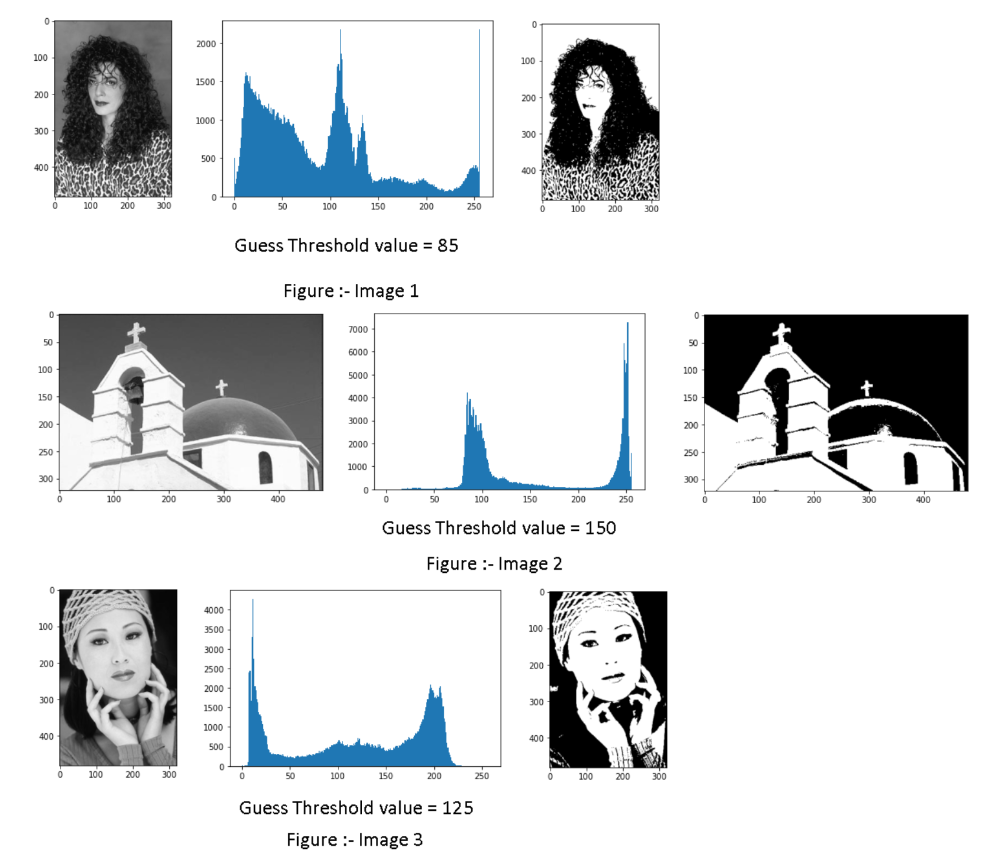
**Answer 2:**

* 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.



**Github link:**

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

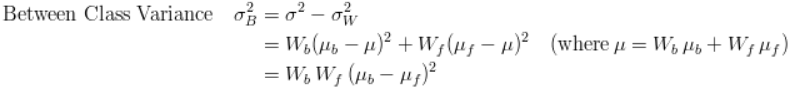
* Implementing a Otsu thresholding

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



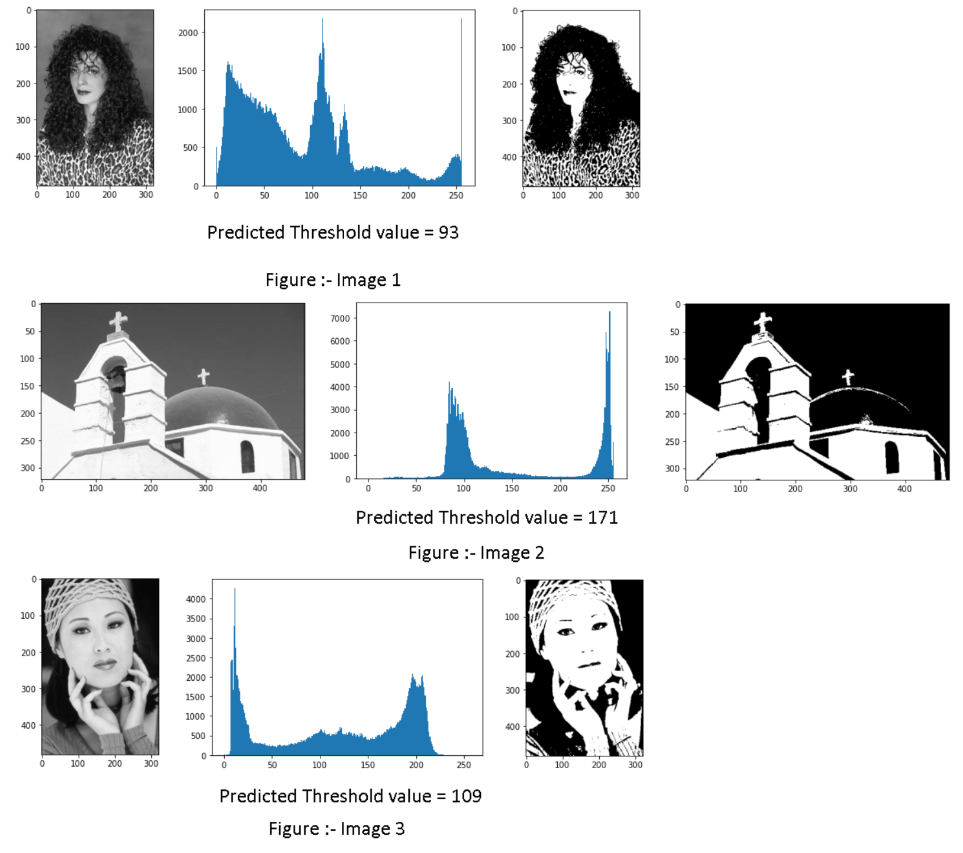
* + - Between Class Variance



* + 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.



**GitHub link:**

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