Lab 3: Thresholding and Segmentation

(10% of final score, due by Feb. 26 11:59 PM)

In the following tasks, you can use your input images to test your codes or pick up images from skimage.data() module: [https://scikit-image.org/docs/dev/api/skimage.data.html]

Upload your results (put all scanned handwritings or typed contents in one WORD or PDF document) and **.py* files to the folder of **Lab 3** under **Assessments > Assignment** in the D2L system. Name the **.py* files according to the series numbers of the questions. The **.py* files should follow the PEP-8 Style Guide for Python Code. [https://www.python.org/dev/peps/pep-0008/]

1. Thresholding. Compute the optimal threshold **by hand** for the following grayscale image using (a) the Ridler-Calvard (also known as inter-means) algorithm, and (b) Ostu's method:

$$\begin{bmatrix} 6 & 5 & 8 & 7 \\ 4 & 2 & 3 & 8 \\ 1 & 8 & 6 & 1 \end{bmatrix}$$

   (c) Find the two algorithms in { skimage.filters } module, use the corresponding functions to test your results in (a) & (b). Moreover, apply the two functions to a graylevel image. Which method works better? Why?

   (d) Compare the two functions with another one { skimage.filters.threshold_local }. Which method works better? Why?

   The pseudocode for (a) and (b) are provided for reference. Not mandatory to use the pseudocode. You can just call functions in Python.

ALGORITHM 10.1 Compute an image threshold using the Ridler-Calvard algorithm

RIDLER-CALVARD(I)

Input: grayscale image I
Output: threshold value τ

$h \leftarrow$ COMPUTEHISTOGRAM(I)
$m_0[0] \leftarrow h[0]$
$m_1[0] \leftarrow 0$
for $k \leftarrow 1$ **to** $\zeta - 1$ **do**
    $m_0[k] \leftarrow m_0[k-1] + h[k]$
    $m_1[k] \leftarrow m_1[k-1] + k \cdot h[k]$
$\tau \leftarrow \zeta/2$ ➢ reasonable initial value, but not important
repeat
    $\mu_{\blacktriangleleft} \leftarrow m_1[\tau]/m_0[\tau]$
    $\mu_{\triangleright} \leftarrow (m_1[\zeta-1] - m_1[\tau])/(m_0[\zeta-1] - m_0[\tau])$
    $\tau \leftarrow$ ROUND $(\frac{1}{2}(\mu_{\blacktriangleleft} + \mu_{\triangleright}))$
until τ does not change
return τ

ALGORITHM 10.2 Compute an image threshold using Otsu's method

OTSU(I)

Input: grayscale image I
Output: threshold value τ

$h \leftarrow$ COMPUTEHISTOGRAM(I)
$m_0[0] \leftarrow h[\ell]$
$m_1[0] \leftarrow \ell \cdot h[\ell]$
for $\ell \leftarrow 1$ **to** $\zeta - 1$ **do**
    $m_0[\ell] \leftarrow m_0[\ell-1] + h[\ell]$
    $m_1[\ell] \leftarrow m_1[\ell-1] + \ell \cdot h[\ell]$
$\mu \leftarrow m_1[\zeta-1]/m_0[\zeta-1]$
$\hat{\sigma}_b^2 \leftarrow 0$
for $\ell \leftarrow 0$ **to** $\zeta - 1$ **do**
    $\sigma_b^2 \leftarrow (m_1[\ell] - \mu m_0[\ell])^2/(m_0[\ell] \cdot (m_0[\zeta-1] - m_0[\ell]))$
    if $\sigma_b^2 > \hat{\sigma}_b^2$ **then**
        $\hat{\sigma}_b^2 \leftarrow \sigma_b^2$
        $\tau \leftarrow \ell$
return τ

2. Segmentation. Download the image < violin-and-hand.jpg > from the folder. Use different methods of segmentation to separate the hands and the violin from the background. The methods are geodesic active contour, watershed, SLIC, and Felsenszwalb algorithm. (a) Use proper pre-processing methods before segmentation. (b) Tune the parameters and observe their effects on the results. (c) Evaluate the four methods and summarize your observation.

3. Evaluating segmentation metrics. Read the tutorial: https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_metrics.html#sphx-glr-auto-examples-segmentation-plot-metrics-py

   The tutorial compares three methods of segmentation: watershed with compactness, Canny edge filter, and morphological geodesic active contours.

   a. Briefly describe the algorithms of the three methods of segmentation.

   b. Which two functions are used for the evaluation? Each function outputs two or three metrics. Briefly explain the metrics. For more information, please check the link: https://scikit-image.org/docs/dev/api/skimage.metrics.html

   c. Based on the outputs of the five metrics, what are the conclusions of the evaluation?

   d. Tune the parameter "iterations" in morphological_geodesic_active_contour(): 10, 50, 100, 150, 200, 250. Observe the output segmented image and the output metrics. Are the results improved? Why? Keep increasing the parameter "iterations" to 300, 400, and 500. Are the results improved? Why?

   e. Download the image < fruit_gray.PNG> from the folder, convert it to a gray image, and run the tutorial program on the image. Adjust the parameters of the methods watershed (elevation_map, markers) and morphological_geodesic_active_contour to get an acceptable result close to "true segmentation." Print out the parameters of the two methods.

   f. The second method, Canny edge filter, has a line "im_test2 = ndi.label(remove_small_objects(fill_coins, 21))[0]", explain the functions ndi.label(), remove_small_objects() and the meaning of "21" and "[0]."