PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043

Department of Electronics & Telecommunication Engineering

CLASS : B.E. E &TC **SUBJECT: DIVP**

EXPT. NO. : 3 **DATE :11-09-2020**

TITLE : PERFORM HISTOGRAM EQUALIZATION ON AN IMAGE

CO 1:	Apply the fundamentals of digital image processing to perform various operations on an image-enhancement in spatial domain/ frequency domain, image-restoration, image compression, video filtering and video compression on a given gray image. Examine the effect of varying the mask size and density of noise in an image and comment on the obtained results.
CO4:	Carry out experiments as an individual and in a team, comprehend and write a laboratory record and draw conclusions at a technical level.

AIM:

1. To plot the Histogram of an image(Dark, Light, Low contrast, High contrast)
2. To Perform the Histogram equalization on an image

SOFTWARES REQUIRED: Matlab 7.0. or above, python

THEORY:

3.1 Basics of Histogram.

The histogram of an image is a plot of the number of occurrences of gray levels in the image against gray level values. Histograms are the basis for numerous spatial domain processing techniques. It plays an important role in enhancement of perceived brightness and contrast of an image. It specifies the number of pixels having each gray level, but gives no

PICT-DIVP

hint as to where those pixels are located within the image. The histogram of an N × M image is defined as the percentage of pixels within the image at a given gray level:

$$h_i = \frac{n_i}{MN} \qquad for\ 0 \le i \le G_{\max}$$

where n_i is the number of pixels at gray level i,

NM is the total number of pixels within the image and

G_{max} is the maximum gray level value of the image.

3.1.1 Properties of Histogram.

1) The histogram is unique for any particular image, but the reverse is not true.
2) The sum of each normalized histogram value over the range of gray levels present within an image equals 1.

$$\sum_{i=0}^{G_{\max}} h_i = 1$$

3) The average gray level of an image in terms of histogram is:

$$avg = \sum_{i=0}^{G_{\max}} i \cdot h_i$$

And standard deviation of the gray levels of an image in terms of its histogram is given as:

$$std = \sqrt{\sum_{i=0}^{G_{\max}} i^2 \cdot h_i - avg^2}$$

4) The histogram of a dark image will be clustered towards the lower gray levels whereas; the histogram of bright images will be clustered towards higher gray levels.
5) For a low-contrast image, the histogram will not be spread equally that is the histogram will be narrow. For a high-contrast image, the histogram will have an equal spread in the gray level.

PICT-DIVP

PICT-DIVP

3.2 Histogram Modelling.

The histogram of an image represents the relative frequency of occurrence of the various gray levels in the image. Histogram modeling techniques modify an image so that the histogram has a desired shape. This is useful in stretching the low contrast levels of images with narrow histograms. Histogram modeling has been found to be a powerful technique for image enhancement.

3.2.1 Histogram Equalization.

The histogram technique that is used to enhance the brightness and contrast of an image is histogram equalization. The goal of histogram equalization is to distribute the gray levels within an image so that every gray level is equally likely to occur. In other words, histogram equalization takes an image's histogram and produces a new image with a histogram that is uniformly distributed. Histogram equalization will increase the brightness and contrast of a dark and low contrast image, making features observable that were not visible in the original image. Since histogram equalization distributes an image's gray levels uniformly about the range of gray levels, all images will have approximately the same brightness and contrast, hence allowing images to be compared equally without a bias due to perceived contrast and brightness differences.

3.2.1 Histogram Matching.

The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification. Histogram equalization automatically determines a transfer function that seeks to produce an output image that has a uniform histogram. Sometimes it is useful to specify the shape of the histogram that we wish the processed image to have. The process of histogram matching takes the concept of histogram equalization one step further, allowing for the specification of the output gray level histogram. Given a desired histogram d_i, the first step is to produce a mapping function $D(d_k)$ that histogram equalizes the desired histogram:

$$D(d_k) = \sum_{i=0}^{k} d_i \qquad for\, 0 \le k \le G_{\max}$$

The next step is to histogram equalize the input image using above equation.

$$b_k = M(a_k) = G_{\max} \sum_{i=0}^{k} h_i \qquad for\, 0 \le k \le G_{\max}$$

Since, the desired histogram and the input image have been equalized, producing approximately uniformly distributed histograms, the output graylevels of the histogram equalized input image can be equated to the mapping function that was used to get the desired histogram:

$$b_k = D(d_k)$$

The new graylevel mapping function that maps the equalized image to the desired histogram is the inverse of the above equation:

$$d_k = D^{-1}(b_k)$$

3.3 Histogram equalization derivation:

Step 1:

s = T(r) where $0 \le r \le 1$

T(r) satisfies,

(a) T(r) is single-valued and monotonically increasing function in the interval $0 \le r \le 1$

(b) $0 \le T(r) \le 1$ for $0 \le r \le 1$

(C) The inverse transformation from s back to r is

$$\boldsymbol{r = T^{-1}(s) \qquad 0 \le s \le 1}$$

PICT-DIVP

PICT-DIVP

Step 2:

Let ,

- ***$p_r(r)$*** denote the PDF of random variable r
- ***$p_s(s)$*** denote the PDF of random variable s

If ***$p_r(r)$*** and ***$T(r)$*** are known and ***$T^{-1}(s)$*** satisfies inverse condition then ***$p_s(s)$*** can be obtained using a formula :

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

The PDF of the transformed variable s is determined by the gray-level PDF of the input image and by the chosen transformation function

Step 3:

A transformation function is a cumulative distribution function (CDF) of random variable r:

$$s = T(r) = \int_0^r p_r(w)dw$$

Where w is a dummy variable of integration.

Step 4:

$$\begin{aligned}\frac{ds}{dr} &= \frac{dT(r)}{dr}\\ &= \frac{d}{dr}\left[\int_0^r p_r(w)dw\right]\\ &= p_r(r)\end{aligned}$$

$$\Longrightarrow$$

$$\begin{aligned}p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right|\\ &= p_r(r)\left|\frac{1}{p_r(r)}\right|\\ &= 1 \quad \text{where } 0 \le s \le 1\end{aligned}$$

PICT-DIVP

3.4 Algorithm:

1. Load required libraries like matplotlib, NumPy, import OpenCV
2. Read the grayscale image.
3. Plot histogram of input image.
4. Equalize the histogram using inbuilt function “cv2.equilizeHist()”.
5. To equalize the histogram without using inbuilt function, first store all histogram values i.e. no. of pixels per level into one array named ‘a’.
6. Find height and width of image.
7. For equalization, assign tmp = 1/(height*width).
8. Create new array ‘b’ such that b = a*tmp for all 256 levels.
9. Round the value of b and multiply it by 255.
10. Remap the equalized pixel values back into the image.
11. Plot the histogram after equalization.
12. Compare the result obtained in step 4 and 11, It will be same.

3.5 Conclusion:

- Histogram equalization is a process of distributing gray levels of an image uniformly over a range of 256 levels.
- Histogram Equalization process increases brightness and contrast of dark and low contrast images.
- It increases the contrast in two gray levels and hence helps in identifying the boundary or edge.

3.6 References:

i. Gonzalez R, Woods R, "Digital image processing", Pearson Prentice Hall, 2008.

ii. Gonzalez R, Woods R, Steven E, "Digital Image Processing Using MATLAB®", McGraw Hill Education, 2010.

iii. Jayaraman S, Esakkirajan S and Veerakumar T, "Digital Image Processing" Tata McGraw Hill, 2010

iv. Joshi, Madhuri A. "Digital Image Processing: an algorithm approach", PHI Learning Pvt. Ltd., 2006.

v. Pictures taken from: http://www.imageprocessingplace.com/root_files_V3/image_databases.html

PICT-DIVP

(Course Teacher)

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE - 411043

Department of Electronics & Telecommunication Engineering

CLASS	**:**	**B.E (E &TC)**	**COURSE**	**:**	**DIVP**
AY	**:**	**2020-21 (SEM- I)**	**DATE**	**:**	**11-09-2020**
EXPT. NO.	**:**	**3**	**CLASS & ROLL NO**	**:**	**BE VIII 42428**
TITLE	**:**	**PERFORM HISTOGRAM EQUALIZATION ON AN IMAGE**			

I. CODE:

Code For Manual Histogram

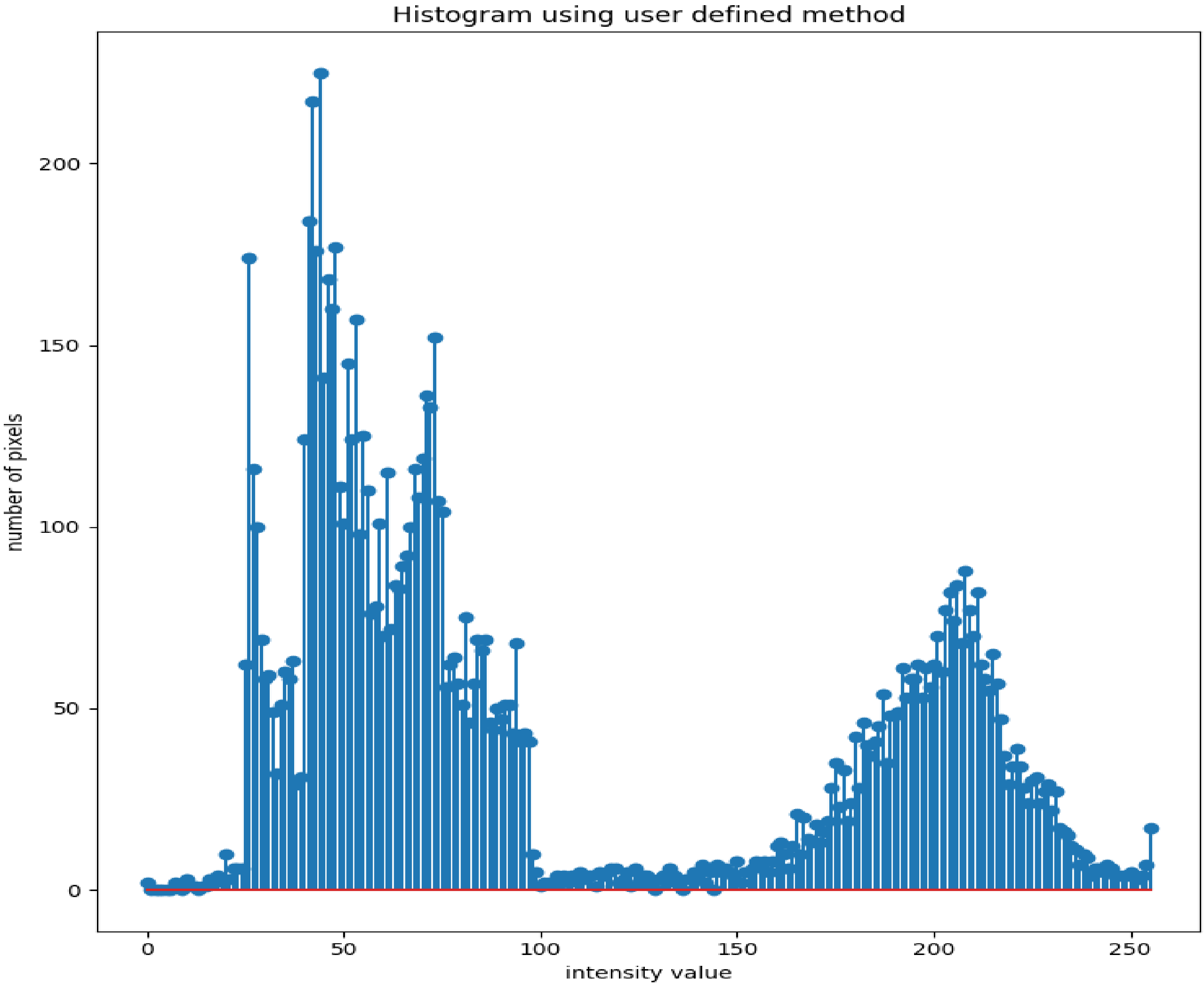
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('Images/dove1.jpg',0)
m, n = img.shape
# empty list to store the count
# of each intensity value
count =[]
# empty list to store intensity
# value
r = []
# loop to traverse each intensity
# value
for k in range(0, 256):
    r.append(k)
    count1 = 0
    # loops to traverse each pixel in
    # the image
    for i in range(m):
        for j in range(n):
            if img[i, j]== k:
                count1+= 1
    count.append(count1)

plt.subplot(1, 2, 1),plt.stem(r, count),plt.xlabel('intensity value'),plt.ylabel('number of pixels'),plt.title('Histogram using user defined method')
plt.subplot(1, 2, 2),plt.hist(img.ravel(),256,[0,256]),plt.title('Histogram Using Inbuilt Function')
plt.show()
```

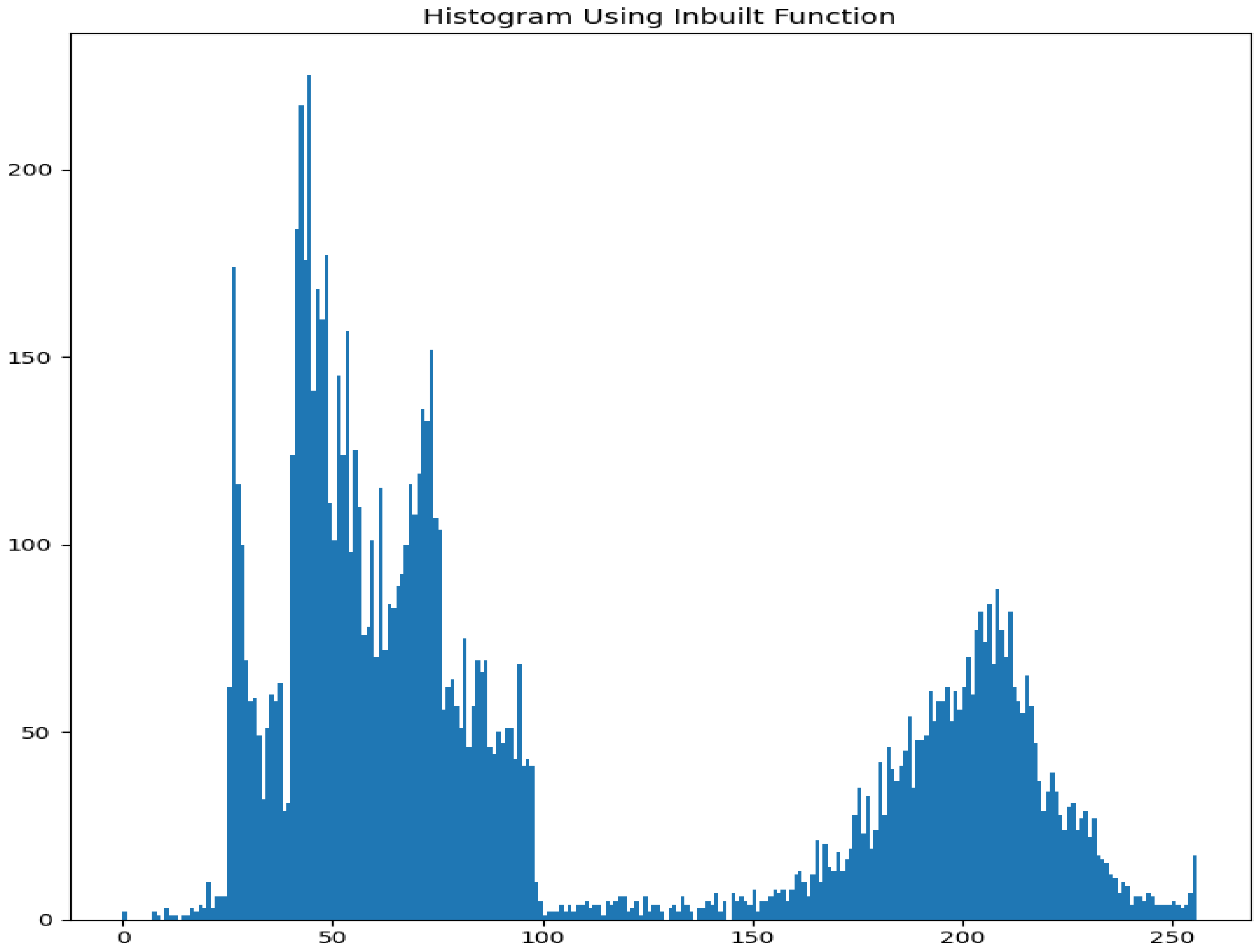
Histogram Equilization

```
import cv2

# import Numpy
import numpy as np
from matplotlib import pyplot as plt
# read a image using imread
img = cv2.imread('Images/babyincradle.png',0)

# creating a Histograms Equalization
# of a image using cv2.equalizeHist()
equ = cv2.equalizeHist(img)

a = np.zeros((256,),dtype=np.float16)
b = np.zeros((256,),dtype=np.float16)
height,width=img.shape

#finding histogram
for i in range(width):
    for j in range(height):
        g = img[j,i]
        a[g] = a[g]+1

print(a)

#performing histogram equalization
tmp = 255/(height*width)

#for i in range(256):
#    for j in range(i+1):
#        b[i] += a[j] * tmp;
#    b[i] = round(b[i] * 255);

b[0]=a[0]*tmp
for i in range(1,256):
    b[i] = b[i-1]+a[i]*tmp;

for i in range(0,256):
    b[i] = round(b[i]);

# b now contains the equalized histogram
#b=b.astype(np.uint8)
b = np.array(b, dtype = np.uint8)
print(b)

#Re-map values from equalized histogram into the image
#As at the end we are getting intensity value to be mapped for that pixel
#just replace that value in the original image with new value(b array)
img1 = np.zeros((height,width),dtype = 'uint8')
for i in range(width):
    for j in range(height):
        g = img[j,i]
        img1[j,i]= b[g]

plt.subplot(2, 3, 1),plt.imshow(img, 'gray'),plt.title('Original Image')
plt.subplot(2, 3, 2),plt.imshow(equ, 'gray'),plt.title('Histogram Equilized Image(inbuilt function)')
plt.subplot(2, 3, 3),plt.imshow(img1, 'gray'),plt.title('Histogram Equilized Image(user defined fun
ction)')
plt.subplot(2, 3, 4),plt.hist(img.ravel(),256,[0,256]),plt.title('Histogram Original Image')
plt.subplot(2, 3, 5),plt.hist(equ.ravel(),256,[0,256]),plt.title('Histogram of Image equilized using i
nbuilt function')
plt.subplot(2, 3, 6),plt.hist(img1.ravel(),256,[0,256]),plt.title('Histogram of Image equilized using
User defined function')
plt.show()
```

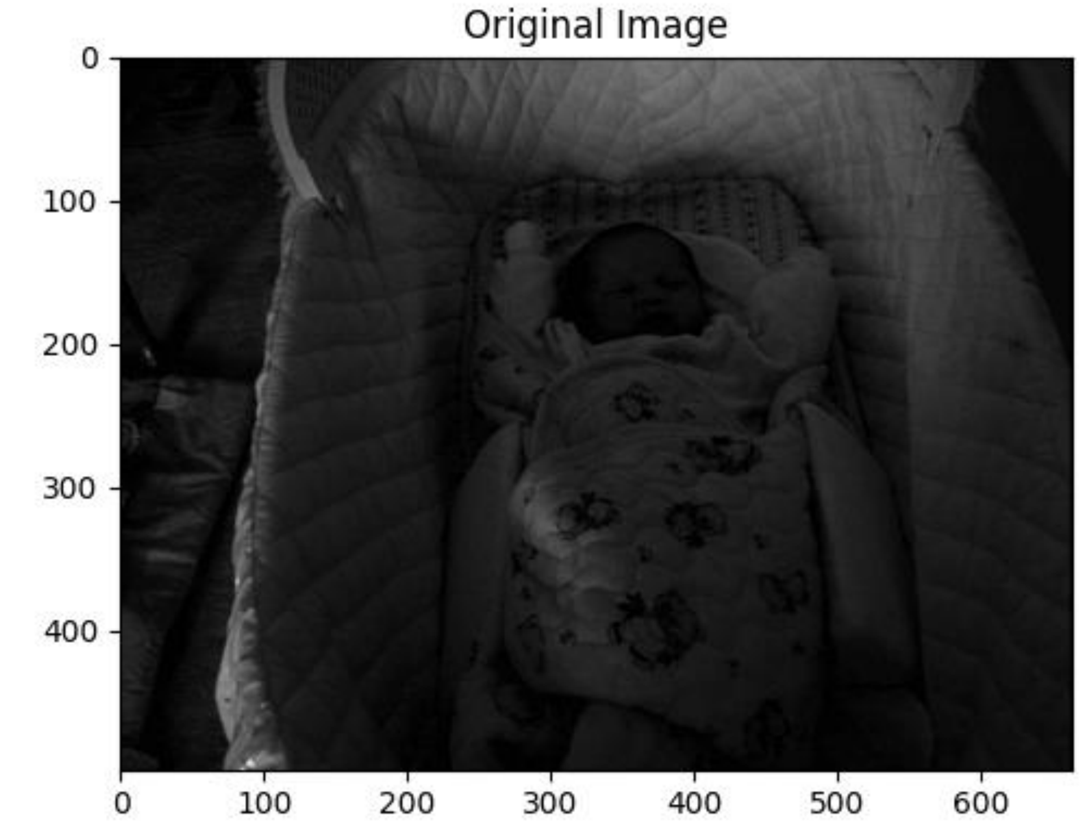
II. RESULTS:

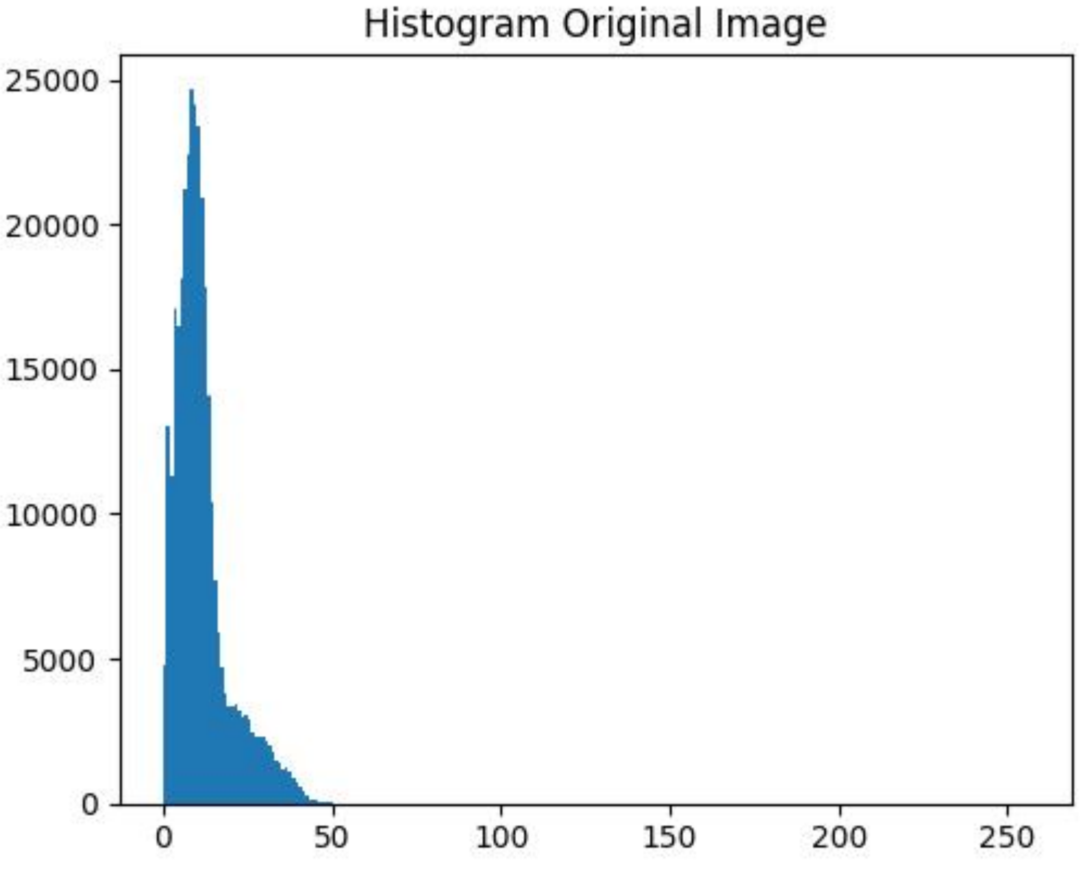
Histogram

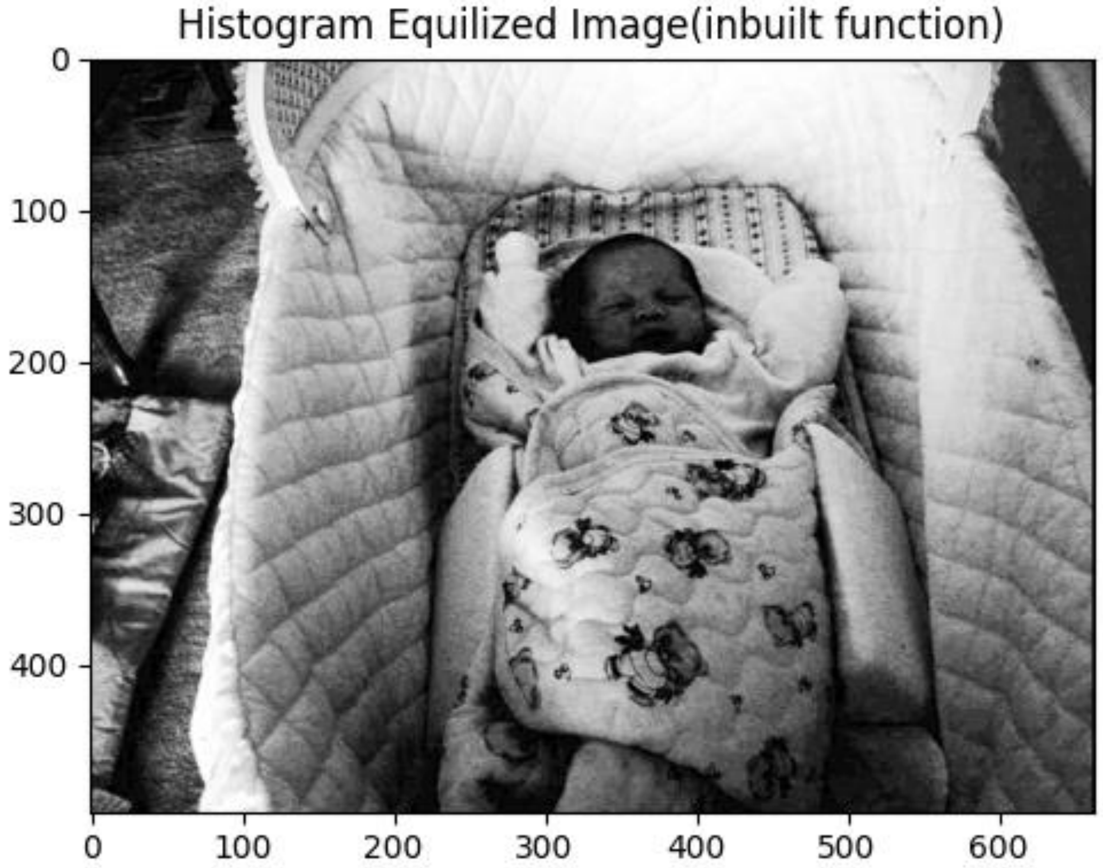


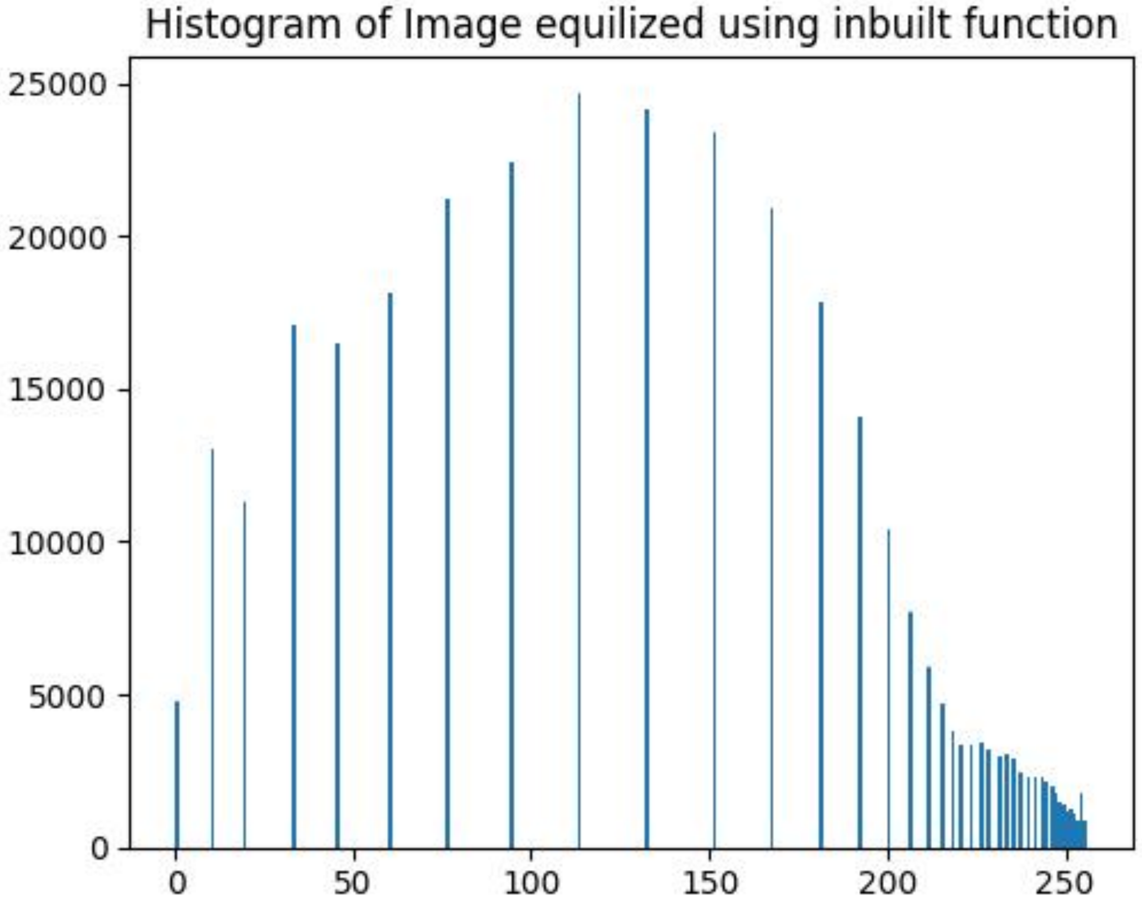

Histogram Using Inbuilt Function
0
50
100
150
200
0
50
100
150
200
250

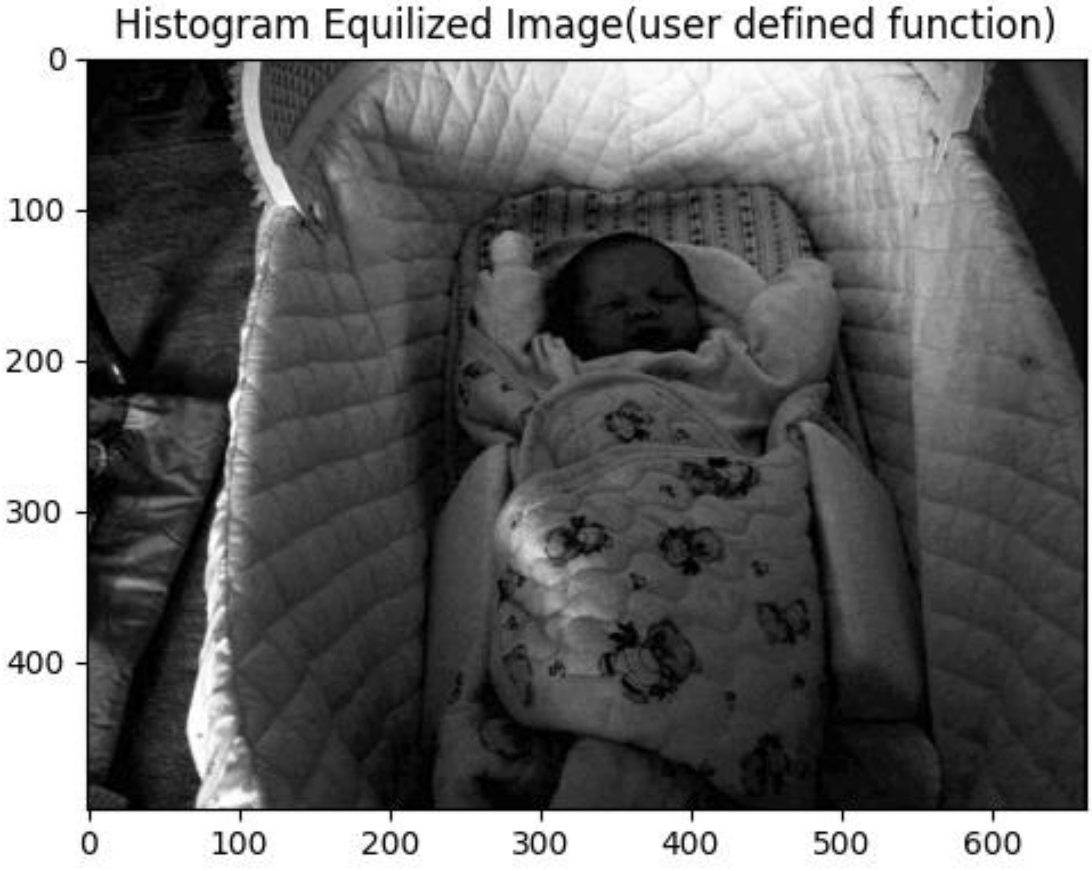
Histogram Equilization

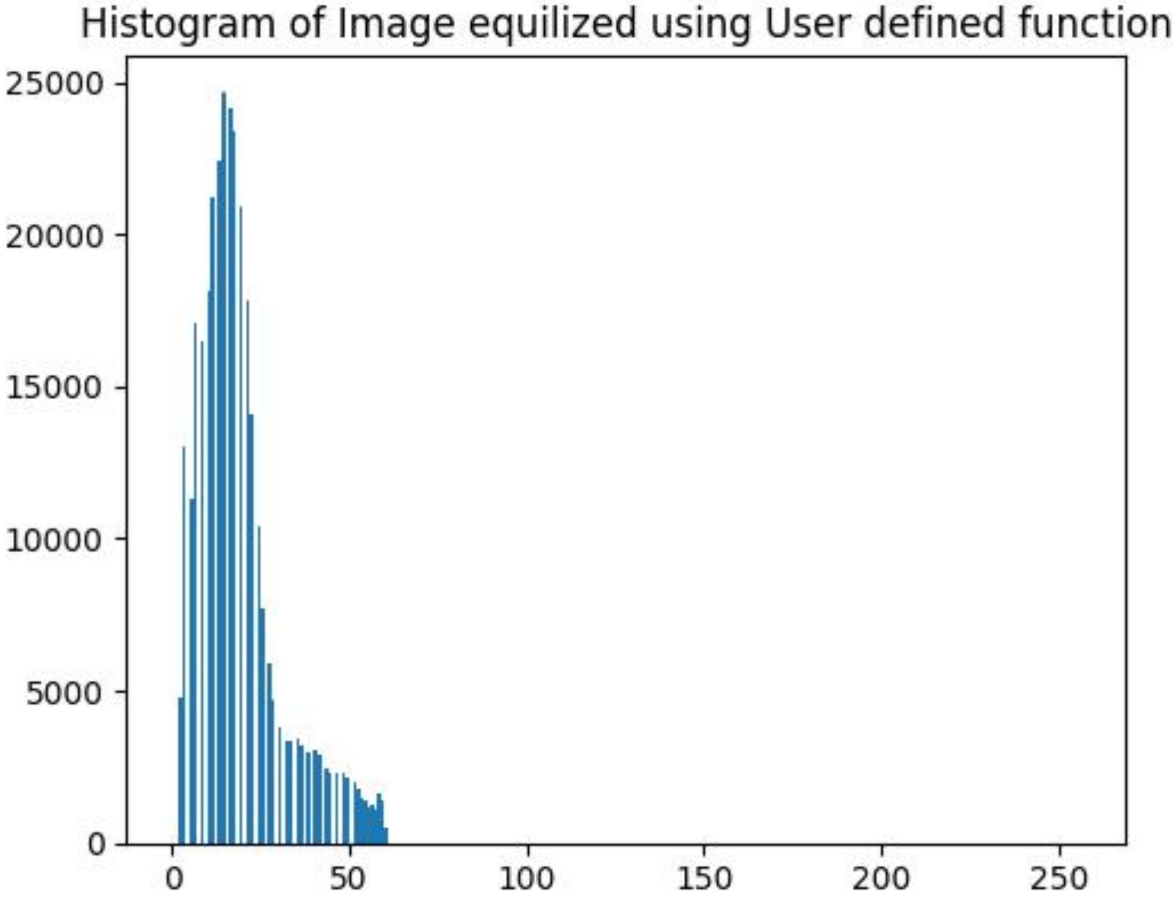





Histogram Equilized Image(inbuilt function)
0
100
200
300
400
0
100
200
300
400
500
600


Histogram of Image equilized using inbuilt function
25000
20000
15000
10000
5000
0
0
50
100
150
200
250


Histogram Equilized Image(user defined function)
0
100
200
300
400
0
100
200
300
400
500
600


Histogram of Image equilized using User defined function
25000
20000
15000
10000
5000
0
0
50
100
150
200
250