Computer Vision (600.461/600.661)
Homework 6: Segmentation and Recognition

Instructor: René Vidal

Due 12/02/2014, 11.59PM Eastern

1. **(25 points) Image Segmentation**

   (a) **(10 points)** Implement the Spectral Clustering algorithm as follows:

   Function `[group, evals, evects] = spectral(W,K,method)`

   Parameters

`W`	$N \times N$ similarity matrix
`K`	the number of groups
`method`	the method to be used from `'unnormalized'`, `'normalized'` and `'symmetric'`

   Returned values

`group`	$1 \times N$ vector with group membership of each point
`evals`	$1 \times K$ vector of the smallest K eigenvalues of the graph Laplacian
`evects`	$N \times K$ matrix of the corresponding eigenvectors of the graph Laplacian.

   Description

   Computes the clustering of points using spectral clustering algorithm.
   `'unnormalized'` uses the graph Laplacian, $L = D - W$
   `'normalized'` uses the graph Laplacian, $L = I - D^{-1}W$
   `'symmetric'` uses the graph Laplacian, $L = I - D^{-1/2}WD^{-1/2}$
   where D is the *diagonal degree matrix* with $d_i = \sum_{j=1}^{N} W_{ij}$.

   You may want to code it in such a way it exploits the sparsity of W whenever the matrix W is sparse.

   (b) **(7.5 points) Intensity-based.** Use the `kmeans` and `spectral` (symmetric) methods to segment the images on the course webpage using intensity only. For `kmeans`, use the building matlab function with multiple restarts. For the `spectral` use your own code with a k-connected neighborhood so that W is sparse. Feel free to choose appropriate values for σ_I and k in defining similarities for intensity and number of neighbors, respectively. Comment on the performance of the different methods on the different images.

   (c) **(7.5 points) Color-based.** Use the `kmeans` and `spectral` (symmetric) methods to segment the images on the course webpage using color only. Use both the (r, g) and the (H, S) color representations. Feel free to choose appropriate values for σ_c and k in defining similarities for color and number of neighbors, respectively. Comment on the performance of the different methods on the different images.

2. **(25 points) Object Recognition**

   (a) **(10 points) Recognition of Object Instances.** Go to images.google.com. Type "tour eiffel". Download the top 11 images. Using image 11 as a template, your goal is to match (recognize) the template to each one of the first 10 pictures. For this purpose, please read and follow all the steps in Vedaldi's object instance recognition tutorial. You should submit a script called `hw6q2a.m` that loads the images, calls Vedaldi's code, and plots the results. If your script calls a new function that you have created, then you should also submit this new function.

   (b) **(15 points) Recognition of Object Categories.** Given images of airplanes and motorbikes, your goal is to train classifiers for determining whether a novel image contains an airplane (motorbike) or not. For this purpose, please read and follow all the steps in Vedaldi's object category recognition tutorial. You should submit a script `hw6q2b.m` that loads the images, calls Vedaldi's code, and gives the results. If your script calls a new function that you have created, then you should also submit this new function. For each dataset, please use the first 200 hundred images for training, and the remaining 600 for testing.

Submission instructions. Send email to vision14jhu@gmail.com with subject 600.461/600.661:HW6 and attachment firstname-lastname-hw6-vision14.zip or firstname-lastname-hw6-vision14.tar.gz. The attachment should have the following content:

1. A file called hw5.pdf containing your answers to each one of the analytical questions. If at all possible, you should generate this file using the latex template hw1-vision14.tex. If not possible, you may use another editor, or scan your handwritten solutions. But note that you must submit a single PDF file with all your answers.

2. For coding questions, submit a file called README, which contains instructions on how to run your code. Use separate directories for each coding problem. Each directory should contain all the functions and scripts you are asked to write in separate files. For example, for HW2 the structure of what you should submit could look like

   (a) README

   (b) hw2.pdf

   (c) hw2q3: hw2q3c.m, hw2q3e.m

   (d) hw2q4: hw2q4b.m, hw2q4c.m

   The TA will run your scripts to generate the results. Thus, your script should include all needed plotting commands so that figures pop up automatically. Please make sure that the figure numbers match those you describe in hw2.pdf. You do not need to submit input or output images. The output images should be automatically generated by your scripts so that the TA can see the results by just running the scripts. In writing your code, you should assume that the TA will place the input images in the directory that is relevant to the question solved by your script. Also, make sure to comment your code properly.