Project 4

Segmentation plays a pivotal role in image processing and computer vision tasks by partitioning an image into meaningful regions or objects. Among various segmentation techniques, histogram-based segmentation methods offer a simple yet effective approach to delineate objects from their background. One such technique involves identifying the valley between two peaks in the histogram of an image, which serves as a natural threshold for separating foreground and background regions. This approach leverages the distribution of pixel intensities to automatically determine an optimal threshold, making it widely applicable across different types of images and scenarios.

Theory and Algorithms

Histogram-based segmentation relies on the histogram of an image, which represents the distribution of pixel intensities. In the context of valley-based segmentation, the algorithm involves the following steps:

1. Compute Histogram: The first step is to generate a histogram of pixel intensities from the input image. This histogram illustrates the frequency of each intensity level present in the image.
2. Identify Peaks: Next, the algorithm identifies peaks in the histogram, which correspond to prominent intensity values present in the image. These peaks typically represent foreground and background intensities.
3. Find Valley: After identifying the peaks, the algorithm locates the valley between them, which corresponds to a region of lower intensity values separating the foreground and background peaks. This valley serves as the optimal threshold for segmentation.

4. Thresholding: Finally, the algorithm applies the threshold obtained from the valley to segment the image, assigning pixels with intensities below the threshold to the background and pixels above the threshold to the foreground.

1. Area Computation: The area of a component in an image represents the total number of pixels enclosed by its boundary. To compute the area:
    - Iterate through each pixel in the component.
    - Increment a counter for each pixel encountered.
    - The final count represents the area of the component.
2. Perimeter Computation: The perimeter of a component is the total length of its boundary. To compute the perimeter:
    - Utilize contour tracing algorithms, such as the Moore-Neighbor Tracing Algorithm (MNTA) or the Freeman Chain Code, to trace the boundary of the component.
    - Sum the lengths of all contour segments to obtain the total perimeter.
3. Centroid Computation: The centroid of a component represents its geometric center, often denoted as (x_bar, y_bar). It can be calculated using the first moments of area.
    - Compute the moments of the component area using the formula: m_pq = ∑(x^p * y^q) for all pixels (x, y) within the component.
    - Calculate the centroid coordinates: x_bar = m_10 / m_00 y_bar = m_01 / m_00
4. Compactness Measure: Compactness is a measure of how closely a shape resembles a circle. It is often calculated using the ratio of perimeter squared to the area, normalized by 4π. The formula for compactness is:
    - Compactness = (Perimeter^2) / (4 * π * Area)

- Higher compactness values indicate shapes that are closer to a perfect circle.

Algorithms:

- For area computation, iterate through each pixel within the component and count the number of pixels.
- For perimeter computation, utilize contour tracing algorithms to trace the boundary of the component and sum the lengths of all contour segments.
- For centroid computation, compute the moments of the component area and calculate the centroid coordinates using the moment values.
- For compactness measure, compute the area and perimeter of the component using the previously mentioned algorithms, then apply the formula to obtain the compactness value.

These algorithms provide essential metrics for characterizing the shape and size of identified components in an image, facilitating various image analysis tasks such as object recognition, classification, and segmentation.

Conclusion

The technique seems to be very useful. The results are highlighted below and by my own standards, they are very good. They “keys.pnm” ran forever, I guess this is because there are too many components in the image. Tiny little details. Overall, this is a very helpful technique. That can be used in many areas such as background and foreground segmentation, medical imaging analysis and many more.

The results are shown below.

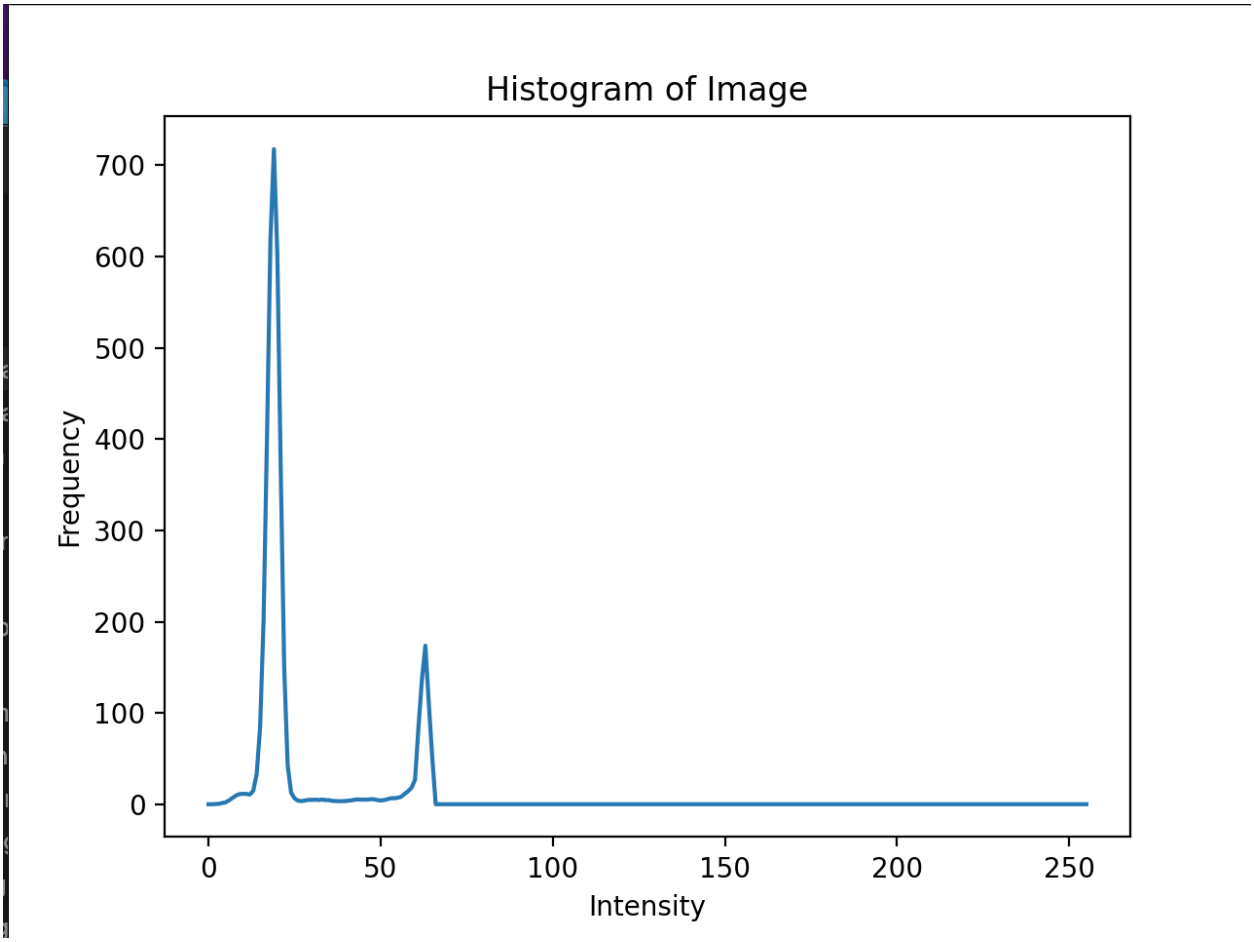
Hinge.pnm

Binary Image

Colored Components

Original Image

Threshold intensity: 43



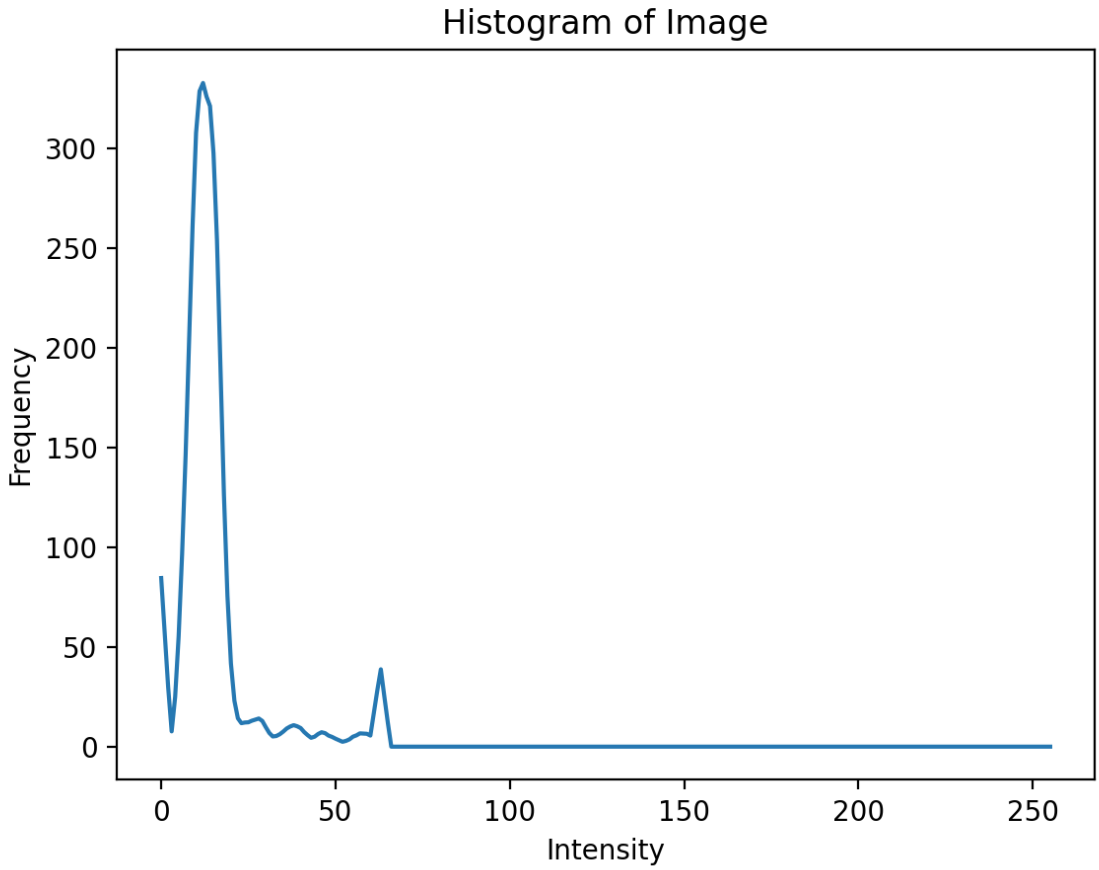
Component 1:
Area: 729
Perimeters: 291.8650048971176]
Centroids: 25, 26
Compactness Measures: 9.298794682875364

Hinges.pnm

Binary Image

Colored Components

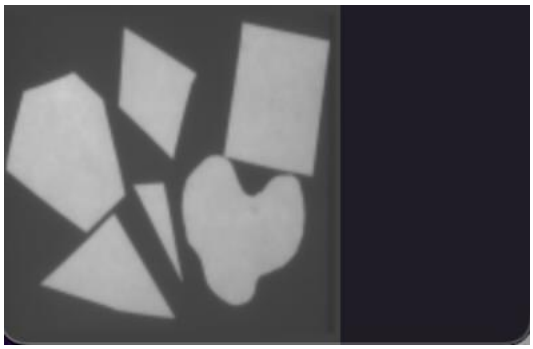
Orginal Image



Threshold intensity: 43

Component 1:
Area: 212
Perimeters: 5290470123]
Centroids: 30,28
Compactness Measures: 11.246712512400832

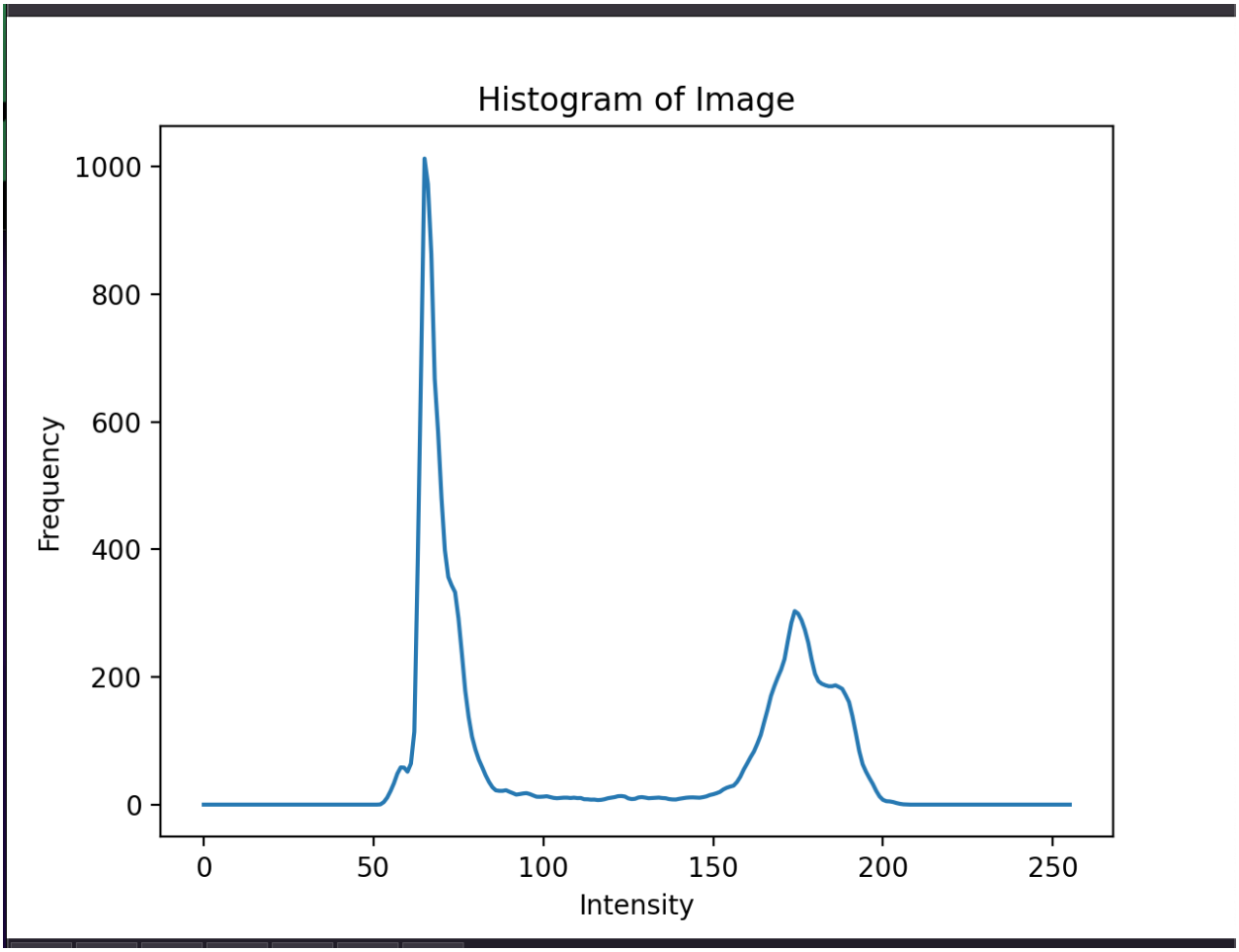
Shapes1.pnm



Orginal Image

Binary image

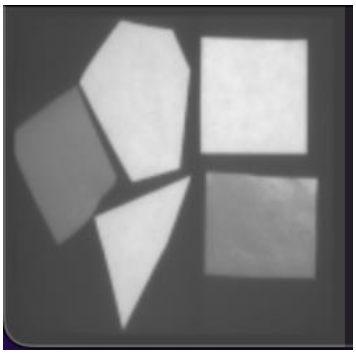
Colored Components



Thrshold Intensity: 123

Component 1:
Area: 7085
Perimeters: 861.3940045833588]
Centroids: 66,61
Compactness Measures: 8.334009108471811

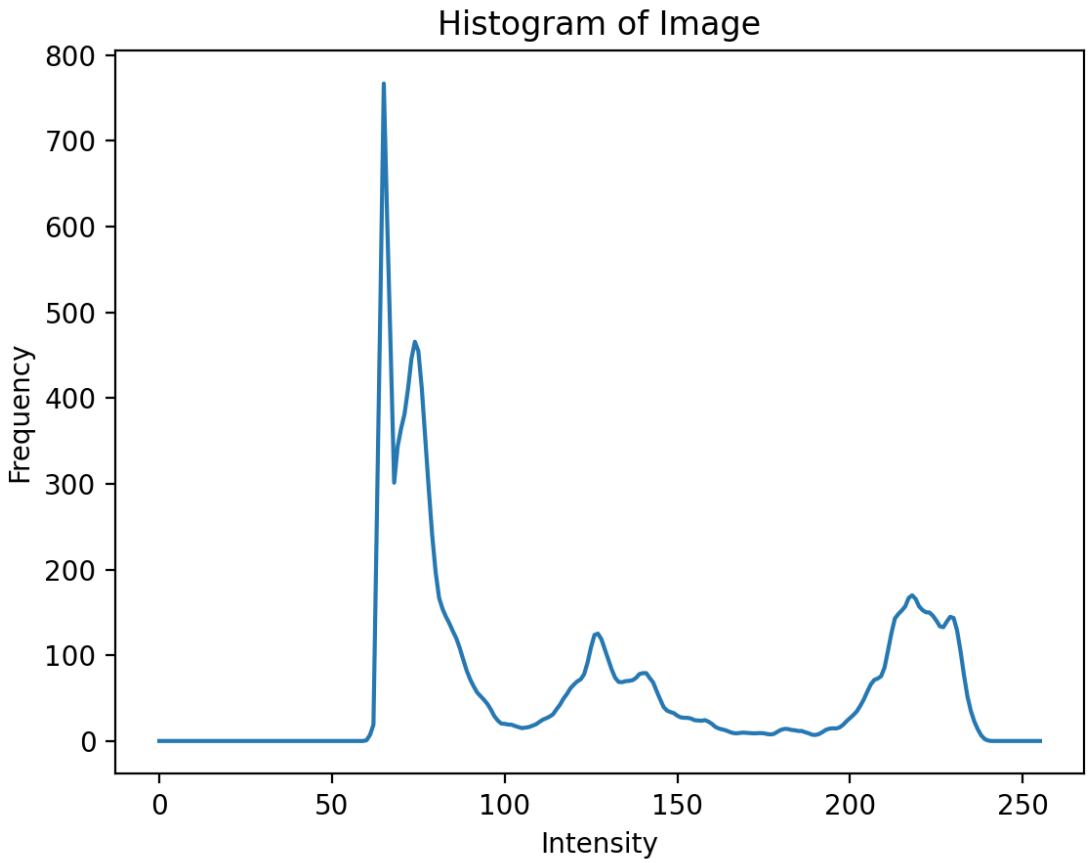
Shapes2.pnm



Original image

Binary image

Colored components



Threshold Intensity: 106

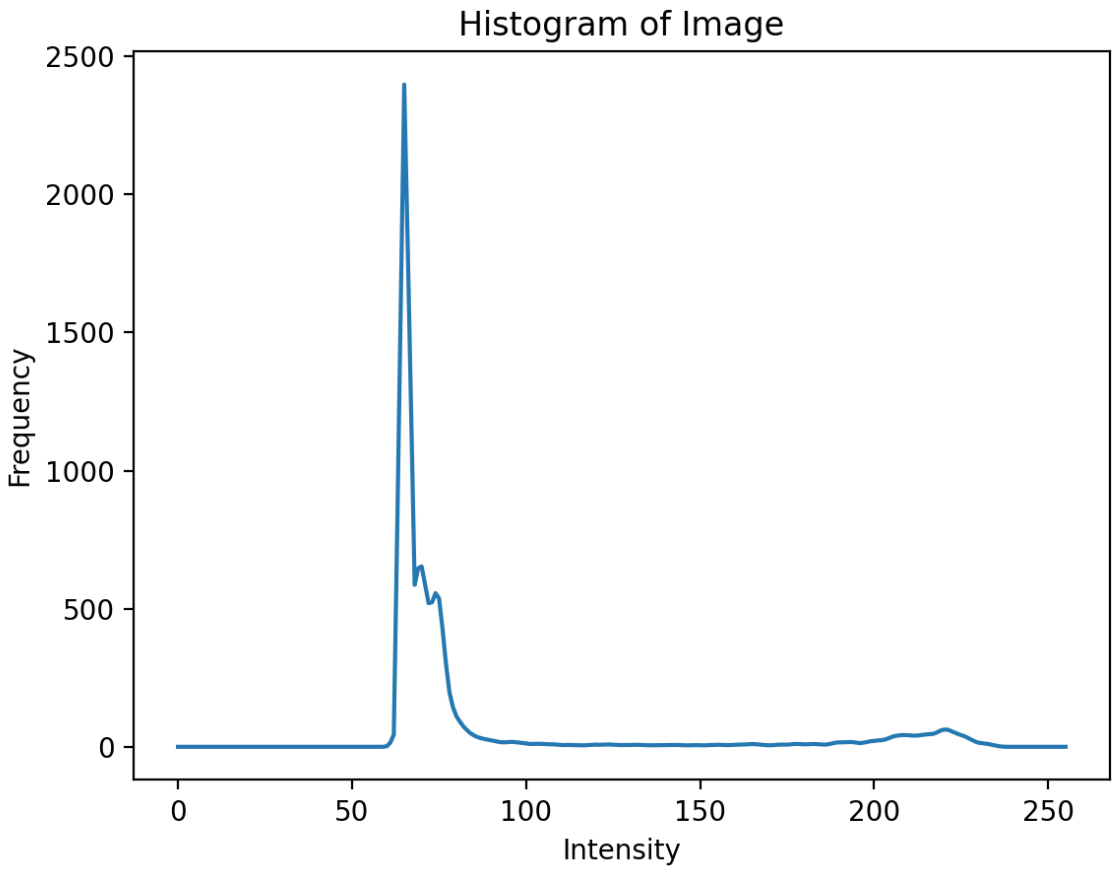
Component 1:
Area: 7413
Perimeters: 798.315795898437]
Centroids: 65,55
Compactness Measures: 6.841409414273735

Shapes3.pnm

Original Image

Binary image

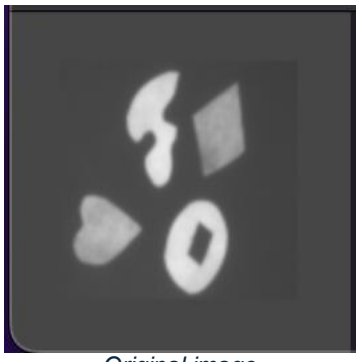
Colored components



Threshold Intensity: 110

Component 1:
Area: 2024
Perimeters: 576.0142800807953]
Centroids: 60,63
Compactness Measures: 13.045061421557762

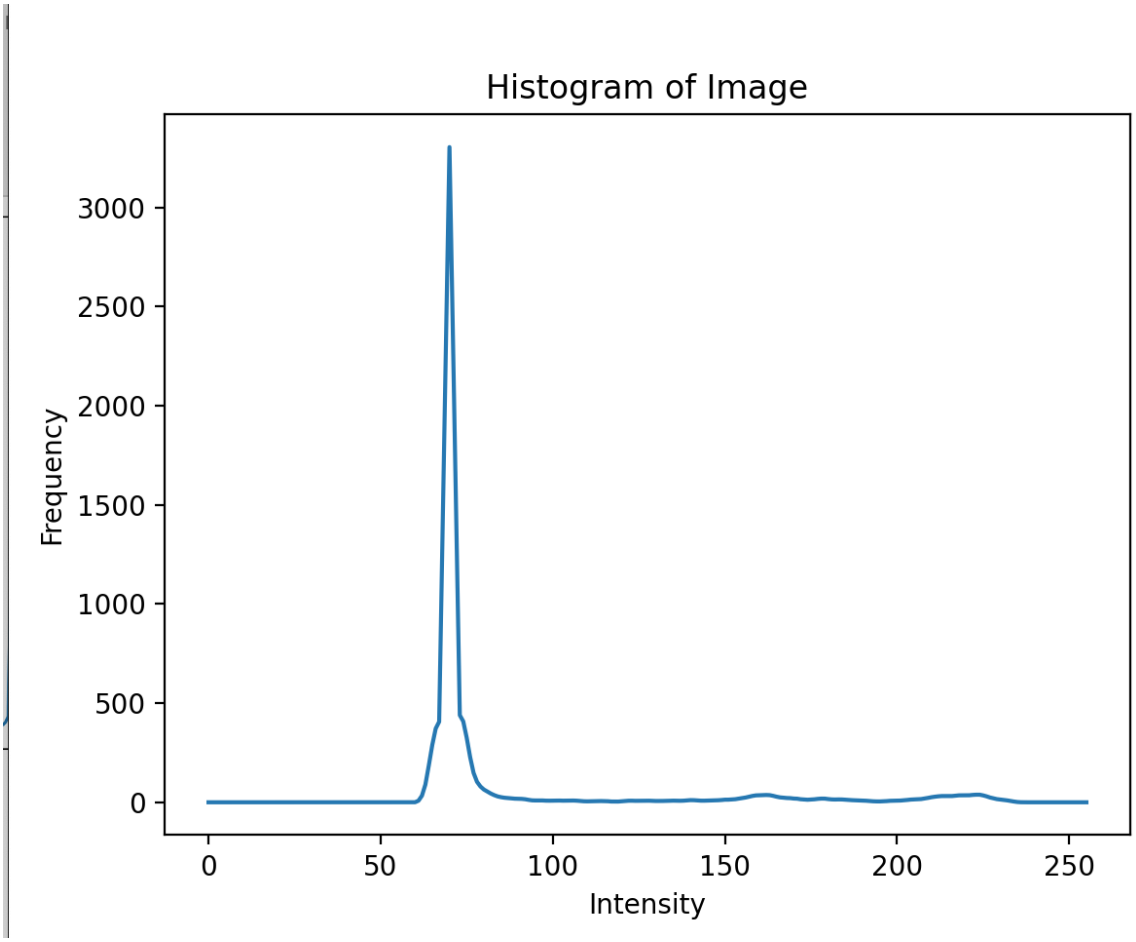
Shapes4.pnm



Original image

Binary image

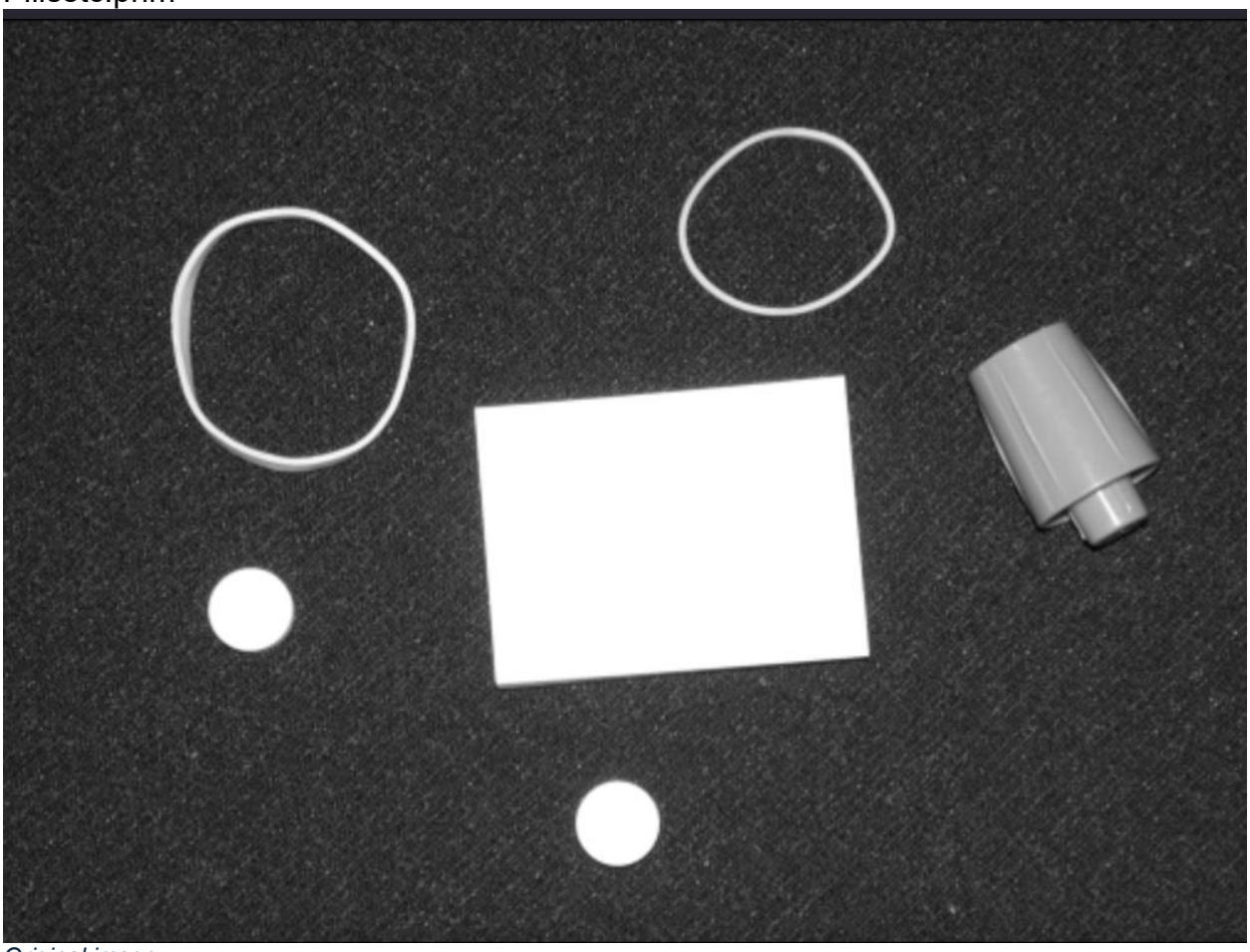
Colored components



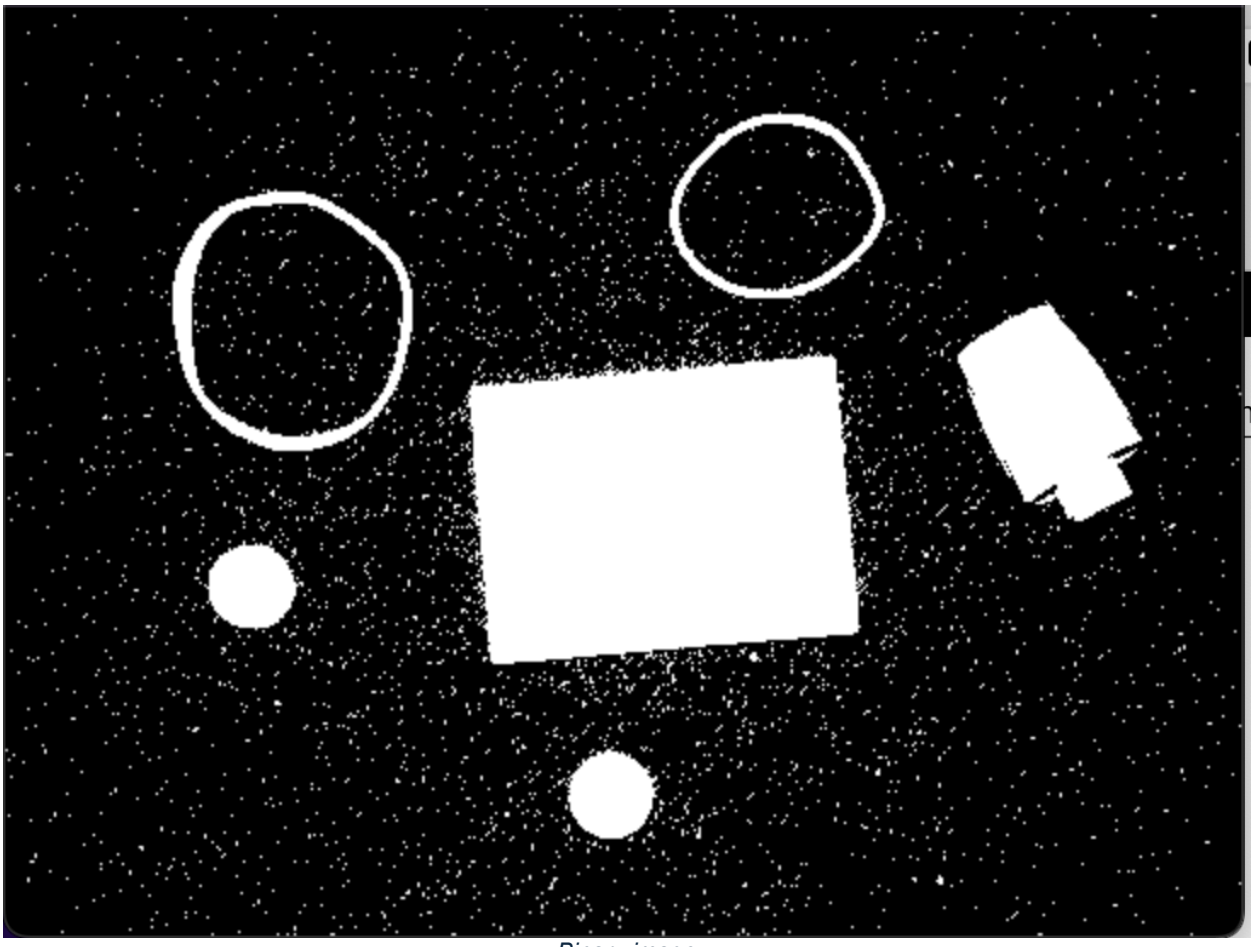
Threshold Intensity: 80

Component 1:
Area: 2489
Perimeters: 405.5046133995056]
Centroids: 59,65
Compactness Measures: 5.2572283161291296

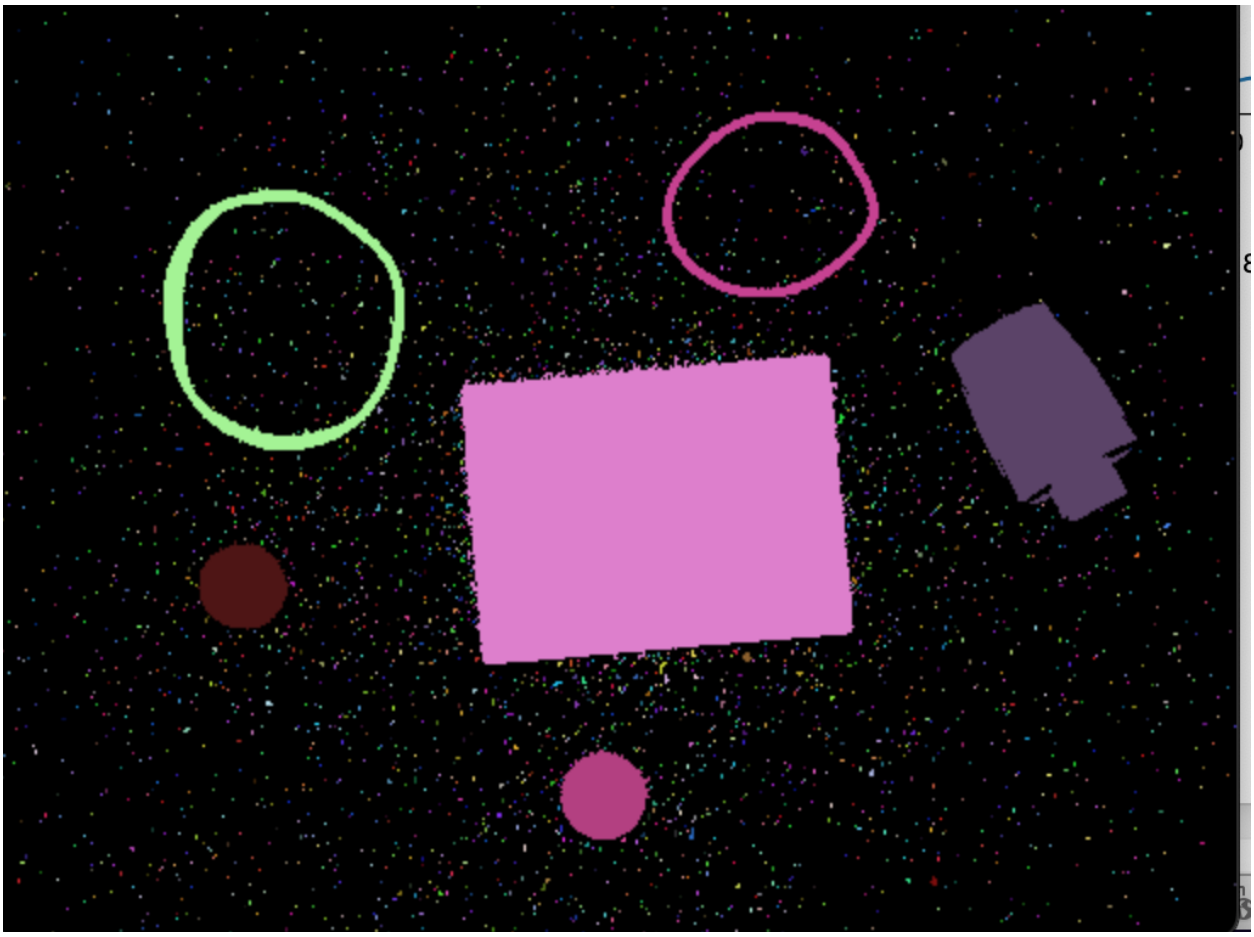
Pillsetc.pnm



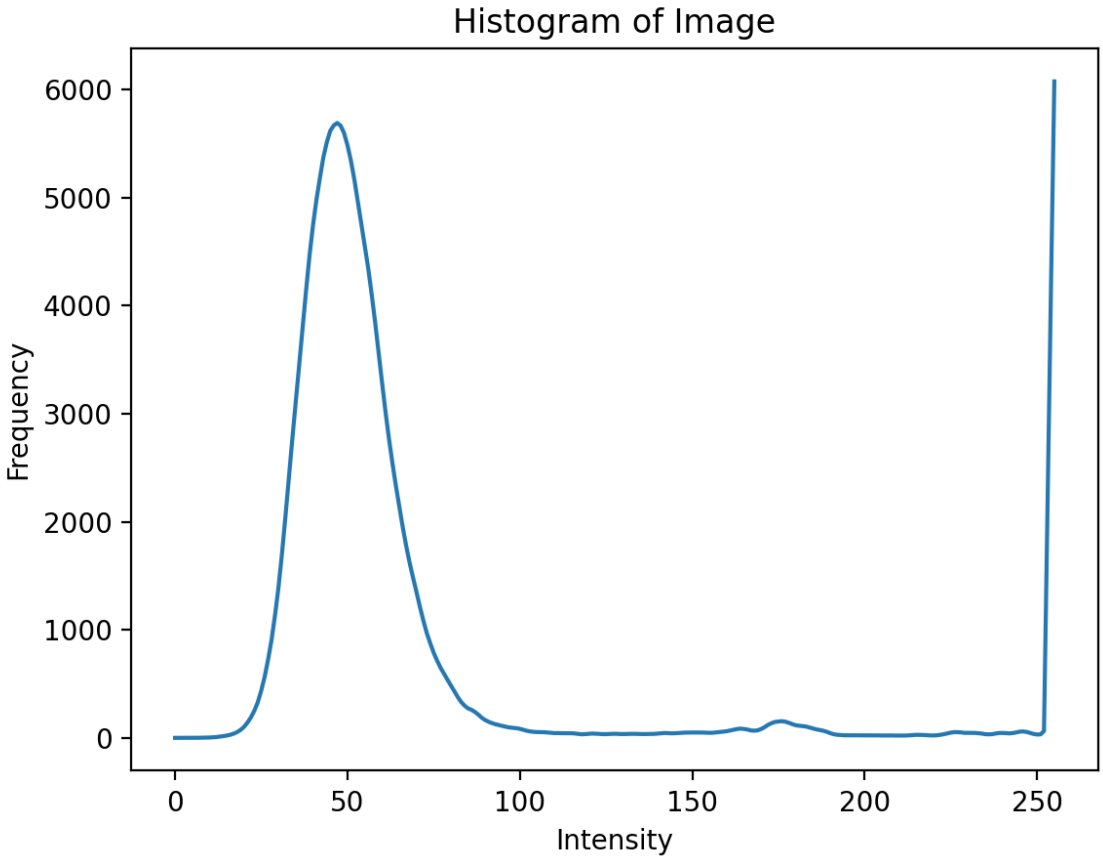
Original image



Binary image



Colored components

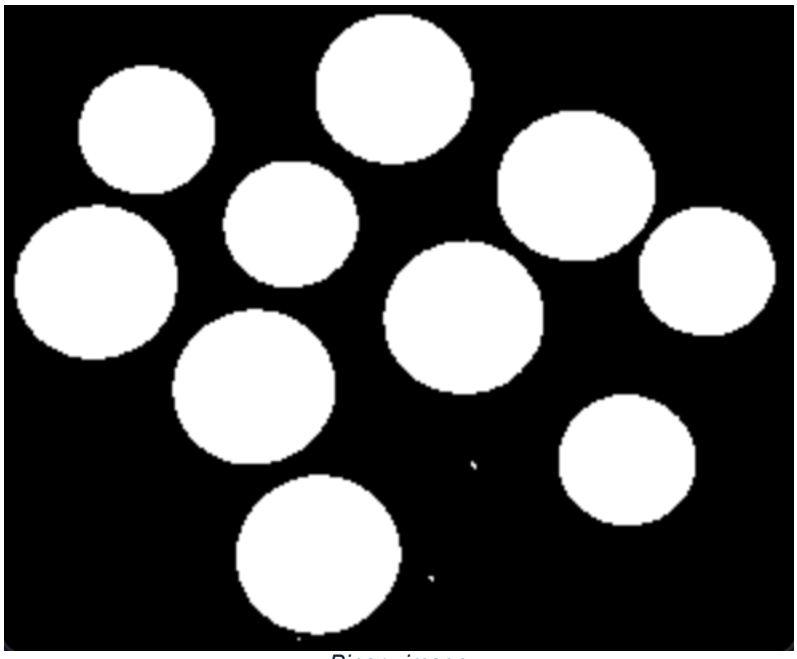


Threshold Intensity: 110

Component 1:
Area: 30341
Perimeters: 3849.330410003662]
Centroids: 277, 199
Compactness Measures: 38.86249031742677

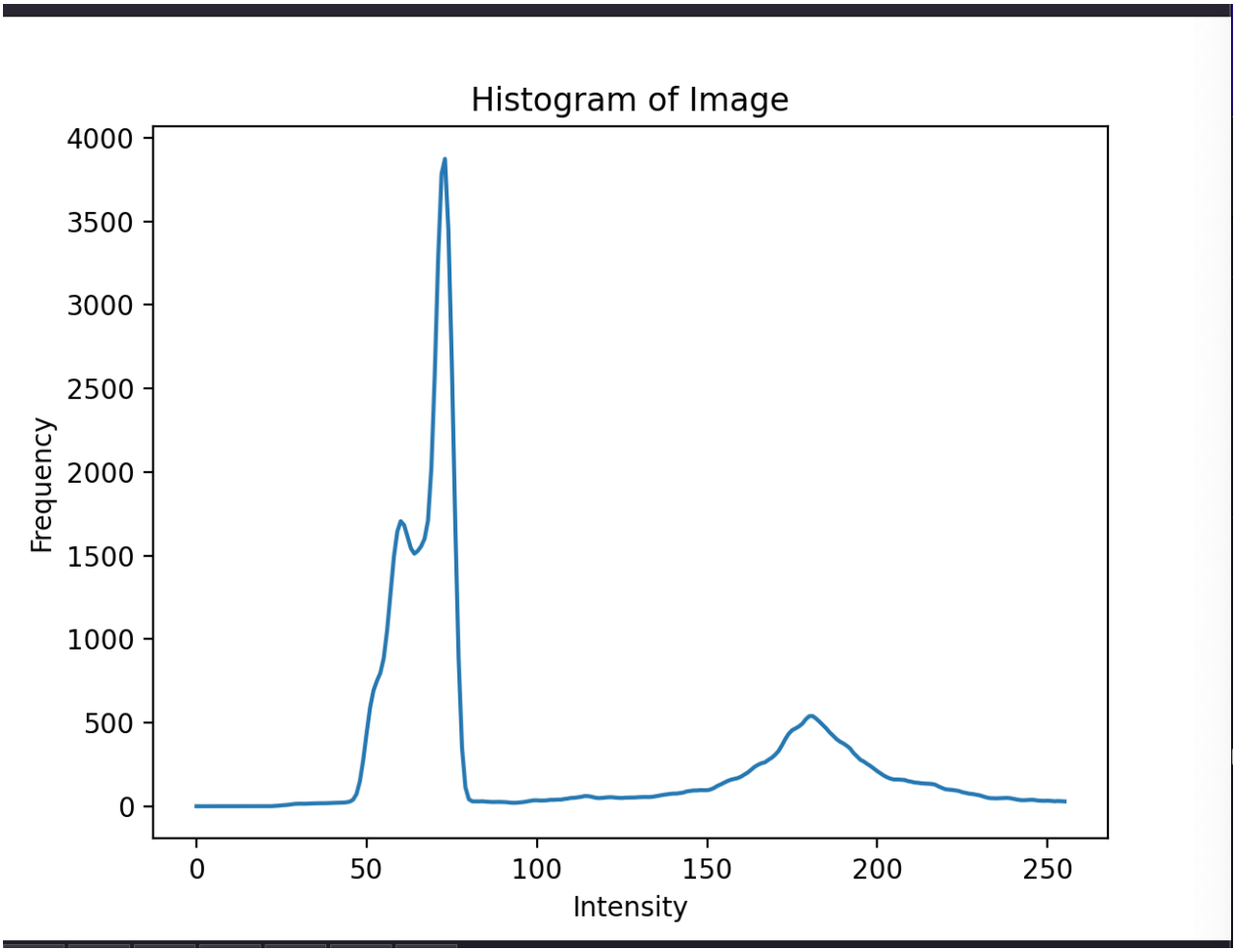
Coins.pnm

Original image



Binary image

Colored componenst



Threshold Intensity: 80

Component 1:
Area: 24101
Perimeters: 1810.1261014938354
Centroids 142, 110
Compactness Measures: 10.818641629425473