EECS 442 Computer Vision: HW5

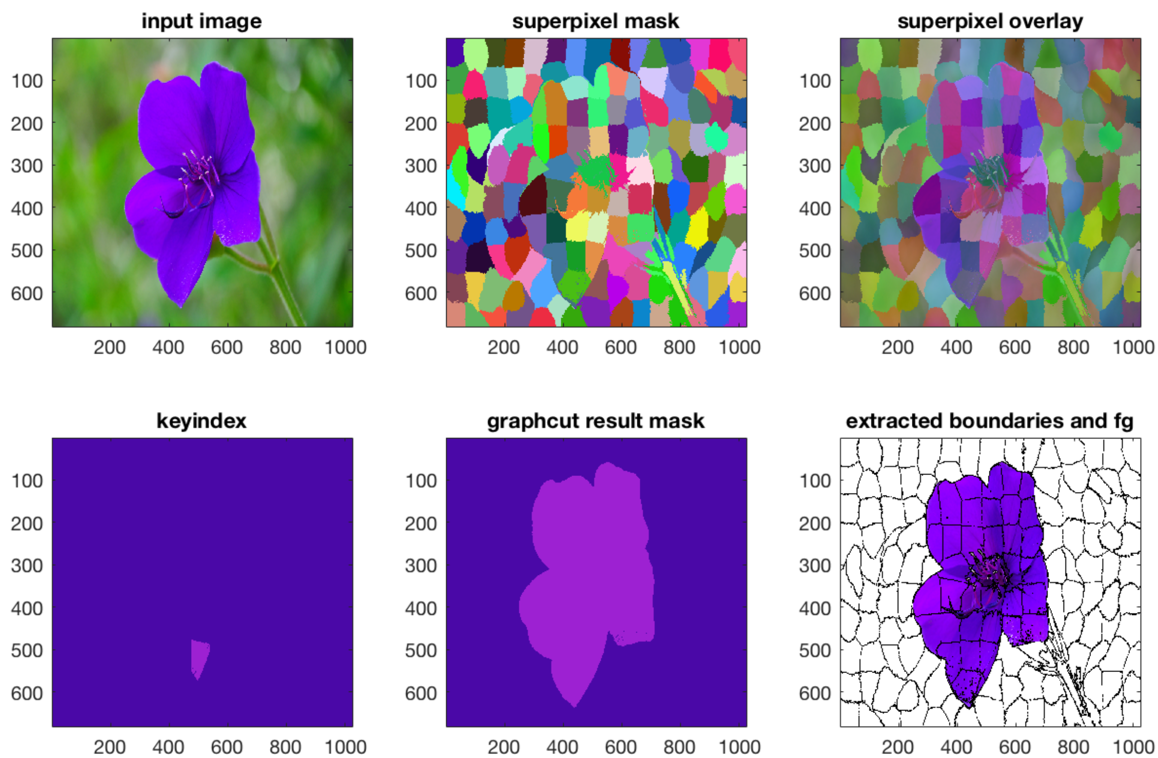
Term: Fall 2017
Instructor: Jason J. Corso, EECS, University of Michigan
Due Date: 11/17 23:59 Eastern Time

Constraints: This assignment may be discussed with other students in the course but must be written independently. Over-the-shoulder Matlab coding is strictly prohibited. **Web/Google-searching for background material is permitted. However, everything you need to solve these equations is presented in the course notes and background materials, which have been provided already.** It is my suggestion you spend the time trying to working through the problems on your own before looking to discuss or search. You may not search explicitly for a problem on the web.

Goals: Deepen understanding of images as graphs by implementing a 2-class graph-cut method.

Data: You need to download the supplemental data to complete this assignment. All paths in the assignment assume the data is off the local directory. The download has data and code; it is named `hw5.zip`.

We discussed the figure-ground graph cuts case study at length in class. You will implement this method for figure-ground extraction on color images using a superpixel graph and color histograms as the feature. A basic MRF is implemented in the graph. The figure below shows the expected results and facets of the assignment.



The provided code includes the core framework, an implementation of the SLIC superpixel method (a pdf of the article describing the method is included), and an implementation of the Ford-Fulkerson algorithm for computing the max-flow. You will implement the histogram feature representation and all aspects of taking the superpixel result and implementing the graph on top of it as well as reading out the results of the graph cut as a two-class segmentation.

The file `example.m` provides a full run through the whole method for you. Your results from running it will be submitted. You will also need to run additional methods.

Note that p-code is provided with a `j_` prefix on the files you need to implement. This is for your help and benefit, and it is used to compute accuracy for submission. You will not report results from the p-code as your own, nor will you make any attempt to deconstruct the p-code.

Problem 1 (10): Color Histogram Features

Implement the missing body of code in `histvec.m`, which creates a color histogram feature vector. Follow the comments in the code for the details. We call this with `C = reduce(im,C,S);`.

Run `q1.m`, which will generate an output image `q1_result.png`. Include this output in the writeup. Also include the code as plain text.

Also, inspect the code in `q1.m` and answer the following question in one sentence: Describe the object in the image that is covered by the superpixel used to compute the histograms in `q1.m`.

Problem 2 (20): Superpixel Adjacency

1. (10) Implement the missing body of code in `segNeighbors.m`, which computes the adjacency matrix for the superpixel graph. Follow the comments in the code for the details. We call this in the graph cuts code.

   Run `q2.m`, which will generate an output image `q2_result.png`. Include this output in the writeup. Also include the code as plain text.

2. (5) Implement a small function to compute the average node degree. Include the code as plain text in the writeup.

3. (5) Why is the adjacency graph not a perfect banded diagonal matrix?

Problem 3 (25): Graph-Cuts

1. (15) Implement the missing two bodies of code in `graphcut.m`, which (1) creates the graph by defining the capacity matrix and (2) extract the results after running the max-flow/min-cut method. See the comments and refer to class notes for details.

   Run `q3.m`, which will generate an output image `q3_result.png`. Include this output and the full implementation as plain text in your writeup.

2. (5) Using the debugger, save an image of your capacity matrix before running graph cuts in `q3.m` and submit it. In a few sentences, relate the adjacency matrix to the capacity image. Be sure to cover **all** nodes of the graph in your description.

3. (5) Please explain why the capacity between adjacent nodes in the graph that have resulted from superpixels is downweighted with respect to the capacity between nodes in the graph and the special source and sink nodes.

Problem 4 (20): Graph-cuts Study

Use the `example.m` code here and change accordingly. You can change other parts of the code too, if needed, but be clear to note it.

1. (5) Run code as provided to run through a full example and show the resulting segmentation on the flower. Include the result.

2. (5) Run the code using the `flag1.jpg` image. Manually select a stripe on the flag. Are you able to get a full segmentation of the stripe and no other regions? If so, explain what you did to make this possible. If not, explain why this is hard. Include an rendering of the figure to substantiate your explanation either way.

3. (5) Run the code using the `porch1.png` image. Are you able to segment the boots perfectly? If so, explain what you did to make this possible. If not, explain why this is hard. Include an rendering of the figure to substantiate your explanation either way.

4. (5) On the porch1.png image again, are you able to segment either of the baskets perfectly? My guess is no. Can you describe (but do not implement) a way to change this system to make this more possible?

Note: For this assignment, you also need to submit all your original .m files (packed as a single file) on Canvas. Specifically, you need to submit your `histvec.m`, `segNeightbors.m` and `graphcut.m`

Submission Process (updated):

Submit a single pdf with your answers to these problems, include the code verbatim as text in the pdf. Include all plots and discussion in the pdf. Submit the pdf to Gradescope. The entry code of this course is **M5VRZV**.

Pack the original program files into one file and upload your code to Canvas. The problem description will clarify whether you should turn in the code for that problem.

Grading and Evaluation: The credit for each problem in this set is given in parentheses at the stated question (sub-question fraction of points is also given at the sub-questions). Partial credit will be given for both paper and Matlab questions. For Matlab questions, if the code does not run, then limited or no credit will be given.