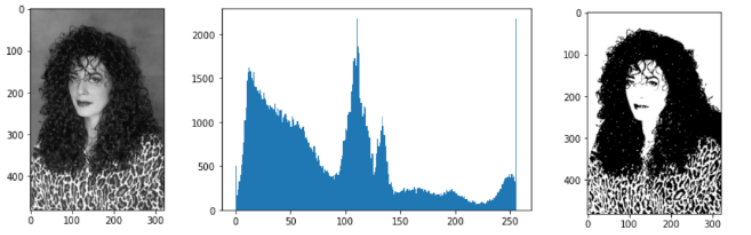
Answer 2:

➢ Implementing a simple thresholding based image binarization algorithm

Steps:

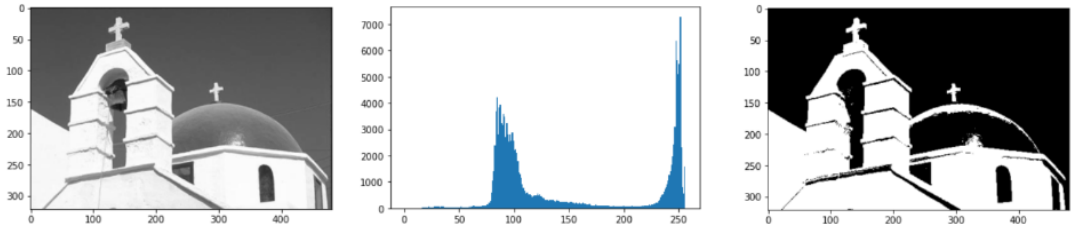
- Read the image
- Convert the image to grayscale, if it is color image
- Display the histogram out of it
- Finding the threshold value in the histogram plot
- Based on threshold, divide the image into two segments black and white

Here, we used three different input images to show it. Please find below the snapshot.



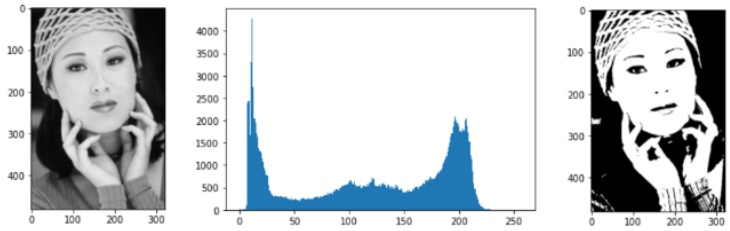
Guess Threshold value = 85

Figure :- Image 1



Guess Threshold value = 150

Figure :- Image 2



Guess Threshold value = 125

Figure :- Image 3

Github link:

https://github.com/SunilDevlops/Programs/blob/master/ComputerVision/ProgramAssignment3/Question2/Performing_Binarization_Image_Segmentation.ipynb

➢ Implementing a Otsu thresholding

Steps:

- This involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background.
- The calculations for finding the foreground and background variances (the measure of spread) for a single threshold can be done in two ways:
  - Within Class Variance

    $$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2$$

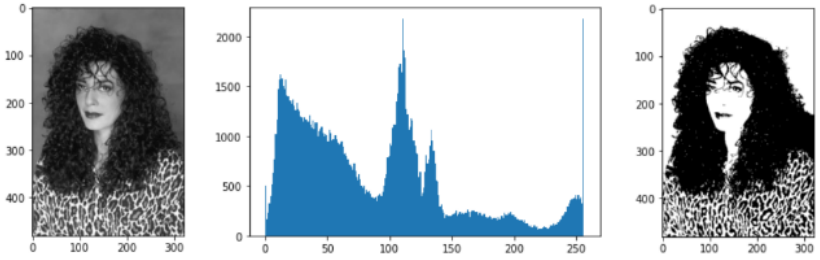
  - Between Class Variance

    $$\begin{aligned}\text{Between Class Variance} \quad \sigma_B^2 &= \sigma^2 - \sigma_W^2 \\ &= W_b(\mu_b - \mu)^2 + W_f(\mu_f - \mu)^2 \quad (\text{where } \mu = W_b\,\mu_b + W_f\,\mu_f) \\ &= W_b\,W_f\,(\mu_b - \mu_f)^2\end{aligned}$$

- We consider the "*between class* variance", which is far quicker to calculate as compare to the other one.
- Finally, the threshold with the maximum "*between class* variance" is considered

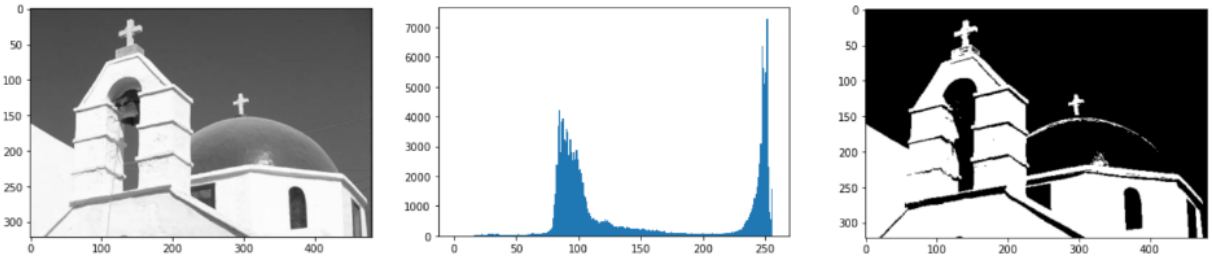
Here, we used same three different above input images to show it.

Please find below the snapshot.



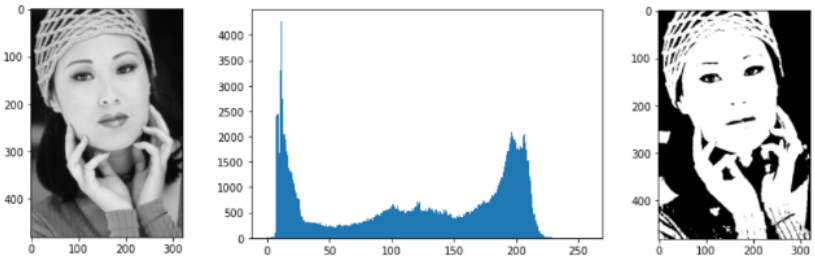
Predicted Threshold value = 93

Figure :- Image 1



Predicted Threshold value = 171

Figure :- Image 2



Predicted Threshold value = 109

Figure :- Image 3

GitHub link:

https://github.com/SunilDevlops/Programs/blob/master/ComputerVision/ProgramAssignment3/Question2/Performing_Otsu_Thresholding_Image_Segmentation.ipynb