CMPE 565- Assignment #1

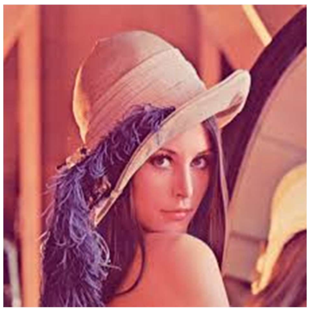
Due: 27.10.2019, 23:59

Note: Homework is to be done individually. You need to submit you reports and codes to Moodle. Please do not insert your codes to your reports. Codes should come separately. In your report just put the input images and output results and explicitly stated parameters being used (like sigma parameter for Gaussian).

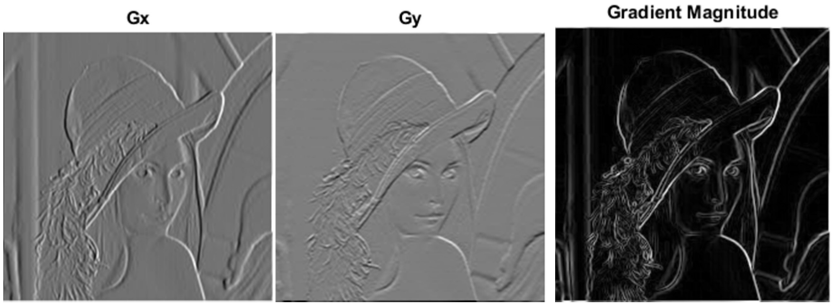
1) **[20 pts] Edge Detection**

   a) **[15 pts]** Implement *SobelEdge (image)* function. This function should receive an image, convert it to grayscale and perform edge detection on the image using separable Sobel operators. This means that you will be implementing the convolution process yourselves. Using Matlab functions are forbidden for this assignment. At the end this function should return Gx and Gy (partial derivative of an image in horizontal and vertical direction). **Pay attention to boundary issues here!! You can assume zero values at the boundaries.**

   b) **[5 pts]** Using the Gx and Gy images obtained from 1.a compute the gradient magnitude of an image. Put the resultant plots to your reports.
   For example:
   Given the following image as an input



   The following images will be your outputs



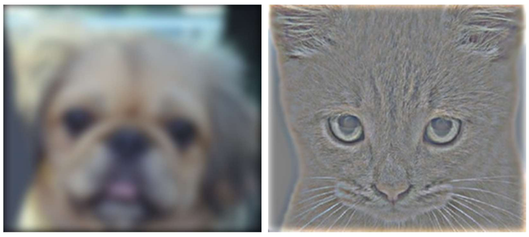
2) **[30 pts] Hybrid Images**

   A hybrid image is the sum of a low-pass filtered version of the one image and a high-pass filtered version of a second image. There is a free parameter, which can be tuned for each image pair, which controls *how much* high frequency to remove from the first image and how much low frequency to leave in the second image. This is called the "cutoff-frequency". In the paper it is suggested to use two cutoff frequencies (one tuned for each image) and you are free to try that, as well. You are going to try different sigma values and observe the hybrid image that you obtained. Report on the result. **You are free to use Matlab functions for this assignment.**

We provide you with 5 pairs of aligned images which can be merged reasonably well into hybrid images. The alignment is important because it affects the perceptual grouping (read the paper for details). You can use those images or provide other images of you own choice.

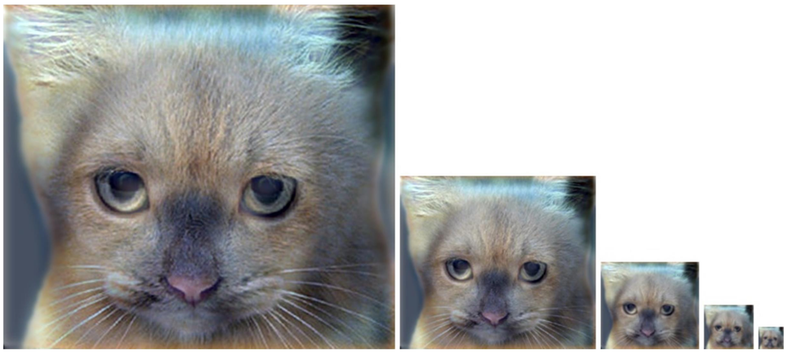
For the example shown at the top of the page, the two original images look like this:

Low-pass and high-pass version of these images are as follows:



The high frequency image is visualized by adding 0.5.

Adding the high and low frequencies together gives you the image at the top of this page. If you're having trouble seeing the multiple interpretations of the image, a useful way to visualize the effect is by progressively downsampling the hybrid image as is done below:



3) **[50 pts] Image Blending**

For this assignment, you are expected to implement a pyramid blending pipeline that takes three images as an input, and outputs blending of these images. The first image is the source image; the second image is a binary mask which contains some region from the source image that you want to blend to the last image, which is a target image. **(This assignment will require you to create a blend using your own images!).**

You codes should contain these three functions:

i. **[15 pts] gaussPyramid (image) -** This function takes an image and builds a pyramid out of it. The first layer of this pyramid is the original image, and each subsequent layer of the pyramid is the reduced form of the previous layer. Put simply, you are iteratively calling the reduce function on the output of the previous call, with the first call simply being the input image. **You can use *imresize(image,0.5)* as reduce function. You can use cells to store image pyramids.**

ii. **[15 pts] laplPyramid(image) -** This function takes a Gaussian pyramid constructed by the previous function, and turns it into a Laplacian pyramid as discussed in the class. **Like with Gaussian pyramids, Laplacian pyramids also can be stored in cells. You can use *imresize(image,2)* as expand function.**

iii. **[10 pts] collapse (image) -** This function is given a laplacian pyramid and is expected to 'flatten' it to an image. We need to take the bottom (smallest) layer, expand it, and then add it to the next layer. We continue this process until we reach the top of the pyramid.

iv. **[5 pts]** for your images with masks.

In the report, put the images that you chose, masks, blending results and clearly explain the steps you have taken as an algorithm.