MATLAB/PYTHON EXERCISES ON IMAGE ENHANCEMENT AND RESTORATION

TUTOR: MIRELA POPA

TEACHING ASSISTANT: ESAM GHALEB

DATE: 15/04/2021

In this exercise, you will familiarize yourself with image enhancement, as well as restoration using median, Gaussian and band-reject filters.

For this exercise, download images **lab2a.png, lab2b.png** and **Unequalized_H.jpg** from Canvas/Modules/week2.

Part 1: Point-based, histogram-based and spatial enhancement (Use lab2a.png for this part)

a) Create two different functions for Contrast Stretching and Clipping. Various parameters should be inserted by the user, and error prevention messages should be displayed. Showcase your results and comment on the use of various parameters (a,b, y_a, y_b, c)

b) Convert the image into grayscale. Examine the image histogram using the function 'histogram' and using a varying number of bins; Perform histogram equalization and show the result. In both cases, show both the resulting image and the accompanying histogram.

   For this task, use Unequalized_H.jpg to see the effect of histogram equalization clearly.

c) Create a 2D blur filter of size 9x9 which sums to 1. Convolve it with the image and display the new image. Create a 1D filter of size 9 which sums to 1. Convolve it with the image. Transpose the filter and convolve it with the **filtered** image. Explain what happens.

d) Now apply the high pass (hp) filter hp=[0 -1 0;-1 4 -1;0 -1 0]. The high pass filtered image will contain negative values, so our result can be better viewed by adding an offset imshow(im_out+128); Comment on your high pass filtering results, comparing with the low-pass case.

Part 2: 2D Discrete Fourier Analysis (Use lab2b.png for this part) - optional

a) Using the grayscale image, compute its 2D DFT magnitude. Display (in a 1x4 subplot grid), the following:
    1. The original grayscale image
    2. Its 2-D DFT magnitude (fft2 for **Matlab** and fft.fft2 in **Python**)
    3. Use the fftshift **(Matlab)** and np.fft.ifft2 (**Python**) function which brings the DC component to the centre of the image, for better visualization.
    4. Apply ifft2 (**Matlab**) and to go back to the image (spatial) domain

b) Adapt properly the codes in (lowpass2_filter.m and butterworth2_filter.m for **Matlab** users and IdealAndButterWorthFilter.py for **Python** users) in order to write a code that chooses one of the filters (Butterworth low/high pass, Ideal Low/High pass) and changes the cut-off frequency and filter order n (in the case of Butterworth).
    - Plot the filter frequency response (using the **mesh(H)** command in Matlab, to visualize the filter in 3D) as well as the filtered image(s). The filters, in the frequency domain can have the same size as the original images (so no need for zero padding). Find a proper pixel transformation for proper visualization of the results!

c) How would you convert the High Pass filters to get the Low-Pass ones in one line of code?

d) Comment on the differences among the different responses to different filters