

---

**Project description:** In the original *Otsu'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 thresholding the grayscale image at value  $t$ :

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

$$\omega_B(t) = \frac{1}{N} \sum_{j=0}^t H(j) \quad (\text{probability of pixel being background})$$

$$\omega_F(t) = \frac{1}{N} \sum_{j=t+1}^{G-1} H(j) \quad (\text{probability of pixel being foreground})$$

with  $H(j)$  the grayscale histogram,  $N$  the total number of pixels in the image, and  $G$  the number of gray level values. Extend the Otsu's method to automatically segment an image that contains anywhere from *two to four regions*, with the background being one of the four regions. The number of regions in the input image is unknown ahead of time. For each test image, your program will output the *number of regions* in the image, the *threshold values* as determined by the extended Otsu's method, and the *segmented image results*. You can use different *colors* or different *gray level values* to represent the different regions in the output image.

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

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 step 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, and for performing mathematical operations other than those described above.

**Testing your program:** Test images of size  $N \times M$  in bitmap (*.bmp*) format will be provided on BrightSpace for you to test your program.

**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 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 images 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 on the PDF document, write down the number of regions and threshold values as found by your program.

---

Please submit (1) to (3) above as separate files on BrightSpace. Do not put them into a single ZIP file or other types of compressed files. If you work in a team of two, only one partner needs to submit but put both partners' names on the source code and PDF document.