CS 6643 **Spring 2025**
Project 1: Extended Otsu's method (Rev 1) **E. K. Wong**

.

Project description: In the original *Otus's method* for automatic thresholding, we seek to find a 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*:

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

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),

$H(j)$ is the *normalized* grayscale histogram, and G is the number of gray level values. Extend the Otsu's method to automatically segment an image into *two, three* or *four* regions, with the background being one of the regions. The number of regions to segment the input image into is an interactive input parameter *N*. For each test image and for each value of $N = 2, 3$ and 4, your program will determine the thresholds and produce a segmentation result (output image.) For two regions, there should be one threshold value *t*. For three regions, there are two threshold values - t_1 and t_2. For four regions, there are three threshold values - t_1, t_2 and t_3. You can use different *colors* or different *gray level values* to represent the regions in the output image. Your program will also output a *histogram* for the input image, the final total variance $\sigma^2(\cdot)$, and the within-group variances $\sigma_j^2(\cdot)$ for $j = 1$ to N.

Testing your : Three color images of size *N* X *M* in bitmap (*.bmp*) format will be provided for you to test your program. First, convert the images to grayscale by using the formula $I = R\text{ound}(0.299R + 0.587G + 0.114B)$, where R, G, and B, are the red, green, and blue components, and *Round* is the round off operator. You are required to write your own code to carry out the conversion. The use of a library function is not allowed. Name the output images by adding "-out" to the image name; e.g., tiger1-out. The output images should be in *.bmp* format.

Implementation: You can use Python, C++/C or Java 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 of the steps that you are required to implement, including the computation of *means* and *variances*. The only library functions you are allowed to use are those for the *reading, writing* and *displaying* of images, and for mathematical operations other than the computation of means and variances. The use of a *histogram* library function is optional. You can use a histogram library function or you can write your own code to compute the histogram.

Team: You can work on the project yourself or you can work in a team of two. You can discuss with your classmates on how to do the project but every team is expected to work on their own coding and turn in their own project.

Submit on BrightSpace by the due date:

1. Your source code file. 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.
2. The output image files (in *.bmp* format) generated by your program for the three test images provided. There should be altogether nine output images.
3. A PDF report that contains:

a) Instructions on how to run your program. If your program requires compilation, instructions on how to compile your program should also be provided.
b) A global histogram for each of the three input images. Copy and paste each output image onto the PDF document. Below (or next to) each output image, put down the threshold values, the final total variance $\sigma^2(\cdot)$, and the within-group variances $\sigma_j^2(\cdot)$ for $j = 1$ to N.
c) Copy and paste the source code onto the PDF document.

Copying-and-pasting of the output images and source code onto the PDF report in (3) above is in addition to the source code and output image files that you have to hand in separately in (1) and (2) above.

If you work in a team of two, only one partner needs to submit but put both partners' names on the source code and the PDF document.