CAP 5415 Fall 2023

Programming Assignment 5

Submitted by: Ehtesamul Azim

UCFID: 5460629

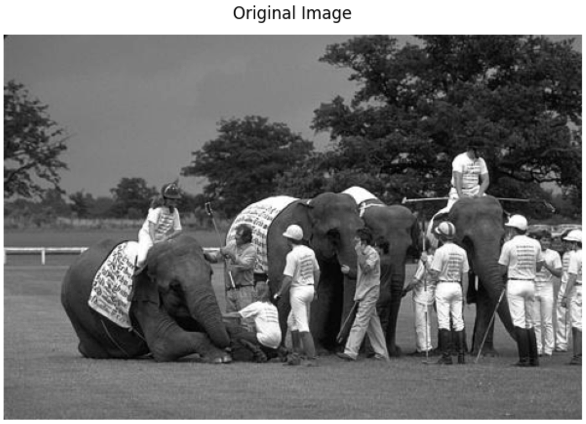
Objective

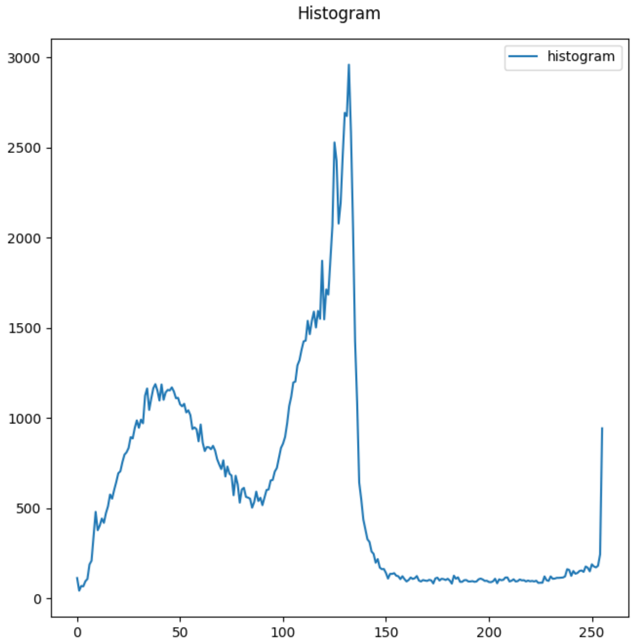
1. To implement image binarization with three thresholds
2. To implement Otsu thresholding

<u>Manual Thresholding:</u>

<u>Image 1:</u>

Original Image in grayscale and its histogram





The following figures will show that the three thresholds chosen from the histogram, resulted in three different binarization scenarios. The middle one is the best balanced among these three. In the last one, the background starts blending into the foreground and in the first one the opposite happens.

Image binarization with three different threshold values (70, 110, and 140)

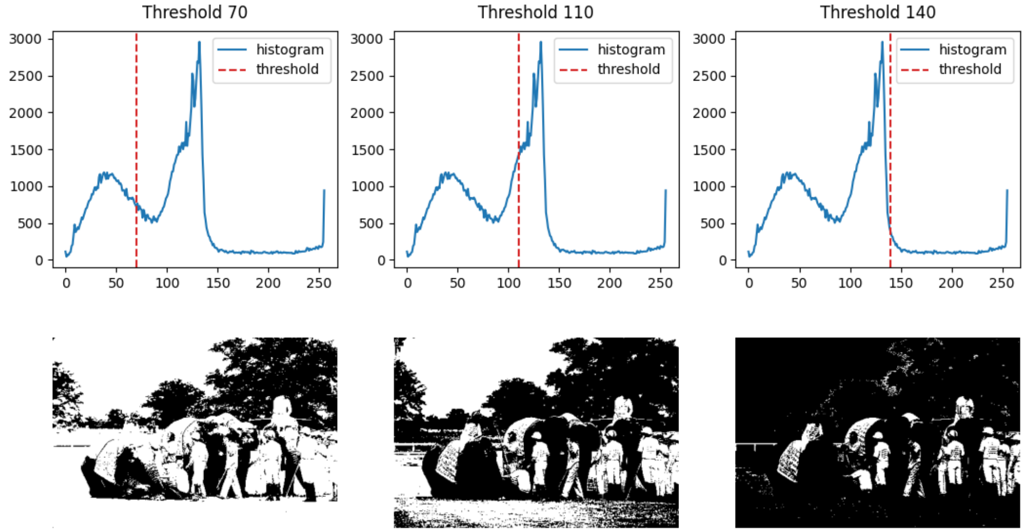


Image 2:

Original Image in grayscale and its histogram

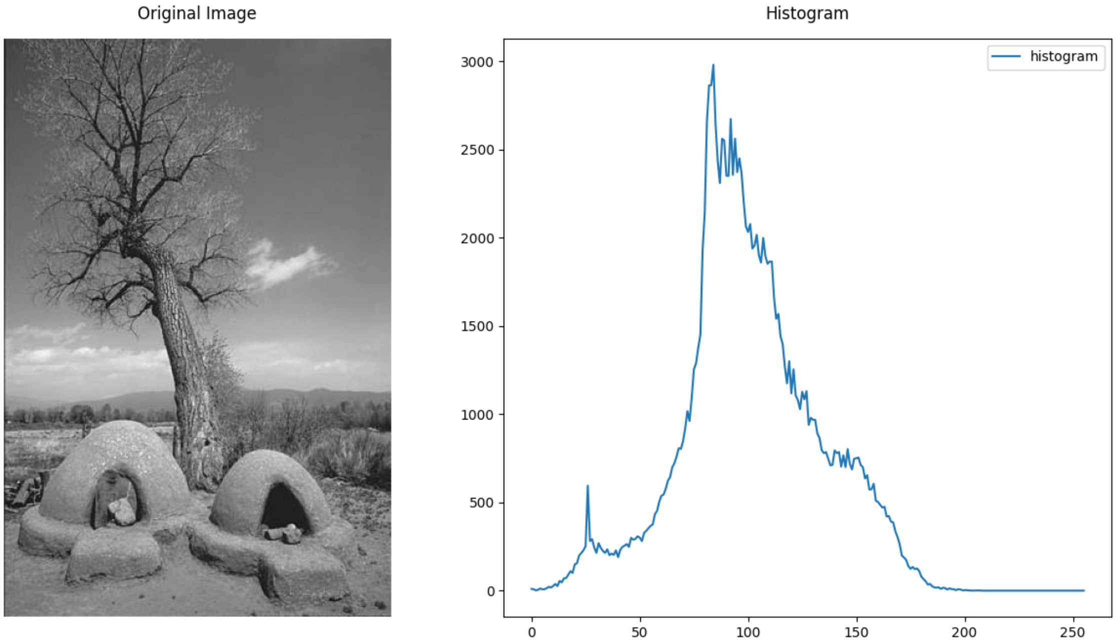
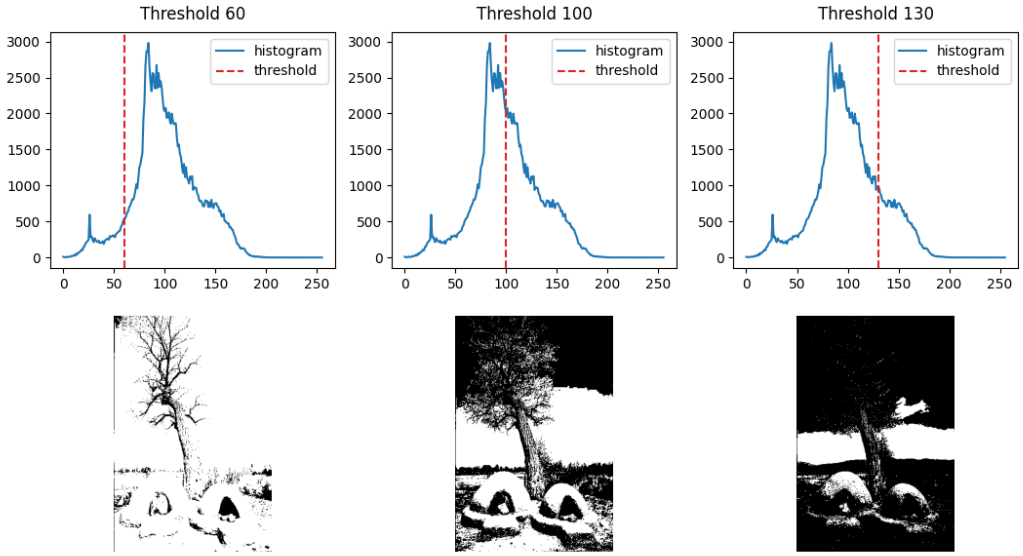


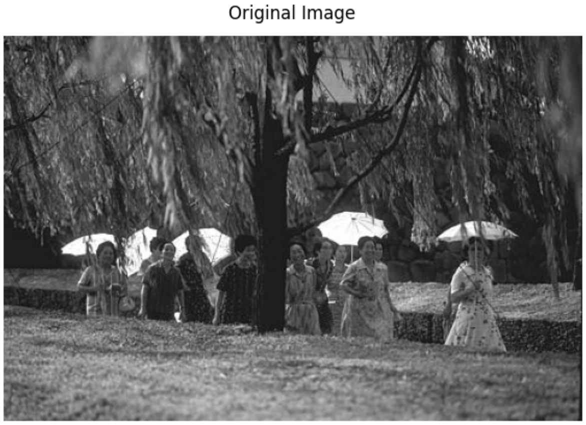
Image binarization with three different threshold values (60, 100, and 130)



In this one, the middle one again seems the most balanced of the three outputs. In the other two, the background and the foreground almost completely blended together.

Image 3:

Original Image in grayscale and its histogram



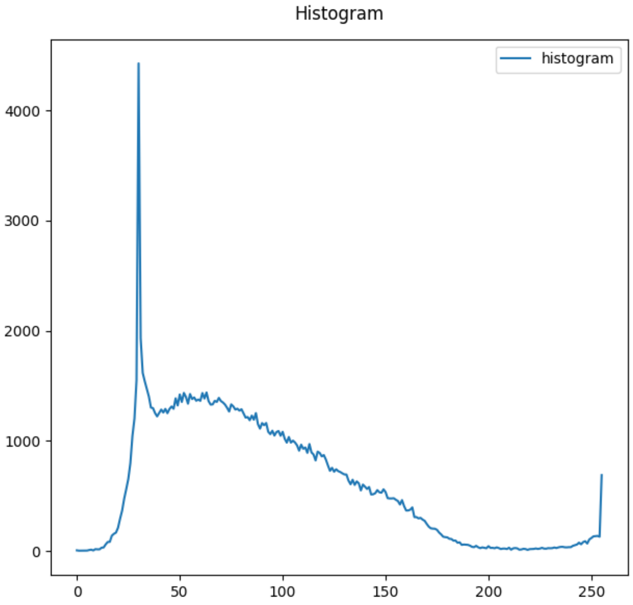
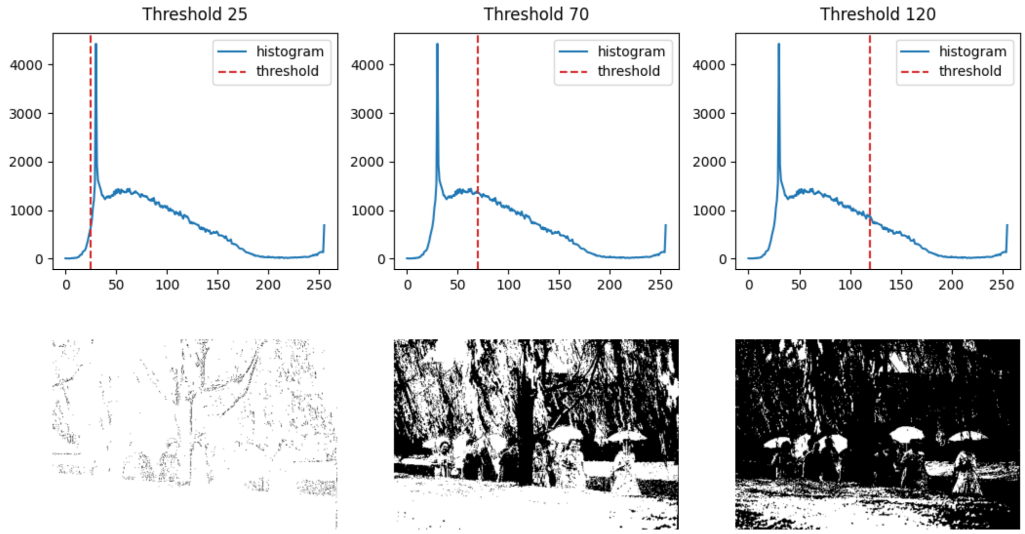


Image binarization with three different threshold values (25, 70, and 120)



The one in the middle looks the most balanced here as well. As threshold increases, the background starts getting overexposed and the foreground starts getting underexposed.

Otsu Thresholding

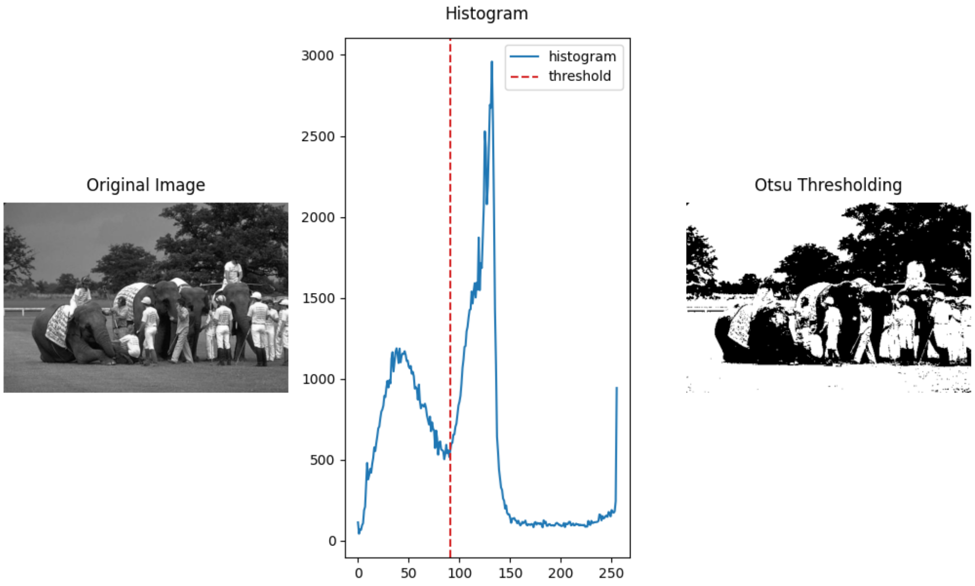
In the provided otsu.py file, these are some key functions I wrote to implement this:

1. **Threshold Image Function:**
   - The `threshold_image` function takes an image and a threshold as input, binarizes the image by setting pixels above the threshold to 1, and returns the resulting binary image.
2. **Compute Otsu Criteria Function:**
   - The `compute_otsu_criteria` function calculates the Otsu criteria for a given threshold. It first binarizes the image using the `threshold_image` function, computes weights and variances for the two classes, and then calculates the Otsu criteria.

3. **Find Best Threshold Function:**
    - The `find_best_threshold` function iterates over a range of possible thresholds, calculates the Otsu criteria for each threshold using `compute_otsu_criteria`, and returns the threshold that minimizes the Otsu criteria.

These functions collectively provide a modular implementation of Otsu's method for image thresholding. The results are given below:

Image 1:



In all the images processed by the otsu thresholding functions, the binarized images look really balanced.

Image 2:

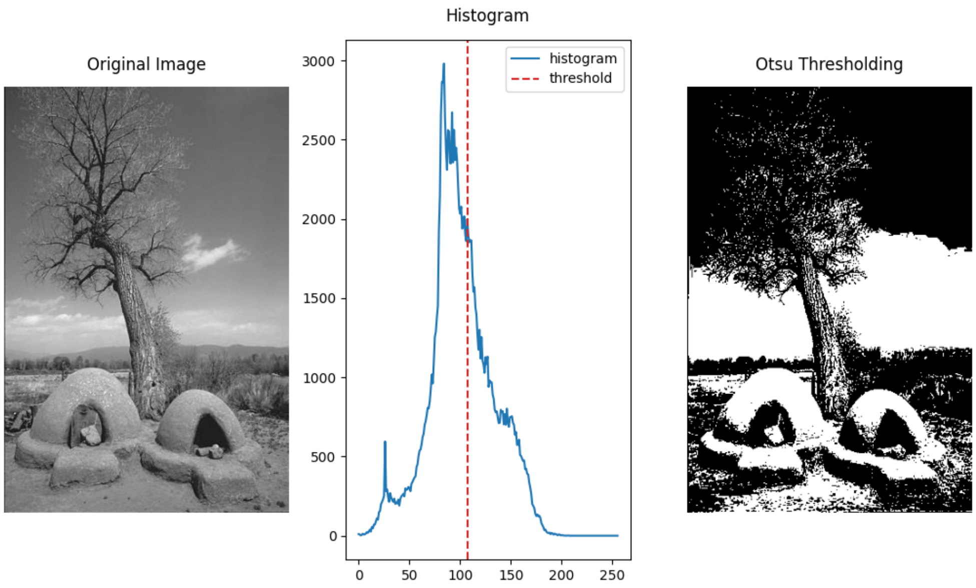
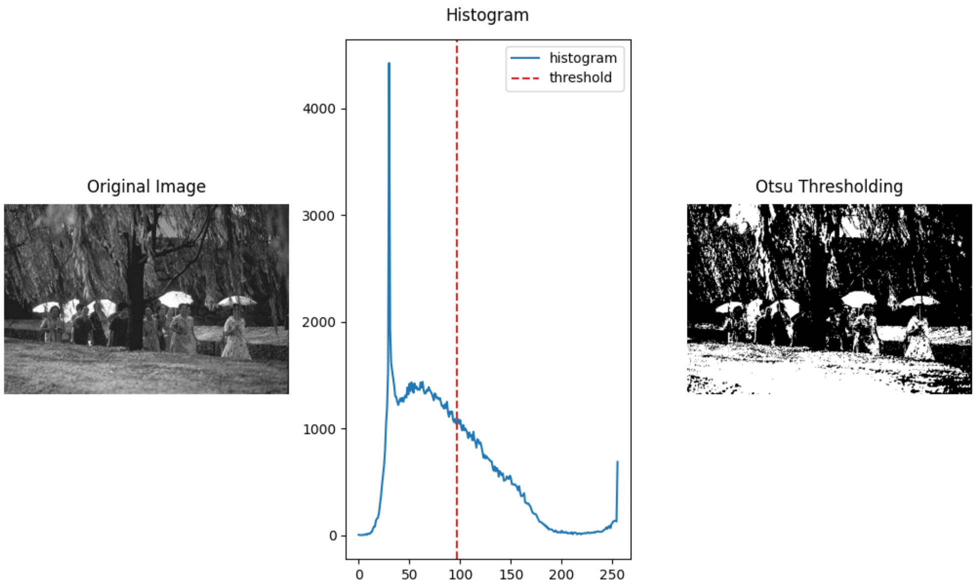


Image 3:


Original Image
Histogram
histogram
threshold
Otsu Thresholding