Segmentation Lab 2

Objectives:

- Learn adaptive thresholding technique(s).

Experiment 1:

In this experiment, we are going to implement a histogram automatic thresholding technique. Slightly different from the technique explained in the lectures.

You are required to implement a function getThreshold: that takes an image and calculates the grey-level threshold according to the following algorithm.

1- Convert the image to Unit8. → *Hint: multiply the matrix by 255 and the use astype('uint8').*
2- Get counts array that contains the number of pixels in each grey-level.→*Hint:use your previously implemented function that calculated histogram.*
3- Get initial threshold by getting the average mean of the pixels' grey-level:

   a. $T_{init} = round((\sum_{NumberOfGreyLevels k=0} greyLevel_k * NumberOfPixels_k) / total\ number\ of\ pixels\ in\ the\ image.)$

   → *Hint: you can get total number of pixel by using the cumulative sum and find last element of it.*

   →*Hint to get the last element of a vector, use the index -1 like M[-1].*

4- Get the two weighted averages (means), one for the lower pixels (have gray level less than Tinit) and the other for the higher pixels (have gray level more than Tinit).
   →*Hint: use the same method you got the Tinit to calculate the mean, just change the gray levels range you are working on.*
   Hint: example to get the range from 0:9 → L=list(range(0,10))
5- Update the New threshold to be the average of the two means.
6- Repeat the same steps [4:5] until threshold saturates.

Pixels with grey-level less than threshold will be equal to pure black and otherwise pixels will be equal to white (0 and 255 for unit8).

Conduct your experiments on “cameraman.png”, “cufe.png” ,”book1.png”and “book.png”.

Experiment 2

Local adaptive thresholding technique applies the adaptive thresholding technique on sub parts of the image to find different local thresholds rather than one global threshold, which may in some images achieve better results.

You are required to partition the image “book.png” to four quarters, apply the adaptive threshold you have implemented in experiment 1 on each quarter alone, and then gather the four quarters together again.

For this exercise show

1. The original image.
2. The image after applying global threshold to it.
3. The whole image after applying local thresholds to it.

Experiment 3 (Bonus):

Improve the output more. [Justify your choice]

Name	Function / Attribute	Usage
np.sum	Function	Summation of matrix or vector
np.multiply	Function	Multiply two matrices element-wise

round	Function	Round a number.
int	Function	Converts to integer
list(range(a,b))	Function	Get list starting from a to less b -1
//	Operator	Integer division
np.cumsum	Function	Get cummulative sum of a matrix