# CS 484, Fall 2019
## Homework Assignment 3: Image Segmentation

## Due: December 16, 2019

Image segmentation is a fundamental problem in computer vision. The goal of this assignment is to obtain an accurate segmentation of images using color and texture information. A common problem in segmentation algorithms is that it is usually hard to find a fixed set of parameters that works well for all images in a data set. The same set of parameters may provide an oversegmentation for some images while they may cause undersegmentation for the others.

In this assignment, we will try to obtain a good segmentation by, first, oversegmenting the images, and then, combining (merging) the regions that have common color and texture characteristics. The data set for this assignment consists of 20 images, shown in Figure 1, that are taken from the Common Objects in Context (COCO) data set (`http://cocodataset.org/`). In addition to these 20 images, you must also use the image that you selected for the take-home quiz in the beginning of the semester.

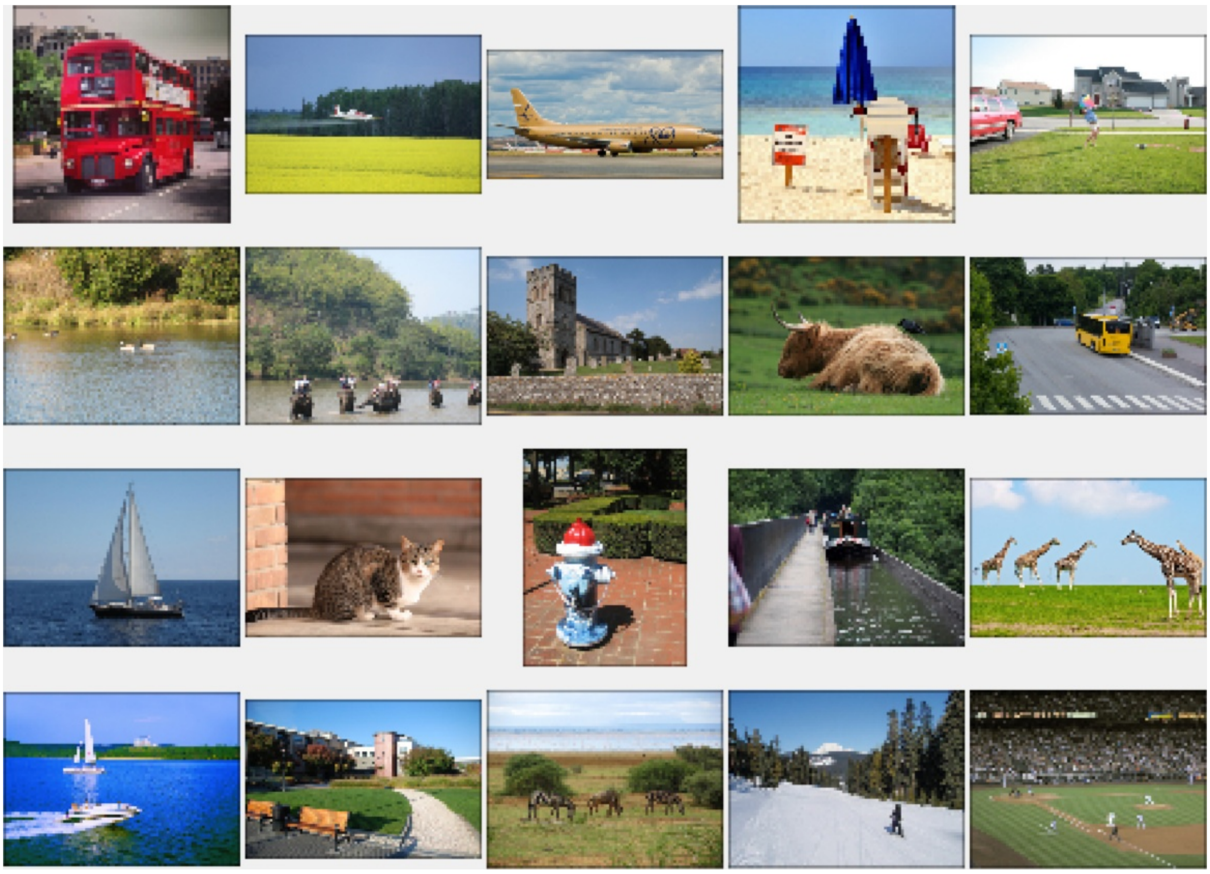
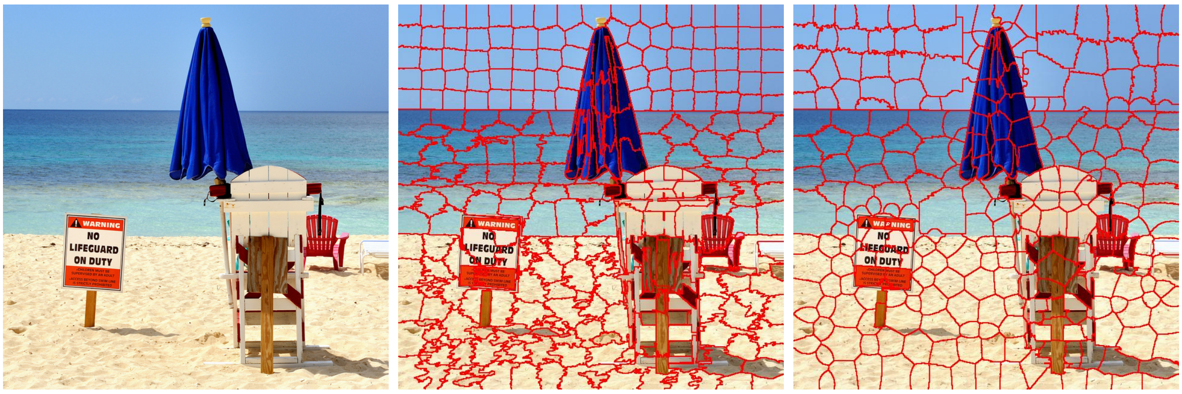


Figure 1: 20 images in the data set of this assignment.




(a) Input image (b) Output of SLIC (c) Output of SLIC0

Figure 2: Example superpixels for an image in the data set.

**Part 1:**

The first step is to obtain an oversegmentation for each image. First, read the following paper (available on the course web page):

> R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, pp. 2274–2282, May 2012.

The algorithm described in the paper can be used to obtain an oversegmentation of an image into superpixels that are locally homogeneous groups of pixels that preserve the object boundaries. Oversegmentation using superpixels can be used as a preprocessing stage that simplifies the computation at later stages.

Second, obtain superpixels for each image in our data set using a SLIC implementation. The authors' codes are available at `https://ivrl.epfl.ch/research-2/research-current/research-superpixels/`. There are two versions (SLIC and SLIC0). You can use either version. Read the instructions for the version you choose carefully. The output that will be necessary in the rest of the homework assignment is a matrix that contains an integer label for each pixel where each label indicates the corresponding pixel's superpixel id. You may need to experiment with the parameters in the code to obtain a meaningful oversegmentation. Example superpixels for an image in our data set are shown in Figure 2.

After trying the code on several images, fix the parameters and use the same settings for segmenting all images. The output of this part for each image is an integer label image where each pixel stores the id of the region it belongs to.

**Part 2:**

The second step is to compute Gabor texture features for all images. You can use the Gabor filter implementation in Matlab's image processing toolbox (`https://www.mathworks.com/help/images/ref/imgaborfilt.html` or the Matlab implementation by Dr. Peter Kovesi (`http://www.peterkovesi.com/matlabfns/PhaseCongruency/gaborconvolve.m`). You can adjust the parameters so that you obtain a filter bank of 4 scales and 4 orientations. You can use the suggested default values for most of the parameters. Note that Gabor filter responses should be computed on the grayscale version of the input image, and the magnitude of the output response should be used in the rest of the assignment. At this step, for each pixel, you will have one response for each filter, e.g., you will have a total of 16 responses per pixel if you use 4 scales and 4 orientations.

**Part 3:**

The third step is to implement your own region merging algorithm as follows:

1. Input to the algorithm is the set of superpixels as individual regions.
2. Compute, for each region (using only the pixels in that region),
   (a) a color feature vector that contains the average of the color features (3 values: average R, average G, and average B, or average L, average a, and average b after converting the RGB image into the Lab color space),
   (b) a texture feature vector that contains the average of the Gabor texture features (e.g., 16 values for 4 scales and 4 orientations),
   (c) a color-texture feature vector that contains the color feature vector in (a) and the texture feature vector in (b) appended together.
3. For each pair of neighboring regions, if the Euclidean distance between their feature vectors is less than a threshold, merge the two regions. Do this for all pairs of neighboring regions. Note that you will have to normalize each feature component to the [0,1] range before computing the Euclidean distance.
4. Repeat steps 2 and 3 until no pair of regions can be merged.

Run the region merging algorithm using only color features, using only texture features, and using color-texture features together. Using a different threshold will give a different level of merging. After trying the algorithm on several images, fix the threshold and use the same threshold for region merging in all images. The final output for each image must be an integer label image where each pixel stores the id of the region it belongs to.

**Submit:**

1. Description of the parameters used for superpixel segmentation.
2. Segmentation results for all images using the superpixel segmentation algorithm (SLIC version) that you chose. Note that you must produce oversegmentations using the same parameters for all images.
3. Description of the parameters used for Gabor texture feature extraction.
4. Gabor texture feature examples (like we had in the lecture slides) for several scales and orientations for at least five different images.
5. Your own source code for the region merging implementation.
6. Description of the thresholds selected for region merging.
7. Segmentation results for all images after region merging using different features (color, texture, color-texture).
8. Detailed discussion of the results with respect to different features, different merging thresholds and other parameters.
9. Citation for any external code used.

**Notes:**

This assignment is due by midnight on Monday, December 16, 2019. An important part of this assignment is the discussion of the results where you must comment on the effectiveness of the segmentation for different images with respect to different features and different segmentation parameters.

You should upload your solutions as a **single archive file** that contains your **code** and **report** (a pdf file that contains the resulting images, descriptions of how you obtained them, and discussion of the results) using the online submission form on the course web page before the deadline. Please see the course syllabus for a discussion of the late homework policy as well as academic integrity. If you have any questions about what is allowed and what is not allowed in a solution, please check the course syllabus on the course web page.