CS 6643 **Fall 2020**
Project 1: Extended Otsu's method **E. K. Wong**

Project description: In the original *Otus's method* for automatic thresholding, we week to find threshold t that minimizes the weighted sum of within-group variances $\sigma_B^2(t)$ and $\sigma_F^2(t)$ for the background and foreground pixels that result from thesholding the grayscale image at value *t*:

$$\sigma^2(t) = \omega_B\,(t)\sigma_B^2(t) + \omega_F(t)\sigma_F^2(t)$$

where

$\omega_B(t) = \sum_{j=0}^{t} H(j)$ (probability of pixel being background)
$\omega_F(t) = \sum_{j=t+1}^{G-1} H(j)$ (probability of pixel being foreground)

and $H(j)$ is the normalized histogram (original histogram divided by total number of pixels in the image) and G is the total number of gray level values. Extend the Otsu's method to automatically determine three thresholds t_1, t_2 and t_3 for dividing the histogram of the input image into four intervals, $[0,\ t_1], (t_1, t_2], (t_2, t_3]$ and $(t_3,\ G$-1], so that the weighted sum of the within-group variances of the four resulting regions is minimized.

You can work on the project yourself or you can form a team of two students to work on the project. You or your team can discuss with other classmates on how to do the project but everyone or every team is supposed to do their own coding and turn in their own program.

You can use Python, C++/C, Java or Matlab to implement your program. If you plan to use another language, send me an email first. You are not allowed to use any built-in library functions to implement any steps that you are required to implement, including the forming of the histogram, computation of means and variances. The only library functions you are allowed to use are those for the *reading, writing* and *displaying* of images.

Testing your program: Test images of size *N* X *M* in bitmap (*.bmp*) format will be provided on NYU Classes for you to test your program.

Submit on NYU Classes by due date:

1. A text file that contains the source code. Put comments in your source code to make it easier for someone else to read your program. Points will be taken off if you do not have comments in your source code.
2. The output image files (in *.bmp* format) generated by your program for the test images provided.
3. A PDF file that contains instructions on how to run your program. If your program requires compilation, instructions on how to compile your program should also be provided. Also, copy and paste the output image files and your source code onto the PDF file (to make it easier for us to grade your project.) This is in addition to the source code file and output image files that you have to hand in separately, as described in (1) and (2) above. Below each output image file on the PDF document, write down the threshold values t_1, t_2 and t_3 found by your program.

You can just submit (1) to (3) above as separate files on NYU Classes. No need to put them into a single ZIP file before submission.