E9 241 Digital Image Processing Assignment 01 Report

Name: Saptarshi Mandal

S.R. No.: 04-03-01-10-51-23-1-22925

Discipline: Signal Processing

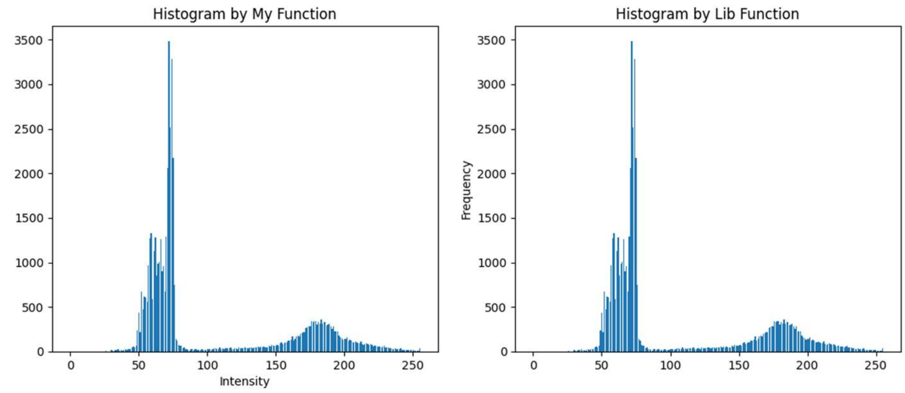
Department: Electrical Engineering

Q1. Histogram Computation

Observations/Results:

Given Image is **coins.png**

The histogram is computed by plotting frequencies with respect to intensity levels and a comparison is made between my function and the library function.



Comments/Inferences: The histogram computed by both my function and the library function is same. From the histogram we can infer that there is **a relatively bigger dark background on which there is/are smaller foreground elements.** (Conclusion is based on the fact that background contains more pixels than foreground)

The average intensity computed by both the functions (Mine and Lib) comes out to be **103.305.**

Q2. Otsu's Binarization

Observations/Results:

In this problem, it has been asked to Binarize the image **coins.png** by finding the optimal threshold t by two methods:

a) Minimizing the within class variance σ_ω^2
b) Maximizing the between class variance σ_b^2

The Optimal Threshold Value computed by both these methods comes out to be **125.**

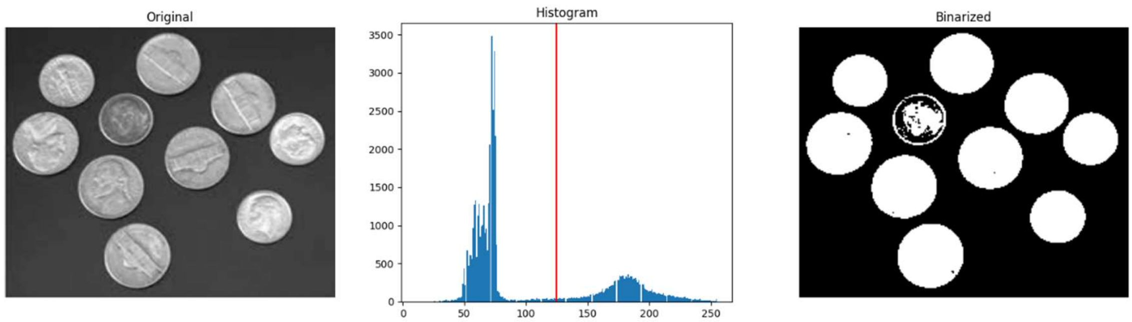
Hence, both methods are equivalent.

The time taken by the first approach (Within Class Variance) is **5.983s**

The time taken by the second approach (Between Class Variance) is **4.438s**

Comments/Inferences: The time taken by the second approach is less compared to the first approach. This is because the computation of within class variance σ_ω^2 requires computation of higher order statistics which is not required in the second approach.

Note: It must be noted that the time taken by both these approaches can be grossly reduced by simply including the histogram of image as an additional input for both the functions. Since, in the question it is already given that the functions will provide only the respective variance values for each value of threshold, each such value of variance is stored in an array and the minimum value/maximum value is chosen to find out the optimal threshold. If the function was to be defined in a way such that it directly gives the optimal threshold value for a given image, then it can be designed to be grossly faster than this approach.

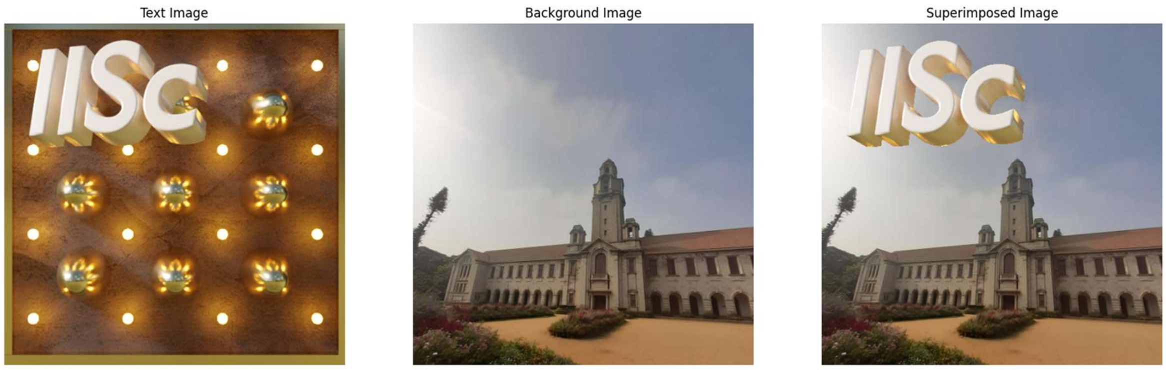


The above figure shows the Original Image, the histogram of the image with the Optimal Threshold Value marked in RED, and the Binarized Image using the optimal threshold.

Q3. Depth Based Extraction

Observations/Results:

In this problem, it has been asked to extract a text from one image and superimpose it on another image using the Inverse Depth Map of the Text Image.



The above figure shows the Text Image, Background Image, and the Expected Output (Superimposed Image).

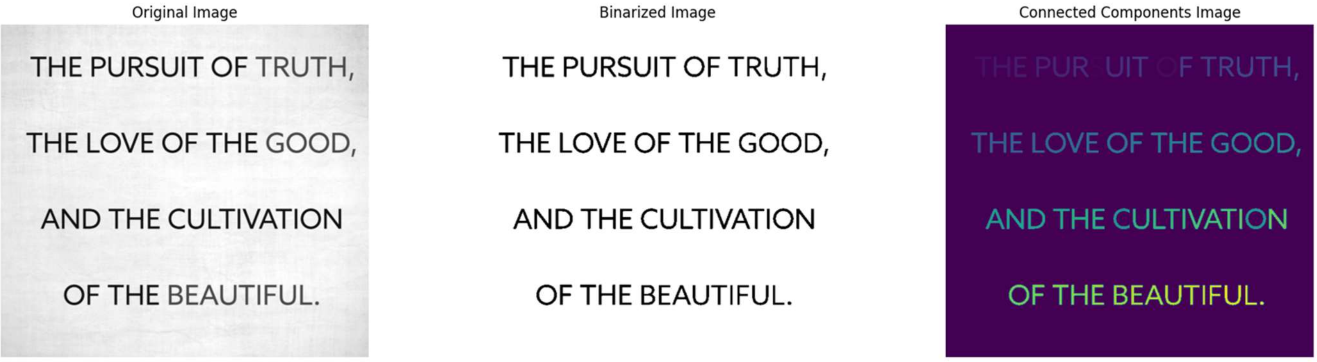
The time taken for computation is **0.55s**

Comments/Inferences: To solve the above problem, an inverse depth map of the text image was required. Binarization of the inverse depth map provided us with the pixels in the image where the Text was present. The next step was to simply replace the corresponding pixels of the Background Image with the pixel values of the Text Image.

Q.4 Connected Components

Observations/Results:

In this problem, it has been asked to find out the number of characters excluding punctuation in the following image.



The above figure shows the Original Image, the Binarized Image and the Connected Components Image (including punctuations)

The number of characters excluding punctuation in the given image is found to be **64.**

The time taken for computation is **0.941s.**

Comments/Inferences: To solve this problem, the first step is to binarize the image to get a binary image with just the characters. The next step is to find out the number of connected components in the image. Once the number of components in the image is found out, the components corresponding to punctuations are removed by a heuristic approach. The number of pixels of the punctuations will be less than the normal characters. The heuristic approach in this case is chosen to be a basic outlier detection method which computes z-score of a data point.

$$Z\_score = \frac{|X[i] - mean|}{Standard\ Deviation}$$

If the Z_score is above a certain threshold, then those data points are treated as outliers and are removed. **In this image the threshold was chosen to be 2** (Corresponds to 95.4% of datapoints in a Gaussian Distribution of data).

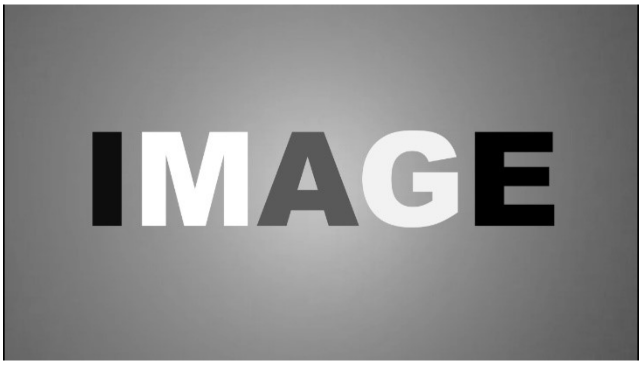
The components found in the image can be seen in the Connected Components image shown in the figure above. The presence of a single color in each letter signifies that the algorithm worked correctly.

Note: In the Code implementation, the binarized image is inverted because the Connected_Component algorithm is coded to search for connected components in White(intensity = 1).

Q5. Maximally Stable Extremal Regions(MSER)

Observations/Results:

In this problem it has been asked to find the number of maximally stable extremal regions in the following image:



Note: In the above image, it can be seen that the charachters/connected components are of **varying intensities ranging from a very low value (≈0) to a very high value (≈255)**. Hence, it is **not possible** to find a specific threshold around which binarization of the image can retain all concerned information about the components. Hence, Otsu's Binarization will fail to provide us the perfect binarized image and subsequently the number of connected components.

To find the number of characters, in this image an approach based on MSER has to be taken and the following is done:

(a) Sweep over all thresholds.

(b) For each threshold, determine connected components in the image.

(c) A connected component is termed an MSER if the size of the component does not change much (within δ) for a small perturbation ϵ in the choice of the threshold. (Both δ and ϵ are parameters that need to be chosen. Determine the stable threshold for each connected component.

(d) Ignore extremely large or extremely small connected components in the above analysis.

In my Algorithm the δ and ϵ are chosen as: $\delta = 10$ and $\epsilon = 50$. The δ signifies the number of pixels and ϵ signifies the number of threshold values for which the Region has been stable for (the size (region width, region height has) not changed beyond δ).

The number of characters in the image is found to be **5.**

The time taken for computation is **136.408s.**

Approach and Inference:

In the first run, the threshold = 0. The regions/connected components detected in that binarized image are stored in the form of vectors in a database, where each vector comprises of [row value of center of the component (i), column value of center of the component (j), no of pixels in the component].

Now the threshold is changed to 1 and again the regions are converted to vectors of the above form. Now, it is checked if the new vectors are already present (within a tolerance δ) in the database. If it is found to be present, then simply a counter increments the number of presence of the region by 1. If a new region is found which was not present in the database, then it is simply added to the database.

The above process is carried out for all thresholds till 255.

Now, the counter is checked for the regions with highest values and the regions with value lower than a certain threshold(ϵ) are rejected. Total regions that are left can be concluded to be a stable region.

Note: The counter provides us with the number of thresholds for which a certain region exists. The rejection is done based on the fact that if the counter value for a certain region is low that implies that the region does not exist for a significant number of thresholds and hence, is not Stable.