E9 241 Digital Image Processing
Assignment 1

August 24, 2025

Dwaipayan Haldar

---

1. **Histogram Computation**

Ans: The histogram is computed with the help of custom made histogram function. It is compared with the Numpy library function for comparison.

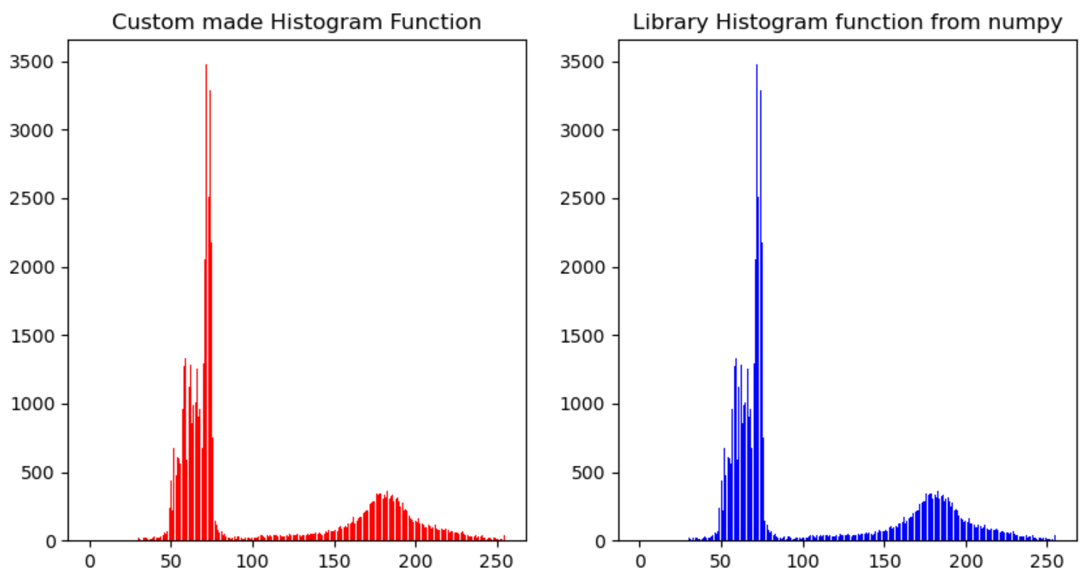

Figure 1: Histogram with custom made histogram function and comparison with library function

Observation: We can see two clear clusters in the histogram, which show that the background and the foreground are well separated. Now, darker portions have more frequency with less variance, so that implies that the background may be dark with low variance. The subject that usually occupies a lower frequency is on the brighter side. The subject has a higher frequency variance. The higher variance in the lighter regions suggests that the subject is not reflective or not that smooth. Also something like 100 (which comes out to be 125 in the second question) seems to be a good spot for segmentation from the histogram.

Average Intensity: The average intensity is calculated from the histogram through another custom function. It turns out to be **103.31**. It is cross-checked with Mean function from the Numpy Module. That confirms the results shown by the custom-made histogram and the intensity function.

2. **Otsu's Binarization**

Ans:

(a) The original image and the Binarized image is shown. Along with Opencv's Otsu Thresholding library function is used for comparison, which gives comparable results.

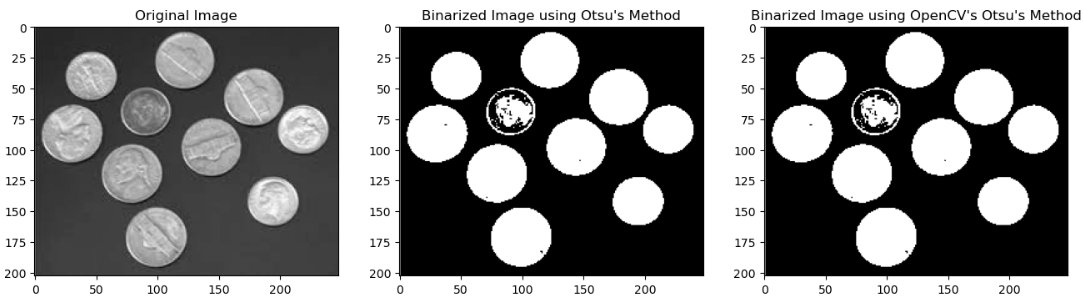

Figure 2: Binarization of Image Coins without offset minimizing within class variance

(b) The image is given an offset of 20 in every pixel and the output is clipped to 255. So there is a minute change in the histogram structure. The output after binarization comes comparable to the previous results.

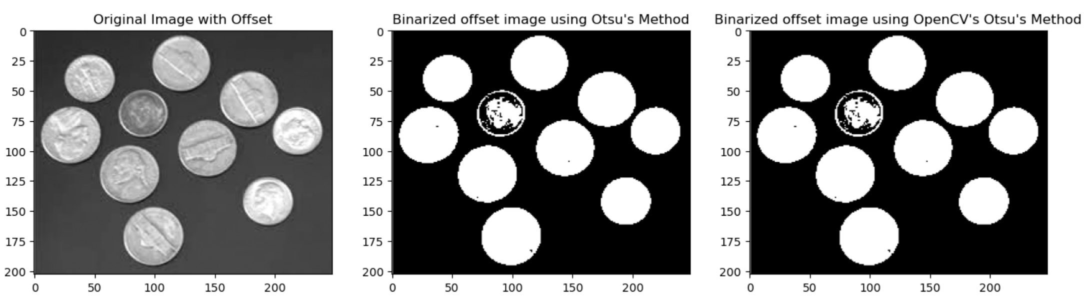

Figure 3: Binarization of Image Coins with offset maximizing between class variance

Observation: One important observation is that the threshold in the first case is 125 and in the second case is 145, which is also 20 offset, Apparently, it seems there is a relation and that can be proved mathematically.

$$
\begin{aligned}
w_0'(t+c) &= \sum_{i'=0}^{t+c-1} p'(i') = \sum_{i'=0}^{t+c-1} p(i'-c) \quad \text{(let } j = i' - c \text{)} \\
&= \sum_{j=-c}^{t-1} p(j) = \sum_{j=0}^{t-1} p(j) = w_0(t) \\
\implies w_1'(t+c) &= w_1(t)
\end{aligned}
$$

$$
\begin{aligned}
\mu_0'(t+c) &= \frac{1}{w_0'(t+c)} \sum_{i'=0}^{t+c-1} i' \cdot p'(i') \\
&= \frac{1}{w_0(t)} \sum_{j=0}^{t-1} (j+c) \cdot p(j) \\
&= \frac{1}{w_0(t)} \left(\sum_{j=0}^{t-1} j \cdot p(j) + c \sum_{j=0}^{t-1} p(j) \right) \\
&= \frac{1}{w_0(t)} \left(\mu_0(t) w_0(t) + c \cdot w_0(t) \right) = \mu_0(t) + c \\
\implies \mu_1'(t+c) &= \mu_1(t) + c
\end{aligned}
$$

$$
\begin{aligned}
\sigma_{b'}^2(t+c) &= w_0'(t+c) w_1'(t+c) [\mu_0'(t+c) - \mu_1'(t+c)]^2 \\
&= w_0(t) w_1(t) [(\mu_0(t) + c) - (\mu_1(t) + c)]^2 \\
&= w_0(t) w_1(t) [\mu_0(t) - \mu_1(t)]^2 \\
&= \sigma_b^2(t)
\end{aligned}
$$

$$
\begin{aligned}
\therefore \arg\max_{t'} \sigma_{b'}^2(t') &= \arg\max_{t} \sigma_b^2(t) + c \\
k_2 &= k_1 + c
\end{aligned}
$$

But there is an important assumption that there is no clipping and there is no change in the histogram pattern. In other words, it is just a shifted histogram. But that is not the case when it is clipped. So, this relation becomes invalid in my case (Since, I have clipped at 255) for very high offset when there is a significant change in the histogram.

3. **Adaptive Binarization**

Ans: The image is treated with different window and 20% overlap and the output is shown as follows.

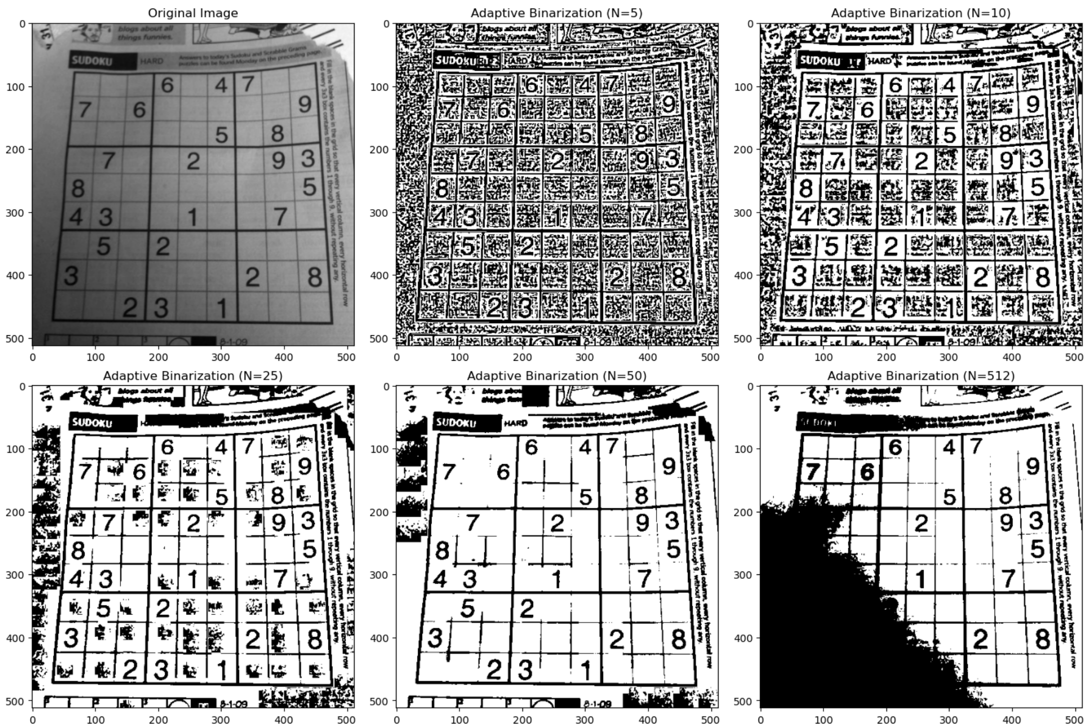

Figure 4: Adaptive Binarization with different windows

Visual Differences The ones with smaller windows appears noisy but they retain the local changes, like the lines are visible. In larger windows, noise is reduced but at the cost of some details. For overall threshold, the binarized image is of no use due to sharp light variation.

Advantages and Disadvantages With *smaller window*, local variation can be captured accurately. But the image comes out to be very noisy. It can be good for medical images where some small tumour or something is to be observed.

With *larger window*, local variation cannot be captured accurately. But there is no noise as such. For our case, adaptive binarization with larger window is better suited as we do not require to retain the small details. We would actually like to extract those characters. It is good for this type of images.

4. **Connected Components**

Ans: The connected components is extracted with the help of the algorithm shown in the class. The punctuations are not removed by filtering out the connected components with less than 100 pixels. All 64 characters are extracted from the image and the largest component N is marked in red.

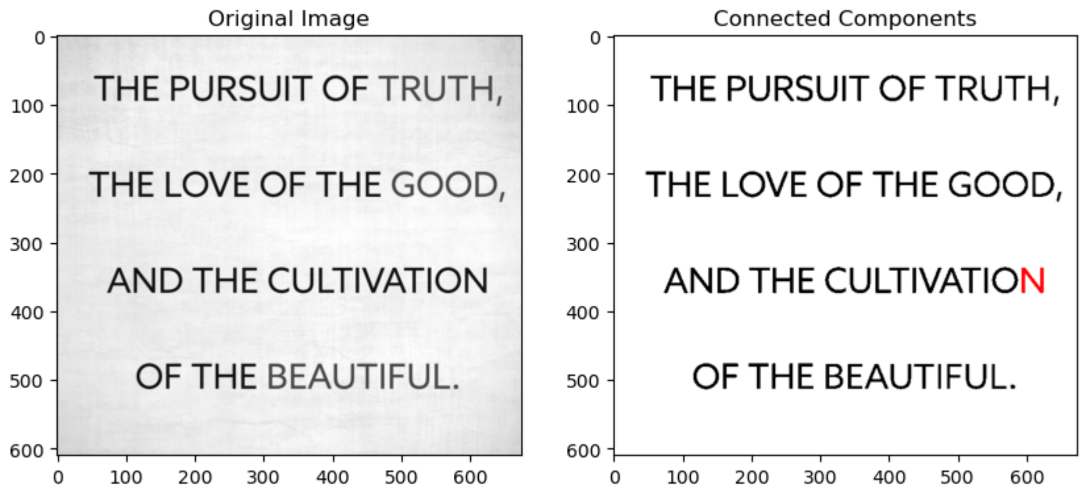

Figure 5: Largest Connected Components