Programming for Data Science
Assignment 4

Dr Hyung Jin Chang

This assignment is worth 25% of the module mark. It is based on the topics that you have been learning so far. In particular, this will focus on computer vision tasks and their related python libraries.

You are required to perform some computer vision tasks on the given images. Please download the image files through the link in the assignment description. Read the tasks below carefully, complete 3 questions with Python programs. For each question, write python programs to solve, and include an explanation of your results and findings as indicated in the question. NumPy, Scipy, OpenCV, Matplotlib, and h5py are likely to be used for implementation.

Instructions are provided first for a reason. Please read the instructions carefully.

Instruction

Submit one Jupyter notebook file (.ipynb) only, including both codes and explanations required in each question (*i.e.* put codes in "code" cells, and descriptions in "markdown" cells). It is not required for you to explain your code in the notebook. The descriptions are for discussing results. You should, however, provide comments in your code, well enough to be understood by directly reading the code. All assignments should be submitted electronically. Handwritten codes or reports are not accepted. All assignments should be in Python 3. In other words, do not use Python 2.7.

Question 1 [5%]

Write a Python script that performs the following:

(a) Reads `input.jpg`, provided in the assignment package, and print the image shape [0.5%].

(b) Resizes the image to 640 x 480 (width x height) [0.6%].

(c) Displays the image using `Matplotlib` or `OpenCV` as inline in the notebook [0.5%].

(d) Inverts the image colour by subtracting the image from a white image [0.6%].

(e) Displays the inverted image [0.5%].

(f) Saves the inverted image to `inverted-output.jpg` [0.6%].

(g) Combines the original image and the inverted image side-by-side as a new one image of size 1280 x 480 (width x height) [0.6%].

(h) Displays the newly combined image [0.5%].

(i) Saves the combined image to `combined-output.h5` in `hdf5` format. `hdf5` is a standard format for storing data easily. Especially with the `h5py` library, the interface is quite straightforward. Save the image in the `data` group [0.6%].

For example, in the Jupyter notebook, you should show that you successfully read and displayed the image, as well as the inverted image and the combined image. You should include in your submission the `inverted-output.jpg` and `combined-output.h5`.

Question 2 [10%]

Now that you can load an image, you can start doing some basic Computer Vision tasks. For this assignment, let's try extracting edges and corners through the difference of Gaussian filtering. Don't worry if you have no idea what this is about, as I don't expect you to. This is just a simple image filtering and processing task. Going into details on the idea and the maths behind how and why the difference of Gaussian filtering detects edges is beyond the score of this module. For this question, implement a python script that performs the following:

(a) Reads the `combined-output.h5` file from **Question 1**, and displays it [0.5%].

(b) Makes image single-channel by taking the mean along the last dimension, and displays the result [1.1%].

(c) Defines three sets of kernel size *k1* and *k2*: $(k1, k2) = \{(2, 4), (3, 6), (4, 8)\}$ (if it works) or $(k1, k2) = \{(1, 3), (3, 5), (5, 7)\}$ (for example, the first set is *k1*=1 and *k2*=3 respectively) [0.3%].

(d) Applies Gaussian filtering on the converted image from Question 2(b) with kernel size *k1*, and displays the result [0.4%].

(e) Applies another Gaussian filtering on the image with kernel size *k2*, and displays the result [0.4%].

(f) Subtracts *k1* result from *k2* result [0.4%].

(g) Normalises the result to be between 0 and 1 [0.5%].

(h) Displays result inline [0.3%].

(i) Saves to `filtered-k1-k2.h5`, again the `data` group (for example, `filtered-2-4.h5` for *k1*=2 and *k2*=4) [0.4%].

(j) Repeats (c)-(i) on the kernel size sets, and displays the three results altogether using subplot [0.3%].

You should include in your submission the `filtered-k1-k2.h5` files.

Question 3 [10%]

The final part of the assignment is to threshold the filtered image and leave only the "interesting" pixel values. Write a Python script that:

(a) Reads the `filtered-k1-k2.h5` file from **Question 2** [0.3%].

(b) Thresholds the filtered image to keeps approximately 20% of the pixels that have the highest values, and the rest is set to zero [1.8%].

(c) Displays result [0.3%].

(d) Saves to `thresholded-k1-k2.h5`, again the `data` group [0.3%].

(e) Repeats (a)-(d) on the kernel size sets, and displays the three results altogether using subplot [0.3%].

(f) Discusses the effects of the different kernel sizes on the results [1.6%].

You should include in your submission the `thresholded-k1-k2.h5` files.