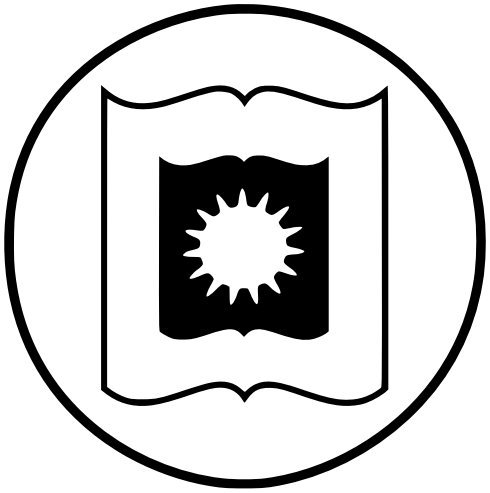

University of Rajshahi
Department of Computer Science & Engineering
CSE4182 - Digital Image Processing Lab

Histogram Equalization in a Gray Scale Image

Prepared by: Saharuzzaman Reyad
student id: 1810576105
Instructor: Sangeeta Biswas, Associate Professor
Date: August 3, 2022

1 Abstract

In this project histogram equalization will be performed with both OpenCV and Implementation.Python code will be used for implementation.
Index terms-histogram equalization,low contrast

2 Introduction

A histogram of an image is the graphical interpretation of the image's pixel intensity values. It can be interpreted as the data structure that stores the frequencies of all the pixel intensity levels in the image.

Histogram Equalization is an image processing technique that adjusts the contrast of an image by using its histogram. To enhance the image's contrast, it spreads out the most frequent pixel intensity values or stretches out the intensity range of the image. By accomplishing this, histogram equalization allows the image's areas with lower contrast to gain a higher contrast.

3 Procedure

Histogram Equalization can be used when you have images that look washed out because they do not have sufficient contrast.
Source image that has been used for histogram equalization is fig- 3.1

Figure 3.1: Source Image(Scenery)

I perform Histogram equalization on following gray scale image- 3.2

Figure 3.2: Gray Scale Image(Source Image)

Algorithm that is used for Histogram equalization is following :-

Algorithm

- Compute the histogram of pixel values of the input image. The histogram places the value of each pixel $f[x,y]$ into one of L uniformly-spaced buckets $h[i]$

$$h[i] = \sum_{x=1}^{N} \sum_{y=1}^{M} \begin{cases} 1, & if\ f[x,y] = i \\ 0, & Otherwise \end{cases}$$

Where L=2^8 and the image dimension is $M \times N$

- Calculate the cumulative distribution function

$$CDF[j] = \sum_{i=1}^{j} h[i]$$

- Scale the input image using the cumulative distribution function to produce the output image.

$$g[x,y] = \frac{CDF[f[x,y]] - CDF_{min}}{(N \times M) - CDF_{min}} \times (L-1)$$

Where CDFmin is the smallest non-zero value of the cumulative distribution function

Python code of Histogram equalization is given below:-
Github code link- click me

```
import cv2
import matplotlib.pyplot as plt
import numpy as np

src_img = cv2.imread('./scenary2.jpeg',0)
plt.imshow(src_img,cmap='gray')
plt.savefig('inputImg.png')
plt.show()

src_img_hist = cv2.calcHist([src_img],[0],None,[256],[0,256])
hist_equ_cv2 = cv2.equalizeHist(src_img)
L = pow(2,8)

def doHistEqu(img,histogram,L):
    CDF = histogram.cumsum()
    CDFmin = CDF.min()
    r,c = img.shape
    size = r * c
    newImg = np.zeros((r,c),np.uint8)
    for x in range(r):
        for y in range(c):
            newImg[x,y] = ((CDF[img[x,y]] - CDFmin) / (size - CDFmin)) * (L - 1)

    return newImg

equ_img = doHistEqu(src_img,src_img_hist,L)
img_set = [src_img,hist_equ_cv2,equ_img]
title_set = ['Source Image','Source Image Histogram','OpenCV Equalized Image',
```

```
                    'OpenCV Equalized Image Histogram', 'Implemented Equalized Image',
                        'Implemented Equalized Image Histogram']

def plot_img(img_set, title_set):
    n = len(img_set)
    r,c = 3,2
    plt.figure(figsize=(20,20))
    for i in range(n):
        plt.subplot(r,c,i*2+1)
        plt.imshow(img_set[i],cmap='gray')
        plt.title(title_set[i*2])
        plt.subplot(r,c,i*2+2)
        plt.hist(img_set[i].ravel(),256,[0,256])
        plt.title(title_set[i*2+1])
    plt.savefig('HistEquOpenCV.png')
    plt.show()

plot_img(img_set, title_set)
```

4 Result

Output of above 3 python code is fig- 4.1

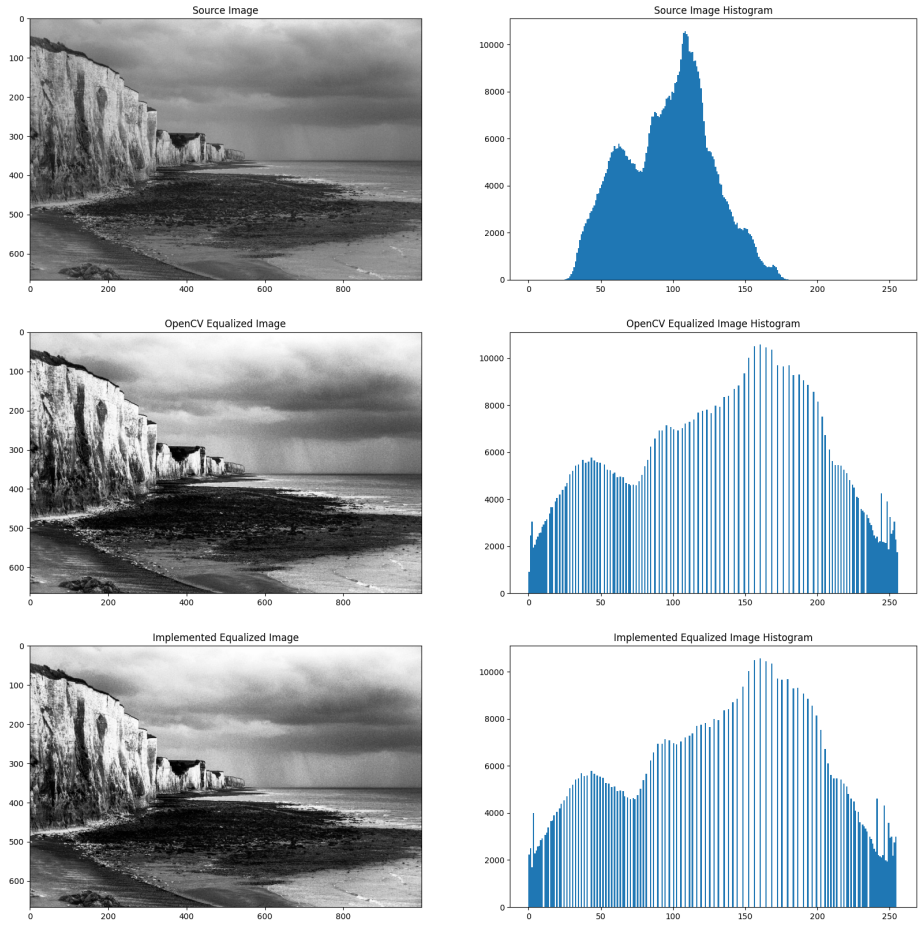

Figure 4.1: Output of Histogram Equalization

5 Discussion

From fig 4.1 we can see that OpenCV built-in cv2.equalizeHist() function and implemented Histogram equalizer function produce the same result. Pixel values in resulted images are scattered. Hence source image which was a low contrast image,after applying histogram equalization it becomes more visually pleasing and detailed.