Lab Assignment #4
Due Monday, November 7 by 11:59pm
(upload as a single PDF to CatCourses)

Title: Histogram Equalization
In this lab, you will implement a function that performs histogram equalization on a grayscale image. You can assume all images are grayscale and have values from 0 to 255.

Read through this assignment in detail before starting.

The lab consists of two experiments:

1. Performing histogram equalization on a dark image to improve how it looks.
2. Performing histogram equalization on a light image to improve how it looks.

See the Python script `test_HistogramEqualization.py` which performs these two experiments. You can use this script. Your task in this lab is to write the Python functions `compute_histogram()` and `equalize()` in the file `My_HE_functions.py`.

Read through the Python script `test_HistogramEqualization.py` to see how it:

- Reads the dark image that you will be applying histogram equalization to: `Lab_04_image1_dark.tif`.
- Shows the image.
- Creates a numpy matrix from the dark image so we can access its pixels. NOTE THAT WE WE ARE CREATING A FLOAT32 ARRAY SINCE WE WILL BE DOING FLOATING POINT OPERATIONS IN THIS LAB.
- Calls the function `compute_histogram()` to compute the normalized histogram of the dark image. We will be using a length 256 numpy vector to represent the histogram of a grayscale image. You will need to write this function.
- Calls the function `plot_histogram()` to plot the histogram of the dark image. This function is provided for you.
- Prints the mean and standard deviation of the pixel values in the dark image.
- Calls the function `equalize()` to apply histogram equalization to the dark image. This function returns a numpy matrix with the pixel values of the equalized image. You will need to write this function.
- Calls the function `compute_histogram()` to compute the normalized histogram of the equalized dark image.
- Calls the function `plot_histogram()` to plot the histogram of the equalized dark image.
- Prints the mean and standard deviation of the pixel values in the equalized dark image.
- Repeats the above for the light image `Lab_04_image2_light.tif`.

In the file `My_HE_functions.py`, write the following functions:

1. Write the function `compute_histogram()` that takes a numpy matrix representing a grayscale image as input and outputs a length 256 numpy vector representing the *normalized* histogram of the image. Here's the code I used to create this vector

```
hist = np.zeros(shape=(256))
```

   Once you have calculated the values of the histogram vector, these values should range from 0 to 1 and should sum to 1.

Be aware that when you use the pixel values to index into the histogram vector, you will probably need to typecast them as integers. For example,

```
hist[int(image_pixels[m][n])] += 1
```

If you don't do this, you will likely get the error "`IndexError: only integers, slices (`:`), ellipsis (`...`), numpy.newaxis (`None`) and integer or boolean arrays are valid indices.`"

2. Write the function `equalize()` that takes a numpy matrix representing a grayscale image as input and outputs a numpy matrix representing the histogram equalized version of the image. This function should:

    - Compute the normalized histogram of the input image by calling your function `compute_histogram()`.
    - Create a length 256 numpy vector representing the transformation function that you will apply to the pixels of the input image to create the output image.
    - Compute the entries in this transformation vector using the normalized histogram computed above. This vector can be considered a look-up-table that you index into with the input image pixel value and the value at that index is the output image pixel value. T should be computed using equation 3.3-8 (3rd edition) or equation 3-15 (4th edition) in the text.
    - Create a numpy matrix representing the pixel values of the output image. This should be the same size as the input image.
    - Use the transformation vector to transform each pixel in the input image to the pixel in the output image.

Once you have written the two functions `compute_histogram()` and `equalize()` you should be able to run the provided Python script `test_HistogramEqualization.py`. Here is the terminal output that I get when I run it:

```
Dark image has mean = 82.816910 and standard deviation = 60.353374
Equalized dark image has mean = 128.906438 and standard deviation = 73.687541

Light image has mean = 182.023697 and standard deviation = 39.150688
Equalized light image has mean = 129.067739 and standard deviation = 74.119361
```

Your values might not be exactly the same as mine but they should be close.

Note/remember:

- When the function `plot_histogram()` uses the matplotlib library to display the histogram, you will need to close the figure window for the script to keep running.
- To save a histogram plot to file, click the save/disk icon in the matplotlib window.
- Provide comments for key lines of code.
- Provide headers for your functions. Headers are comments which appear right after the function declaration that summarize what the function does its calling syntax, the input and output parameters, and the function history. You must provide headers for the two functions you will write: `compute_histogram()` and `equalize()`.

Questions:

1. Discuss if you think the histogram equalized versions are visual improvements over the dark and light images.
2. Discuss how this improvement is reflected in the differences between the histograms of the dark/light images and the histograms of their equalized versions.
3. Discuss how this improvement is reflected in differences between the mean and standard deviations of the pixel values in the dark/light images and the mean and standard deviations of the pixel values in their equalized versions.
4. What was the most difficult part of this assignment?

What to submit:

A lab report as a single PDF containing:

- Lab number and title, your name, your lab section, your TA's name, and the date.
- Abstract: In a few sentences, describe the purpose of this lab.
- Figures containing the dark image, the plot of the histogram of the dark image, the equalized version of the dark image, and the plot of the histogram of the equalized version of the dark image.
- A table containing the mean and standard deviation of the pixel values of the dark image and the equalized version of the dark image.
- The above two bullet points for the light image.
- Questions: Your answers to the four questions above.
- The Python script: `test_HistogramEqualization.py` (This could be exactly the same as the one provided or one with some modifications.)
- The Python file `My_HE_functions.py` which contains your two functions: `compute_histogram()` and `equalize()`.

See Lab_04_sample_report.pdf as an example.

What your report will be graded on

- Whether you included an abstract.
- Whether your report includes the figures you were asked to include and they have appropriate captions.
- Whether you included a table with mean and standard deviation of the pixel values of the dark/light image and their histogram equalized versions.
- Whether you answered the assignment questions.
- Whether your report includes the code you were asked to include.
- Whether your code is correct.
- Whether your code is commented and your functions have headers.