# Otsu Threshold

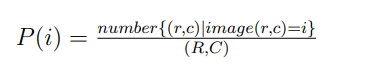
Binarization plays an important role in digital image processing, mainly in computer vision applications. Thresholding is an efficient technique is crucial in binarization.

Otsu's thresholding method corresponds to the linear discriminant criteria that assumes that the image consists of only object (foreground) and background, and the heterogeneity and diversity of the background is ignored. Otsu set the threshold so as to try to minimize the overlapping of the class distributions. according the definition of Otsu , the Otsu's method segments the image into two light and dark regions T0 and T1, where region T0 is a set of intensity level from 0 to t or in set notation T0 = {0, 1, ..., t} and region T1 = {t, t + 1, ..., l − 1, l} where t is the threshold value, l is the image maximum gray level (for instance 256).

The goal is to find the threshold value with the minimum entropy for sum of foreground and background. Otsu’s method determines the threshold value based on the statistical information of the image where for a chosen threshold value t the variance of clusters T0 and T1 can be computed.

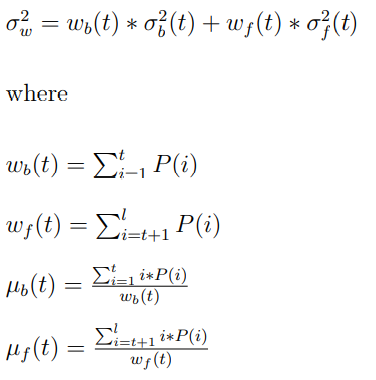
Steps:

* Otsu Global Threshold:
  + calculate the threshold by minimizing the sum of the weighted group variances, where the weights are the probability of the respective groups,



Where r, c is index for row and column of the image, respectively, R and C is the number of rows and columns of the image, respectively. Wb(t), µb(t), and σ 2 b (t) as the weight, mean, and variance of class T0 with intensity value from 0 to t, respectively.

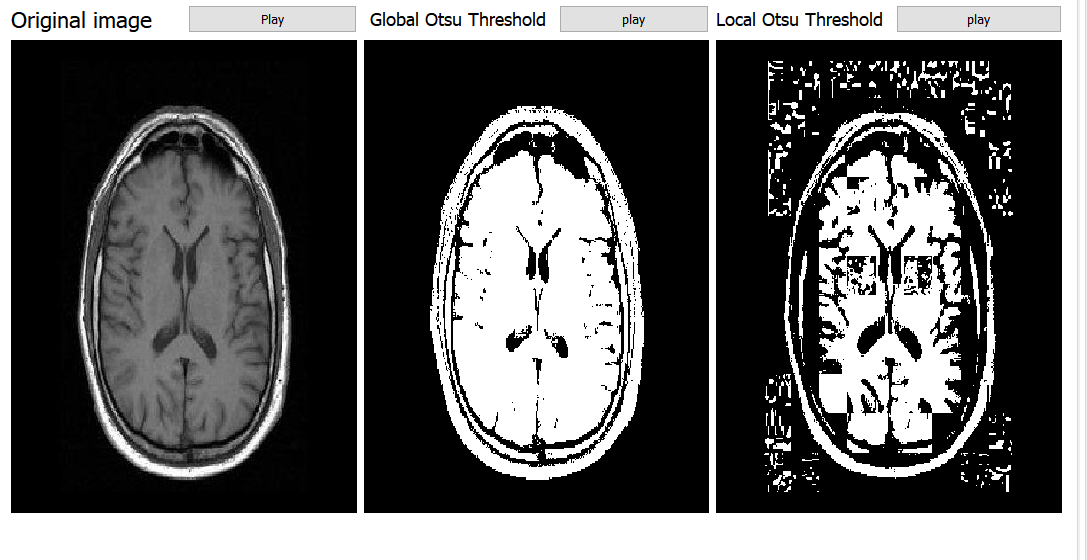
* was the weighed sum of group variances. The best threshold value t\* is the value with the minimum within class variance. The within class variance defines as following:



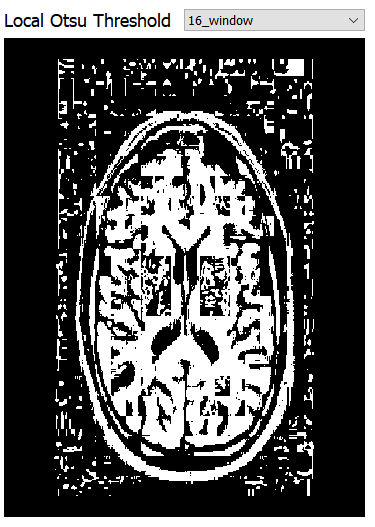
Otsu Local threshold:

* Then, we have to apply this threshold in global threshold as previous steps. But when we need to apply the threshold locally; we split the image to many size of window and get the threshold for each window with Otsu algorithm. Then use each threshold of each window and apply in it and collect these thresholded windows in the original image.
* Note: We make the shape of image to be square to make the small windows applying in all pixels.

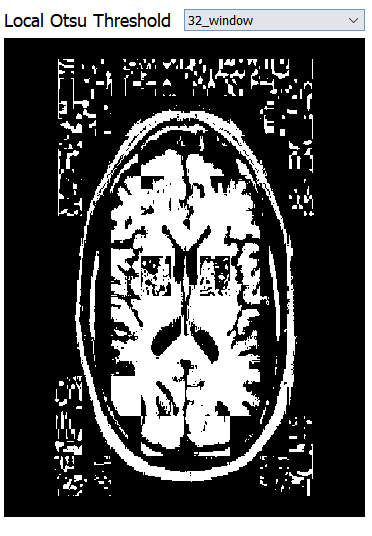
Results:

* when we apply this threshold in "MRIbrain1.jpg", we get:

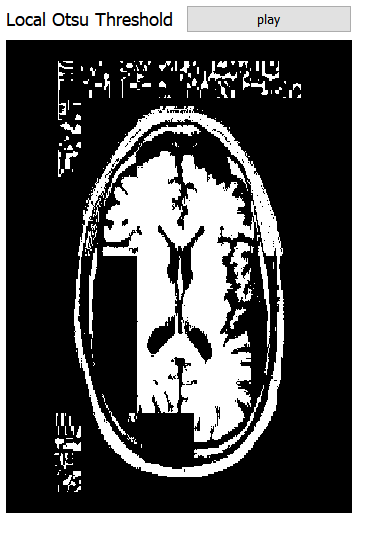
* when we apply Local threshold with different sizes of window as follow:
* **16 window size:**



* **32 window size:**



* **64 window size:**



* **128 window size:**

