Introduction

For this assignment, we will use a bank of texture filters along with a clustering algorithm, to segment an image into foreground and background regions. The foreground region will then be extracted from the original image and placed in a separate background image. As an application of this technique, you will transfer an animal from one image to another as shown in Figure 1. To complete the assignment you will need to download the **homework2.zip** file from myCourses as it contains the images and some code that is required.

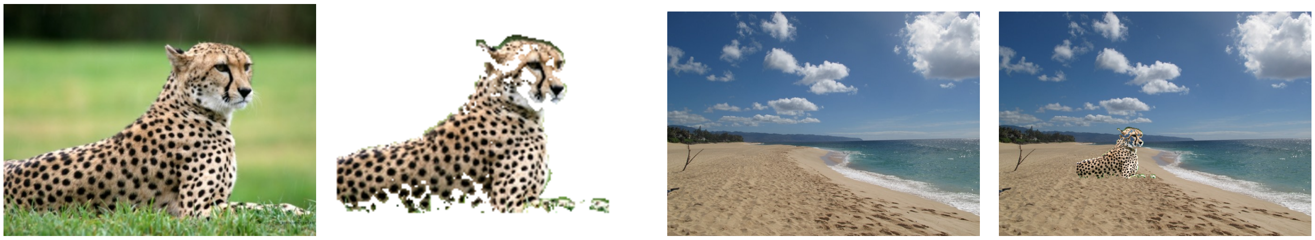

Figure 1: Transferring an object from one image to another. Far left: original image; center left: segmented textured animal; center right: original background; far right: composite image - where we "grab" the animal from the first image and place it in the background image.

.

Requirements

You should perform this assignment in Python and feel free to use routines such as convolution functions from OpenCV or other simlar libraries. The assignment is due on **Sunday October 21st by 11:59pm**. You are strongly encouraged to start early. You are welcome to ask questions and have discussions about the homework on myCourses but please do not post your solutions or any closely related material. If there are parts of the assignment that are not clear to you, or if you come across an error or bug please don't hesitate to contact the TAs or the Instructor. Chances are that other students are also encountering similar issues.

You are allowed to collaborate with other students as far as discussing ideas and possible solutions. However you are required to code the solution yourself. Copying others' code and changing all the variable names is not permitted. You are not allowed to use solutions from similar assignments in courses from this or other institutions, from other students who are taking this or a similar course, or similar solutions found on the web. If you access such solutions YOU MUST refer to them in your submission write-up. Your solutions should be submitted via Dropbox on myCourses.

The data and starting code are provided in the zipped file **homework2.zip** which can be downloaded from myCourses. This contains image files for testing along with some prewritten code to facilitate the homework.
Your submitted zipped file for this assignment should be named **LastnameFirstname_hw2.zip**. Failure to follow this naming convention will result in delays in grading your work. Your

zipped file should contain: (i) a PDF file named LastnameFirstname_hw2.pdf with your report, showing output images and explanatory text, where appropriate; (ii) the source code used to generate the solutions (with code comments). You do not need to include any images with your final submission (but you should have images of your results embedded in your report). For grading, we will be testing your code on both the given images and on a different set of images you have not seen.

Problem 1. Foreground-background texture-based segmentation via clustering (Total 100 points)

The goal of this assignment is learn to segment an image that contains multiple textures into a foreground and background region, using a bank of N_{fil} filters to identify the different textures in the image. By convolving the image with the N_{fil} filters in the bank, each pixel is now transformed to an N_{fil}-dimensional vector. These vectors are then clustered using the k-means algorithm. A subset of the k segments that represents the foreground region is then transferred into a different background image. See Figure 1.

You are provided with the Leung-Malik (LM) bank of filters. This LM filter set is a multi scale, multi orientation filter bank with 48 filters. It consists of first and second derivatives of Gaussians at 6 orientations and 3 scales making a total of 36; 8 Laplacian of Gaussian (LOG) filters; and 4 Gaussians. We consider an over-complete bank where the filters occur at the basic scales $\{\sigma = \sqrt{2}, 2, 2\sqrt{2}, 4\}$. The first and second derivative filters occur at the first three scales with an elongation factor of 3 (i.e. $\sigma_x = \sigma$ and $\sigma_y = 3\sigma_x$). The Gaussians occur at the four basic scales while the 8 LoG filters occur at σ and 3σ. The filter bank is shown in Figure 2.

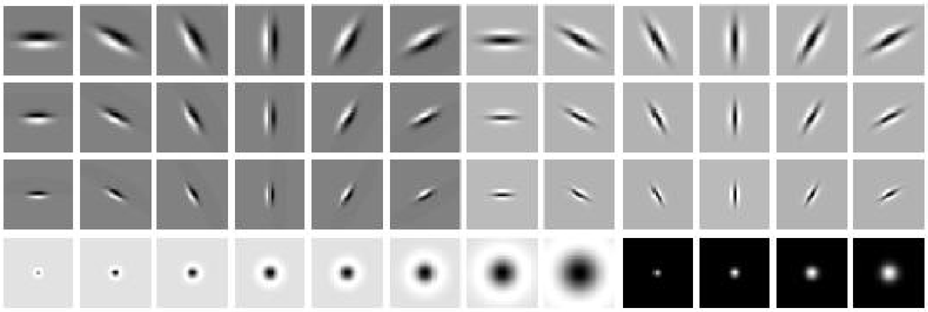

Figure 2: The LM filter bank has a mix of edge, bar and blob detectors at multiple scales and orientations. It has a total of 48 filters - 2 Gaussian derivative filters at 6 orientations and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters.

Your tasks for the assignment include the following:

(a) Download the code and data from the zipped file. The data which is stored in the `\images` directory, consists of four textured animal images along with three background images, all named accordingly. You will be segmenting out each of these animals from the original images and transferring them to any background images of your choice.

(b) (10 points) Before implementing your segmentation you should write your own version of the k-means clustering algorithm although you are strongly advised to use the an implementation from an existing library such as scikit-learn for the rest of your work (The scikit-learn version has been optimized significantly). If you choose not to write your own version you will lose the points allotted for this portion of the problem.

(c) (60 points) You are now ready to segment. First create a function `segmentImg` and that can read in an image file (`.png` or `.jpg` file) and output a segmented version of the image. In this function, you will create a bank of filters and convolve each filter in the bank with your input image. You are provided with a file `makeLMfilters.py` which returns a bank of 48 filters as shown in Figure 2. Then you will use the absolute value of your filter responses to construct a data matrix `X`. Finally, use k-means (either yours or the optimized library version) to cluster the 48-dimensional points in `X`. You should have a total of k clusters. Reshape your clustering result into the dimensionality of the input image. Play with different values of k to see which gives you the best segmentation. The output of this should be a segmentation of the input image.

(d) (15 points) Next, create a function `transferImg` to transfer your segments (obtained via k-means) into a new background image of your choice. The signature for `transferImg` should be `[fgs, idxImg, sImg, tImg]`. The first variable `fgs` is a vector containing the indexes representing the foreground values from your segmentation, for example `fgs = [1, 3, 4]`. The second variable `idxImg` is the segmented image obtained from function `segmentImg` above. The third and fourth variables `sImg` and `tImg` stand for your source and target images respectively, where `sImg` is the initially loaded image `I` and `tImg` is any one of the background images given. The output of this function should be the composite image where the animal has been transferred into a new background. Additional details for `transferImg`:

(i) First, crop the sides of `idxImg` and `sImg` to remove border problems

(ii) Next, resize both `idxImg` and `sImg` by 0.5 before you perform any transfer, otherwise the source image will dominate your new background.

(iii) Lastly, for every pixel in `idxImg` check if its value exists in `fgs`, and if so, copy the true pixel value from `sImg` to `tImg`. Stay well within the bounds of the target image when copying over pixels from the source image.

(e) (15 points) A well-written, neat and concise report which includes a section for each of the following questions:

(i) Briefly describe your implemented solution, focusing especially on the interesting parts of the implementation. What are some artifacts and/or limitations of the implementation, and what are possible reasons for them?

(ii) Please indicate specifically whether or not you implemented your own version of k-means clustering. If your report does not indicate this, we will not specifically check the code and it will be assumed that you did not, and no grade will be given for implementation.

(iii) For the dog image and at least one other animal image given, display the original source image and the segmented animal transferred in to a new background image. If you want to display just your segmentation, use a plain white image as the background to transfer to.

(iv) As you would have noticed in your results, many of the segmentations have holes in them and the animals are not wholly segmented out. Discuss why this is the case and how different choices in our segmentation strategy could improve the results. Feel free to include additional input images or run additional experiments to illustrate your points.

(f) For extra credits (maximum of 10 points extra), you can augment the pixel features by not only using textures, but also including color to improve the quality of the final segmentation. You can also use windowed averaging to see if this improves your overall results. If you use color, you may want to down-weight the textures and up-weight the color channels. You could also try increasing the number of iterations in the k-means segmentation.

You should turn in both your code and report discussing your solution and results to get full credit.