COMPUTER VISION (CAP5415)-PA-5 REPORT

IMAGE SEGMENTATION

Introduction

We are creating two thresholding-based image segmentation algorithms for this task. In the first, a threshold is manually set for binary image segmentation; in the second, Otsu thresholding is used to automatically extract the ideal threshold from the input image.

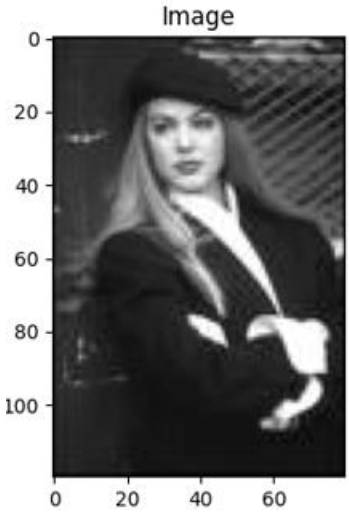
Thresholding Segmentation

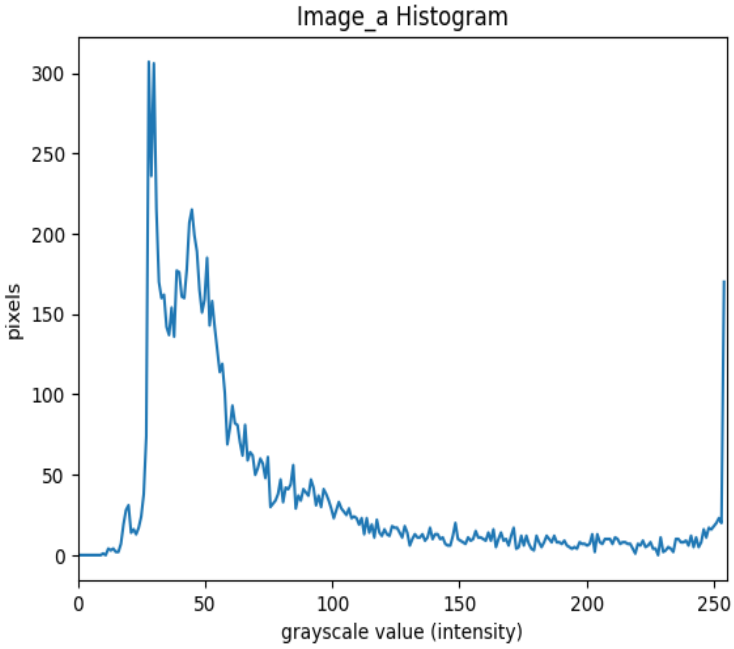
In order to determine the ideal threshold for image segmentation, Otsu's method is an automated binary thresholding algorithm that maximizes the variance between the intensities of the foreground and background pixels. To comprehend the distribution of pixel intensities, the algorithm first computes the image histogram. This distribution is then normalized to obtain a probability mass function. By choosing the threshold that maximizes the between-class variance, the optimal threshold is found, simplifying the binary classification process based on the statistical properties of the image.

Results:

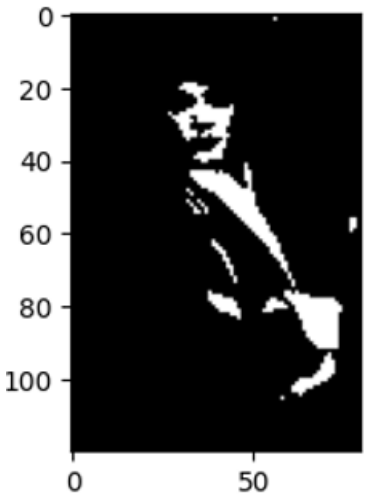
There are 3 different input samples were used to test the binary classifier and Otsu's method; the results are shown in Fig. 1. Four results are shown for each sample, along with the corresponding image histograms. The first three findings deal with binarization segmentation using various thresholds that are obtained from histogram analysis, and the fourth finding is the result of Otsu's thresholding. Each output had a distinct set of individual binary segmentation thresholds, represented by T in Fig. 1. For instance, Otsu determined a threshold of 132 in Image(a), while binary thresholds were set at 89, 121, and 170.

Original Image (a):

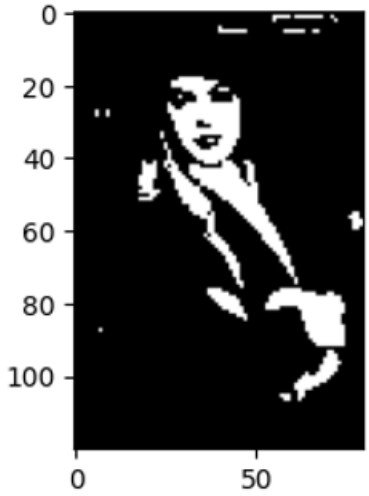




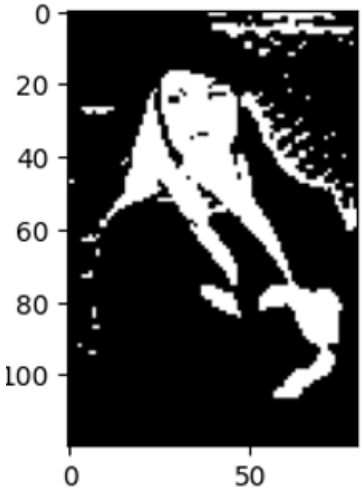
Histogram



Binary Segmentation, T = 90

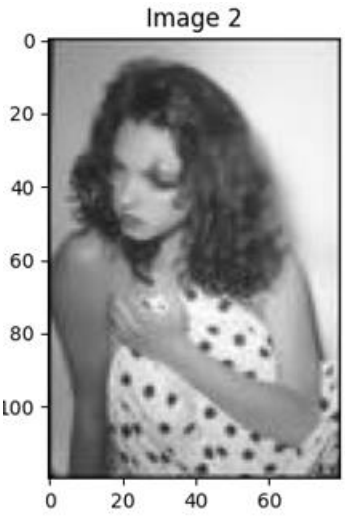


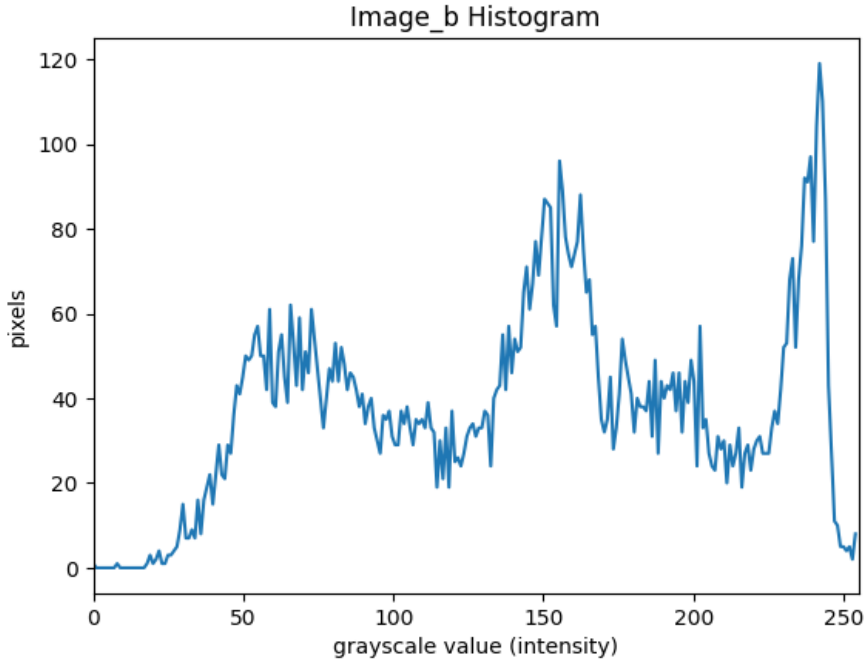
Binary Segmentation, T = 118



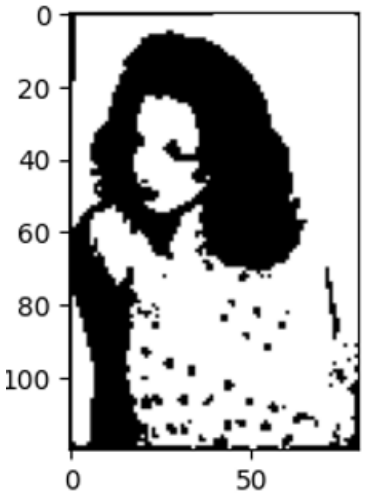
Binary Segmentation, T = 160

Original Image (b):

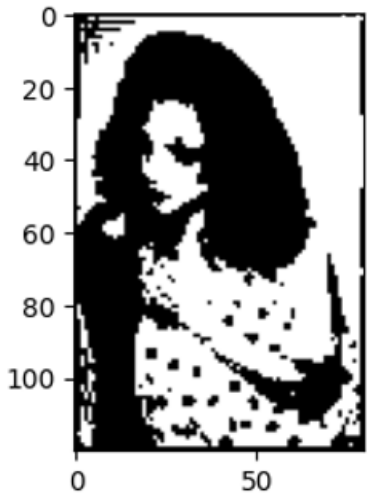




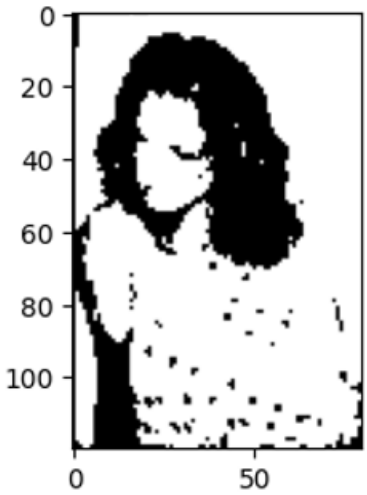
Histogram



Binary Segmentation, T = 50

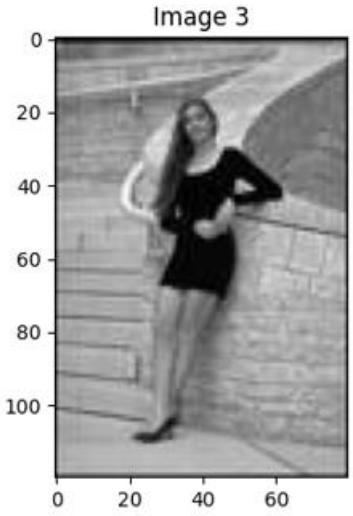


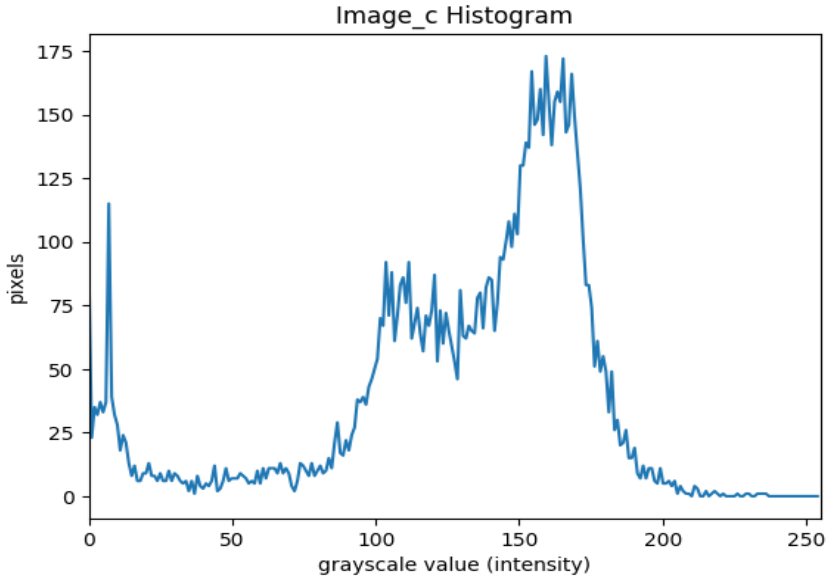
Binary Segmentation, T = 90



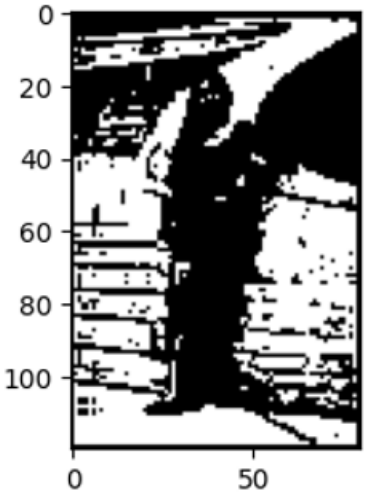
Binary Segmentation, T = 160

Original Image (c):

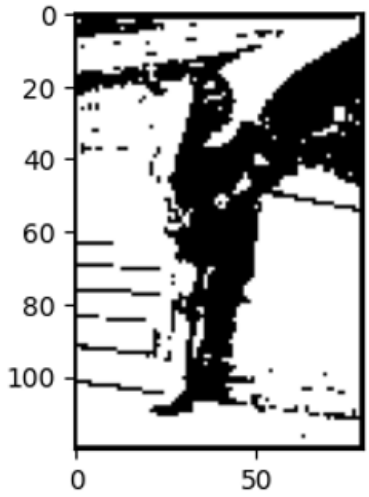




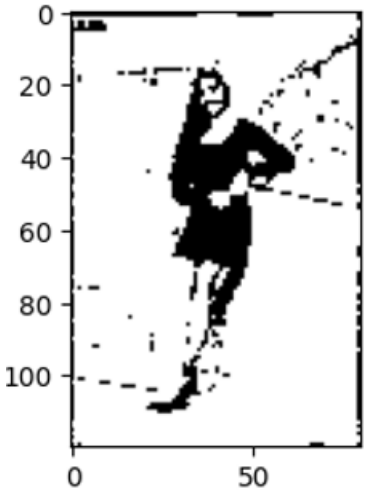
Histogram



Binary Segmentation, T = 85

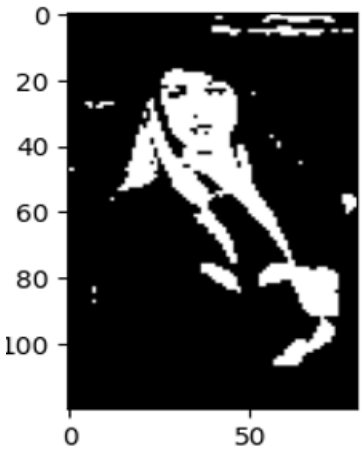


Binary Segmentation, T = 160

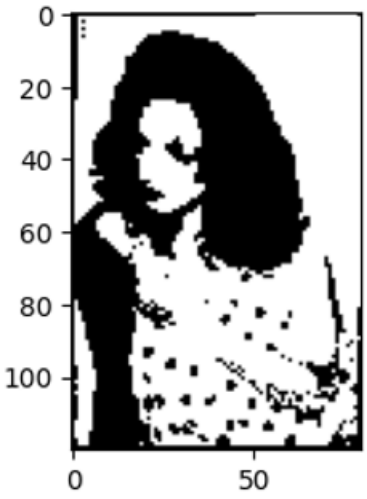


Binary Segmentation, T = 220

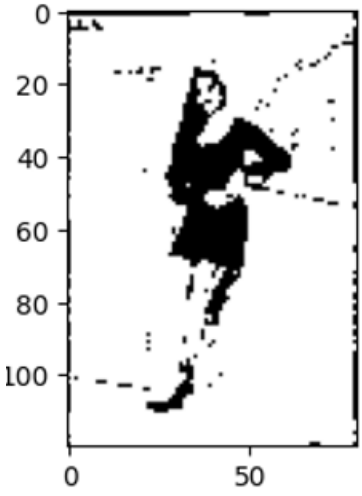
OSTU:



Otsu Segmentation, T = 43



Otsu Segmentation, T = 92



Otsu Segmentation, T = 70

Results:

There are 3 different input samples were used to test the binary classifier and Otsu's method; the results are shown in Fig. 1. Four results are shown for each sample, along with the corresponding image histograms. The first three findings deal with binarization segmentation using various thresholds that are obtained from histogram analysis, and the fourth finding is the result of Otsu's thresholding. Each output had a distinct set of individual binary segmentation thresholds, represented by T in Fig. 1. For instance, Otsu determined a threshold of 132 in Image(a), while binary thresholds were set at 89, 121, and 170.

Conclusion:

Based on the results, we can see that setting the threshold will impact the details that remain in the picture and should set based on the application. However, finding the optimal threshold manually based on local minima is difficult since most histograms have many local minima and we need an extra criterion to find the best local minima among all of them as the threshold. As may be seen from the results, none of the estimated threshold around histogram extrema were close to the Otsu. Therefore, Otsu is a better way of implementing a thresholding technique as it does not require the designer to manually select the threshold.