing angel difference. Perform the same image pairing between 1) *School_bus1* and *Bus*, 2) *School_bus1* and *Sedan*. Show and comment on the matching results. Explain why it works or fails in some cases.

(c) Bag of Words (10%)

Apply k-means clustering to extracted SIFT features from three images (Bus, Sedan, and School_bus1) to form a codebook. The codebook contains K=8 bins, where each bin is characterized by the centroid of the SIFT feature vector. In other words, each image can be represented as a histogram of SIFT feature vectors. This representation is called the Bag of Words (BoW). Create codewords for all four images (Bus, Sedan, School_bus1, and School_bus2), and match School_bus2's codeword with other images. Show the results and discuss your observation.

Appendix:

Problem 1: Optical Character Recognition (OCR)

Training.raw	512x256	24-bit	color(RGB)
Test_ideal1.raw	620x759	24-bit	color(RGB)
Test_ideal2.raw	250x400	24-bit	color(RGB)
Test_night.raw	267x360	24-bit	color(RGB)
Test_shade.raw	215x361	24-bit	color(RGB)

Problem 2: Contour Modeling

brian.raw	637x713	8-bit	grayscale
coronary.raw	481x446	8-bit	grayscale
blood_cells.raw	836x672	8-bit	grayscale
spine.raw	600x390	8-bit	grayscale

Problem 3: SIFT and Image Matching

Bus.raw	400x300	24-bit	color(RGB)
Sedan.raw	400x233	24-bit	color(RGB)
School_bus1.raw	374x200	24-bit	color(RGB)
School_bus2.raw	356x200	24-bit	color(RGB)

Sample Code Provided:

kmeans.c	C code provided for k-means algorithm

References

[1] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, 60(2), 91-110, 2004.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346--359, 2008

[3] Michael Kass, Andrew Witkin, Demetri Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision* 1(4), 321-331, 1988.

[4] Tony F. Chan and Luminita A. Vese, “Active contours without edges,” *IEEE Trans. on Image Processing*, Vol. 10, No. 2, February 2001.

Reference Images

All images in this homework are from Google images or the USC-SIPI image database.