Assignment 3 (50 points)

Histogram Equalization *(10/15)*

a) Read and display an image of your choice. (To help you with the next problem, look for an image that has interesting objects of different intensities.)

b) Calculate and display the histogram of this image.

c) Enhance the contrast of the intensity image using histogram equalization and display both the uniform histogram and the newly enhanced intensity image.

d) Explain why the two histograms (of the original image and of the enhanced image) are different.

481 Students: ***(10/0)*** Apply a local enhancement approach on this image and show your results. Before you start, consider how your image might call for a particular window size. For fun, you might want to try a few different window sizes. One student actually put the window size in a loop from 1 to the image size and showed the results in a video. The gauntlet has been thrown. ☺

Histogram-based segmentation *(20/30)*

Implement histogram based segmentation on your image as follows:

a) Show your image.

b) Display the histogram and identify the peaks of your histogram with the “objects” that they correspond to.

c) Specify the ranges that you will use to identify the binary objects.

d) Show the identified objects as binary images for each range. (Remember to scale the images for display so that objects can be seen.)

e) Finally construct the histogram-based segmented image, by combining the binary images.

Bit Plane Splicing *(15/0)*

Bit place splicing (**https://en.wikipedia.org/wiki/Bit_plane**) is a simple form of frequency analysis in which the frequencies are defined by the bits representing the intensity of the pixels. Write a program to perform bit-place splicing on an image such that you can generate a figure similar to the one shown in the Wikipedia article: your original image and each of the 8 bit planes in it. Perhaps the key lesson is that each bit-position represents a different binary image.

General submission instructions:

(a) Be kind to your aging, over-worked professor and submit only a single document. This can be pdf, MS Word, OpenOffice, etc. Do not submit a zip file.

(b) Your single document should include the input image for your problem, if required, and answers to each of the sub-problems (text, image or both, as appropriate). Your document should also include code that you wrote to generate your answers.

(c) You may use any images you like for the programming; I encourage you to use images that might be useful/interesting for your final project.

(d) Feel free to use whatever functions MatLab supplies. Also feel free to write your own, if you are so inclined; it will take more time, but you will gain a deeper understanding of the material.

(e) Point values for each question are indicated as ***(x/y)*** in which ***x*** is the point value for 481 students and ***y*** is the point value for 381 students.