Computer Vision Course, 2023
Exercise Lab 03-Homework

Problem 1: Image filtering, enhancement, and edge detection.:

1. Study Matlab/Octave functions *imread, brighten, contrast, histeq, imcontrast and imadjust*. Write a script *filtering.m* that applies each one of these functions to image *peppers.png*. Plot the original and the transformed images.

2. Study the functions *imnoise, medfilt2, conv2, filter2, fspecial, imfilter, and edge*. Write a script *filtering1.m* that does the following. Load image *peppers.png*. Convert it from RGB to grayscale using the function *rgb2gray*. Add salt and pepper noise to the image. Filter the resulting image using a 3x3 mean filter, a 3x3 median filter, and Gaussian filter. Repeat, but this time add Gaussian noise with $\sigma = 1$ in the [0, 255] range ($\sigma = 1/256$ in the [0, 1] range) instead of salt and pepper noise. Plot each one of the images and comment on what works best.

3. Write a script *filtering2.m* that does the following. Load image *peppers.png*. Convert it from RGB to grayscale using the function *rgb2gray*. Find the edges in the image using the Matlab/Octave function *edge*. Use the following methods: *Sobel, Prewitt, Roberts, Laplacian of Gaussian and Canny*. Compare your results

Problem2: Color-based face detection: One way to detect faces in color images is to search for pixels that have a skin-like color. In this exercise, you will implement this simple color-based faced detection algorithm.

1. Study the following MATLAB functions *colormap, hsv2rgb, rgb2gray, rgb2hsv, rgb2ntsc, and rgb2ycbcr*.

2. Use image *picture.jpg*. Convert the RGB coordinates to HSV coordinates and plot the following grayscale images: R, G, B, H, S, V. Notice on what you see, e.g. in which regions of which image skin color is more visible.

3. Convert the RGB coordinates to normalized rgb coordinates using following equations:

$$r = R./(R + G + B + 1e - 10);$$
$$g = G./(R + G + B + 1e - 10);$$
$$b = B./(R + G + B + 1e - 10);$$

   Plot g versus r. Find a region of the r-g space that corresponds to the face colors. That is, find a function f such that face pixels can be determined by a rule of the form:

$$\text{pixel}(x, y) \text{ is a face pixel if } f(r(x, y), g(x, y)) \geq 0$$

You can do this by trial and error (we will learn automatic methods later in the course). For instance, choose a function f, define the mask: $M(x, y) = 1$ if $f(r(x, y), g(x, y)) \geq 0$, otherwise $M(x, y) = 0$, and plot the image $J(x, y) = I(x, y)M(x, y)$. Repeat this until faces are "correctly" detected in $J(x, y)$.

4. Repeat part (3) using the hue and saturation coordinates of the HSV color representation. If $H(x, y)$ and $S(x, y)$ are respectively the hue and saturation at pixel (x, y), then the rules to determine a face pixel may be of the form $a \leq H(x, y) \leq b$ and $c \leq S(x, y) \leq d$.

5. Apply the rules you learned in parts (3) and (4) to the test image named *friends.jpg* and image of your class.