Programming Assignment 5

Vijay Prakash Reddy Kovuru

CAP5415 Computer Vision

Dec 1st, 2022
Instructor Dr. Yogesh Singh Rawat

Image Segmentation

A. Abstract:

Implementing a simple thresholding based image binarization algorithm and then Implementing the Otsu thresholding to perform image segmentation without using any Python built-in functions.

B. Methods:

Image thresholding is a process for separating the foreground and background of the image. There are lots of methods for image thresholding, Otsu method is one of the methods proposed by ***Nobuyuki Otsu****. The Otsu algorithm is a variance-based way to automatically find a threshold value by which the weighted variance between foreground and background is the least.*

1. Image Binarization

Image binarization applies often just one global threshold T for mapping a scalar image I into a binary image.

The global threshold can be identified by an optimisation strategy aiming at creating "large" connected regions and at reducing the number of small-sized regions, called ***"Artifacts"****.*

2. Otsu Thresholding

The method uses grey-value histogram of the given image I as input and aims at providing the best threshold.

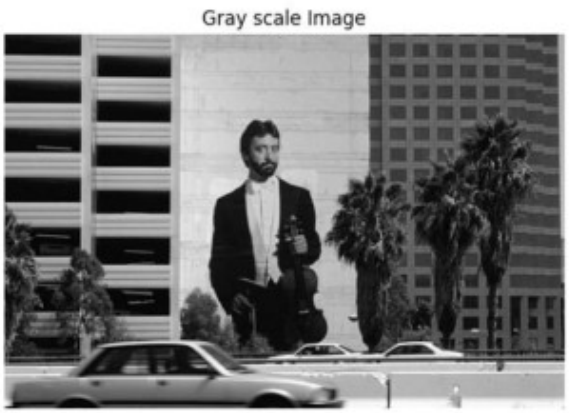
Otsu's algorithm selects a threshold that maximises the between-class variance.

C. Results:

Chosen 3 coloured Images (converting them to greyscale in the code)

For 1st Image results;

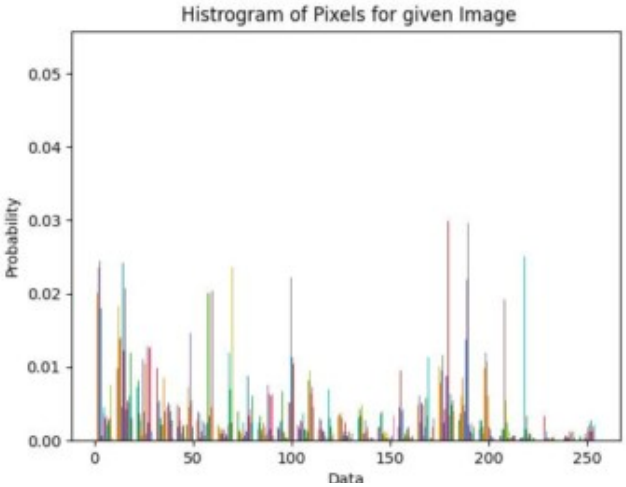
Input Image



Grey scaled Image

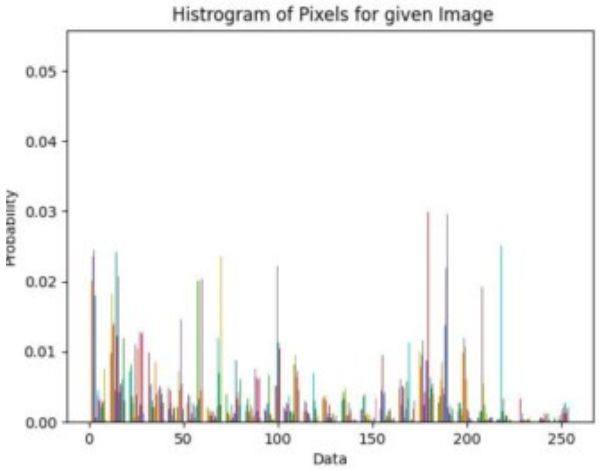
Image Binarization and Histogram at Threshold of 100;

Threshold 100



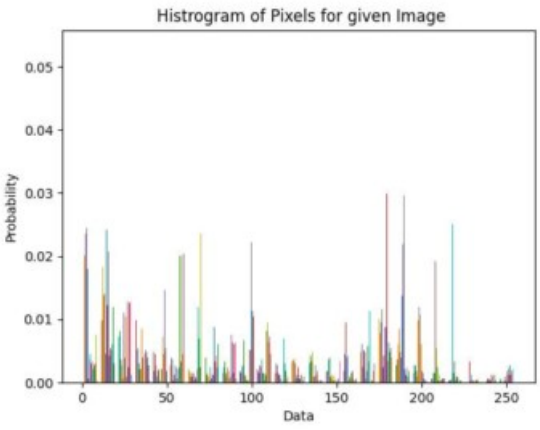
Histogram for Threshold 100

Image Binarization and Histogram at Threshold of 120;



Threshold 120 **Histogram for Threshold 120**

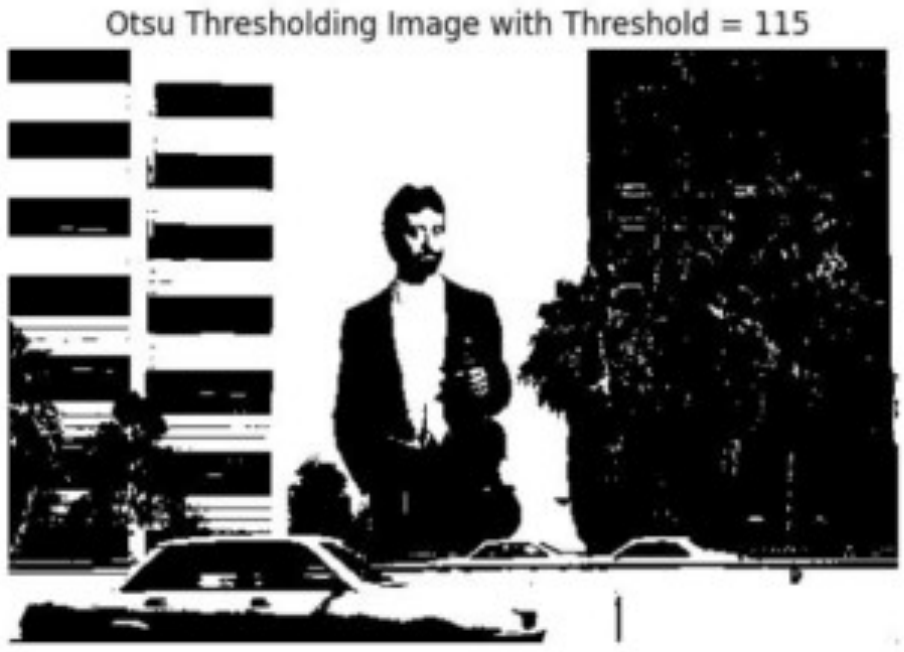
I
mage Binarization and Histogram at Threshold of 140;



Histogram for Threshold 140

Threshold 140

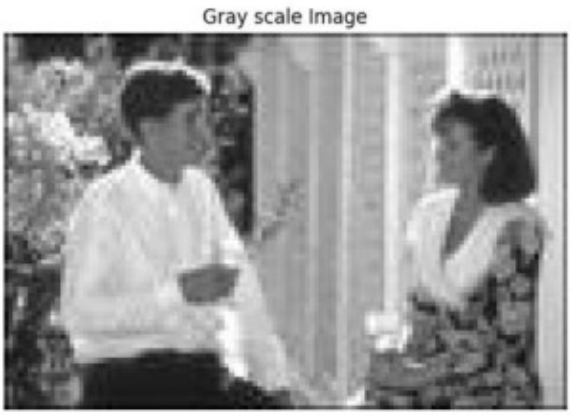
Least Variance can be achieved between object and background classes for given Image after Iterations at a threshold of : 115



Otsu Threshold Image

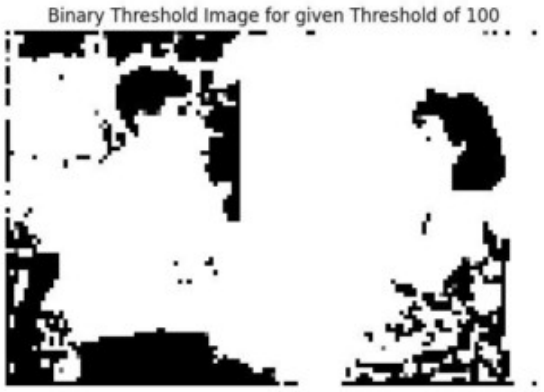
For 2nd Image results;

Input Image 2

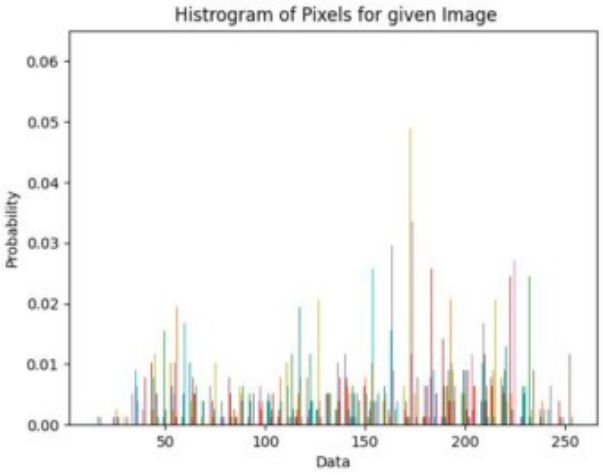


Grey-scaled Image

Image Binarization and Histogram at Threshold of 100;



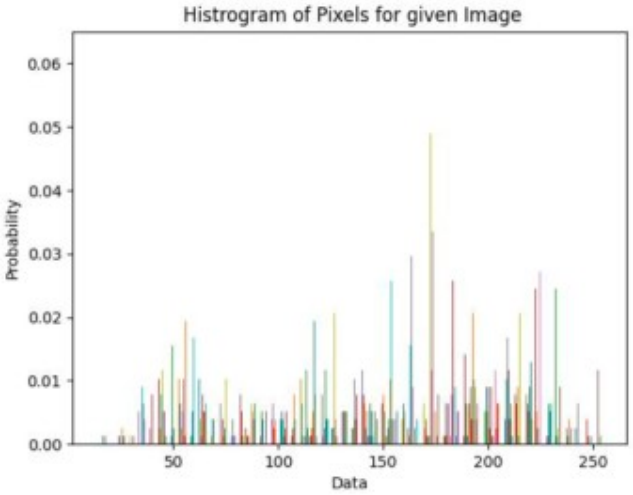
Threshold 100



Histogram for Threshold 100

Image Binarization and Histogram at Threshold of 80;

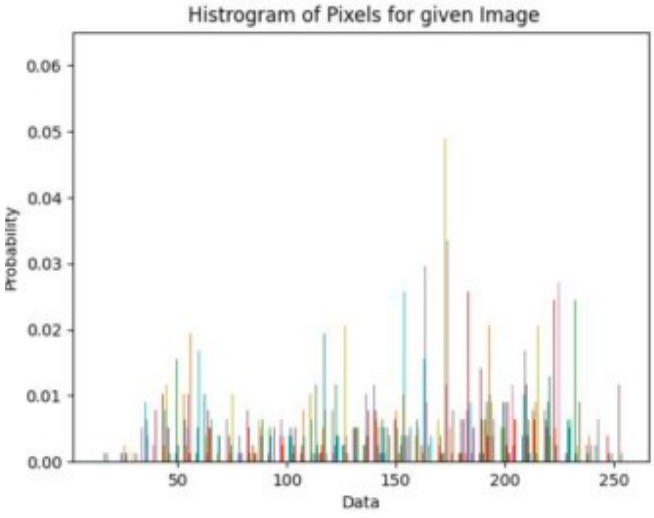
Threshold 80



Histogram for Threshold 80

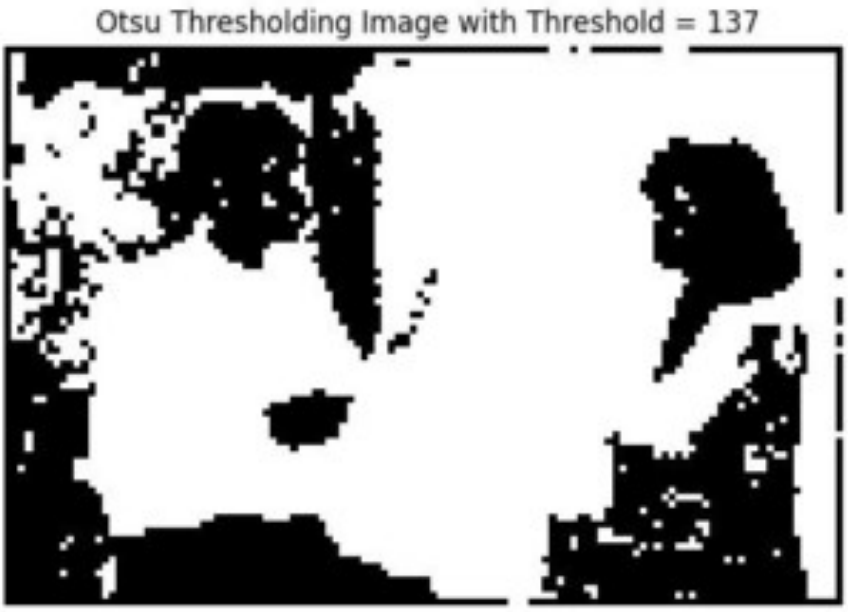
Image Binarization and Histogram at Threshold of 120;

Threshold 120



Histogram for Threshold 120

Least Variance can be achieved between object and background classes for given Image after Iterations at a threshold of : 137



Otsu Threshold

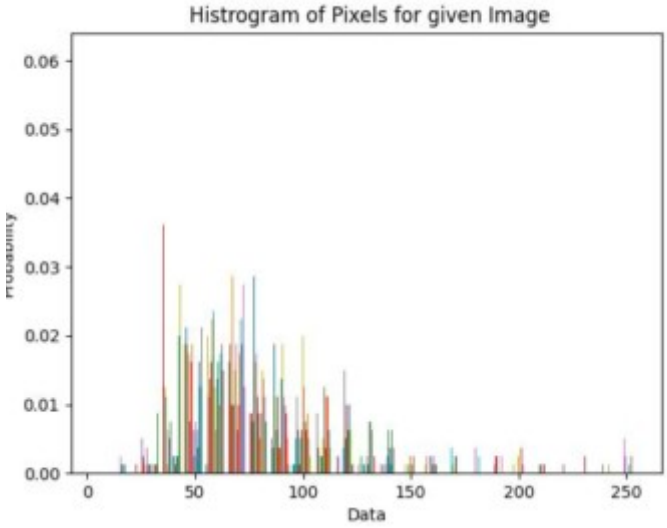
For 3rd Image results;

Input Image

Grey-scaled Image

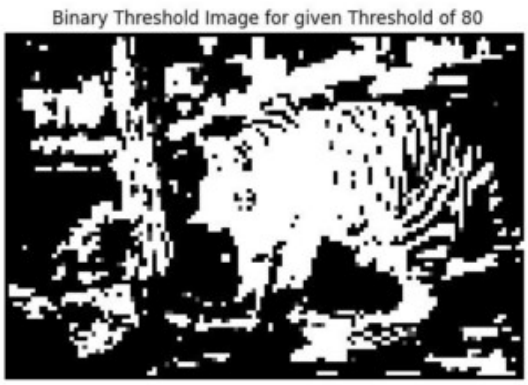
Image Binarization and Histogram at Threshold of 100;

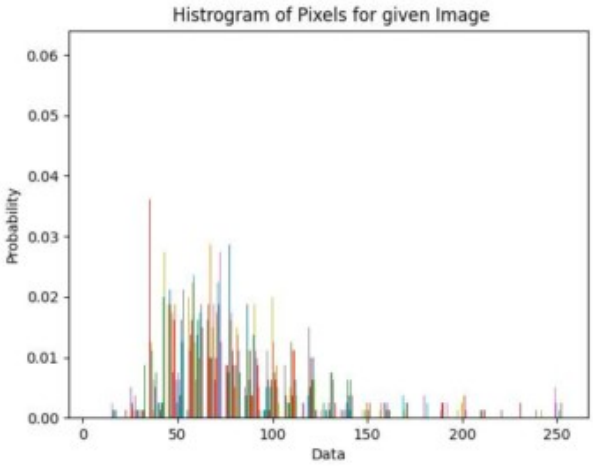
Threshold 100



Histogram for Threshold 100

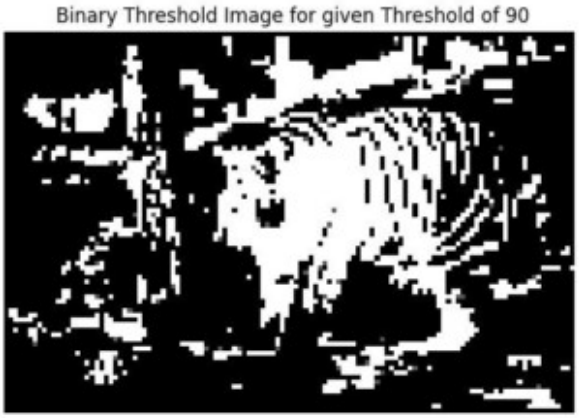
Image Binarization and Histogram at Threshold of 80;

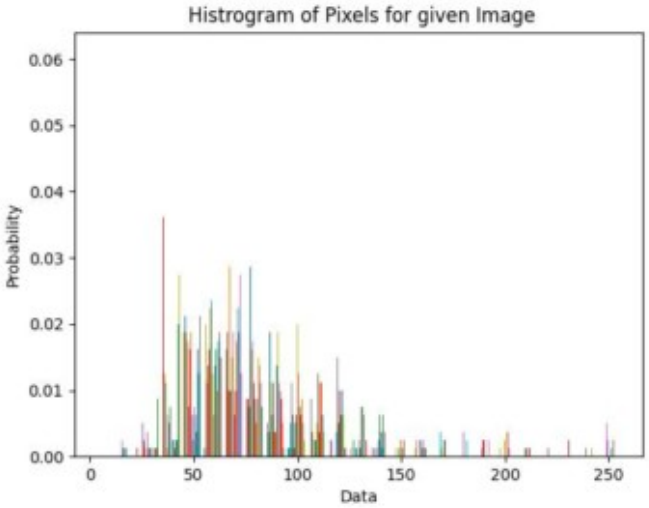




Threshold of 80 ***Histogram for Threshold 80***

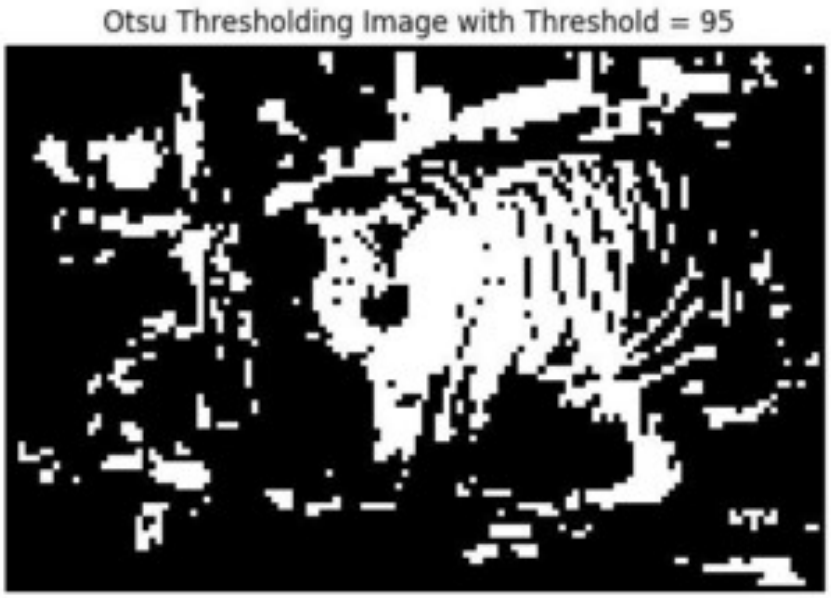
Image Binarization and Histogram at Threshold of 90;





Threshold of 90 ***Histogram for Threshold 90***

Least Variance can be achieved between object and background classes for given Image after Iterations at a threshold of : 95



Threshold 95

Conclusions:

- *As Increasing the Threshold value, the Histogram is **Right Skewed**. i.e,, Black pixel values are getting increased and as decreasing the threshold value, the Histogram is **Left skewed**. I.e., The white pixel values are getting increased in the Image.*

[Github Code Link]

[Website link for Otsu Image Segmentation]

Change the theme of the application to Light mode for better view experience.