Assignment 2 - Clustering and Segmentation

IMAGES. All images may be found in the TEST IMAGES folder `https://www.dropbox.com/sh/7aw3coygs3r8naw/AACUIg6mQyI1__815I5zOOlYa?dl=0`.

1. **Independent Reading**.
   (i) Normalized Cuts for Segmentation: `https://ieeexplore.ieee.org/document/868688`
   (ii) Active Contours: `https://link.springer.com/article/10.1007/BF00133570`

2. **(5 pts)** In this problem, you will be reconstructing images using principal components. Vectorize each image in the "pca face images" dataset (`https://www.dropbox.com/s/0g15mi9czdbcqfy/pca_face_images.zip?dl=0`). Compute the mean of the images and subtract out the mean. Let us call each zero mean vectorized image to be X. For N images, you have $X_1, X_2, \ldots, X_N$. Compute $\left[XX^T\right] = R$.

   (a) Perform a principal components analysis of $R = U\Sigma U^T$.
   (b) Analyze the eigenvalues in Σ and decide which eigenvalues to retain and which can be set to zero. You may want to plot them and set a threshold.
   (c) Reconstruct an approximation of each X after removing some of the small eigenvalues. (Display only a couple of them.)
   (d) Compute the error between the reconstructed X and original image (you will need to add the mean back).
   (e) Analyze by choosing different numbers of eigenvalues to be zeroed out.
   (f) Cluster the data using the PCA coefficients. Do the images belonging to an individual cluster together?

3. **(4 pts)** On the `peppers_color.tif` image, implement the Expectation Maximization algorithm for mixture of Gaussian model based on color features for segmenting the image. Again, visually analyze the results and comment on the accuracy of the results. Focus on implementing the basic mathematical steps properly.

4. **(2 pts)** Run Deeplabv3 on `lena_color_256.tif` and compare the results with the results of the segmentation algorithm in Question 3 or Question 4 (depending on your choice) for `lena_color_256.tif`. The deeplabv3 starter code can be found in `https://colab.research.google.com/github/tugstugi/dl-colab-notebooks/blob/master/notebooks/TorchvisionDeepLabV3.ipynb`. Make the necessary changes related to uploading of the image file and add the comparative analysis in this colab notebook itself. Give your thoughts on the two segmentation maps. Make sure to rename the notebook to Problem 4.

5. **(3 pts)** While the fundamental concept of the Fourier Transform has long been familiar to the image processing community, a noteworthy application pertaining to style transfer has only recently emerged, as detailed in this work: `https://openaccess.thecvf.com/content_CVPR_2020/papers/Yang_FDA_Fourier_Domain_Adaptation_for_Semantic_Segmentation_CVPR_2020_paper.pdf`. For this assignment, your task is to solely incorporate the core principle of Fourier style transfer. This entails writing code that takes a collection of source and target images and outputs the contents of the source image styled according to the target image. Source and target images can be found within the assignment folder. Additionally, you are encouraged to utilize any built-in library implementations for the basic execution of the Fast Fourier Transform.

Submission Protocol. You must submit codes written in Python for every problem. Unless otherwise specified, the code for each problem must be a Jupyter notebook i.e. `.ipynb` file. You can use Colab if you want. You should add comments to your codes to make them reader friendly. All codes must be uploaded in separate folders named after the problem number. Keep all the images necessary to run a code in the same folder as the code while you are submitting. You must provide explanations in your notebooks related to each problem (if required). Also, please include a report of your findings. The zip file to be uploaded must have your name.