DIP_ASSIGNMENT 2

SUBMISSION DATE: 16-02-2022

NAME OF STUDENT: Vishal Dange

BRANCH: Information Technology

ID: 191080020

AIM: Write Programme for Histogram Display and Histogram Equalization.

THEORY:

Why Histogram Equalization Needed ?

Image processing is one of the rapidly growing technologies of our time and it has become an integral part of the engineering and computer science disciplines. Among its many subsets, techniques such as median filter, contrast stretching, histogram equalization, negative image transformation, and power-law transformation are considered to be the most prominent.

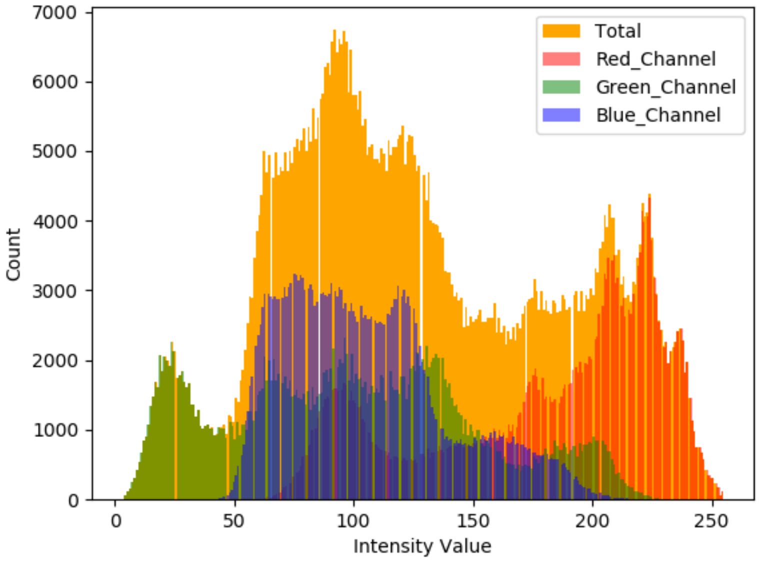
Definition

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 used to enhance contrast. It is not necessary that contrast will always be increase in this. There may be some cases were histogram equalization can be worse. In that cases the contrast is decreased.

Example :

As we can see in the image below the X-axis represents the pixel intensity levels of the image. The intensity level usually ranges from 0 to 255. For a gray-scale image, there is only one histogram, whereas an RGB colored image will have three 2-D histograms — one for each color. The Y-axis of the histogram indicates the frequency or the number of pixels that have specific intensity value



Why do you use Histogram Equalization?

Histogram Equalization can be used when you have images that look washed out because they do not have sufficient contrast. In such photographs, the light and dark areas blend together creating a flatter image that lacks highlights and shadows.

Code :

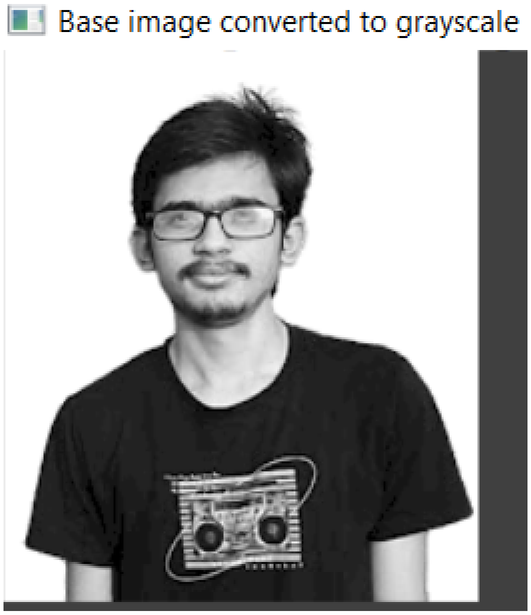
```
 """
Name : Vishal Dange
ID : 191080020
Branch : TY IT

"""
```

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


# Converting Base Image to Gray Scale After Reading it


image_path = "dip.png"
grayscale_image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
cv2.imshow("Base image converted to grayscale", grayscale_image)
cv2.waitKey()


# Getting histogram of grayscale base image


hist, _ = np.histogram(grayscale_image.flatten(), 256, [0, 256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()
plt.plot(cdf_normalized, color="b")
plt.hist(grayscale_image.flatten(), 256, [0, 256], color="r")
plt.legend(("CDF", "Histogram"))
plt.xlim([0, 256])
plt.title("Histogram of Image before histogram equalization")
plt.show()


# Displaying the image processed by histogram equalization

enhanced_image = cv2.equalizeHist(grayscale_image)
cv2.imshow("Enhanced Image", enhanced_image)
cv2.waitKey()

# Calculating the equalized histogram
```

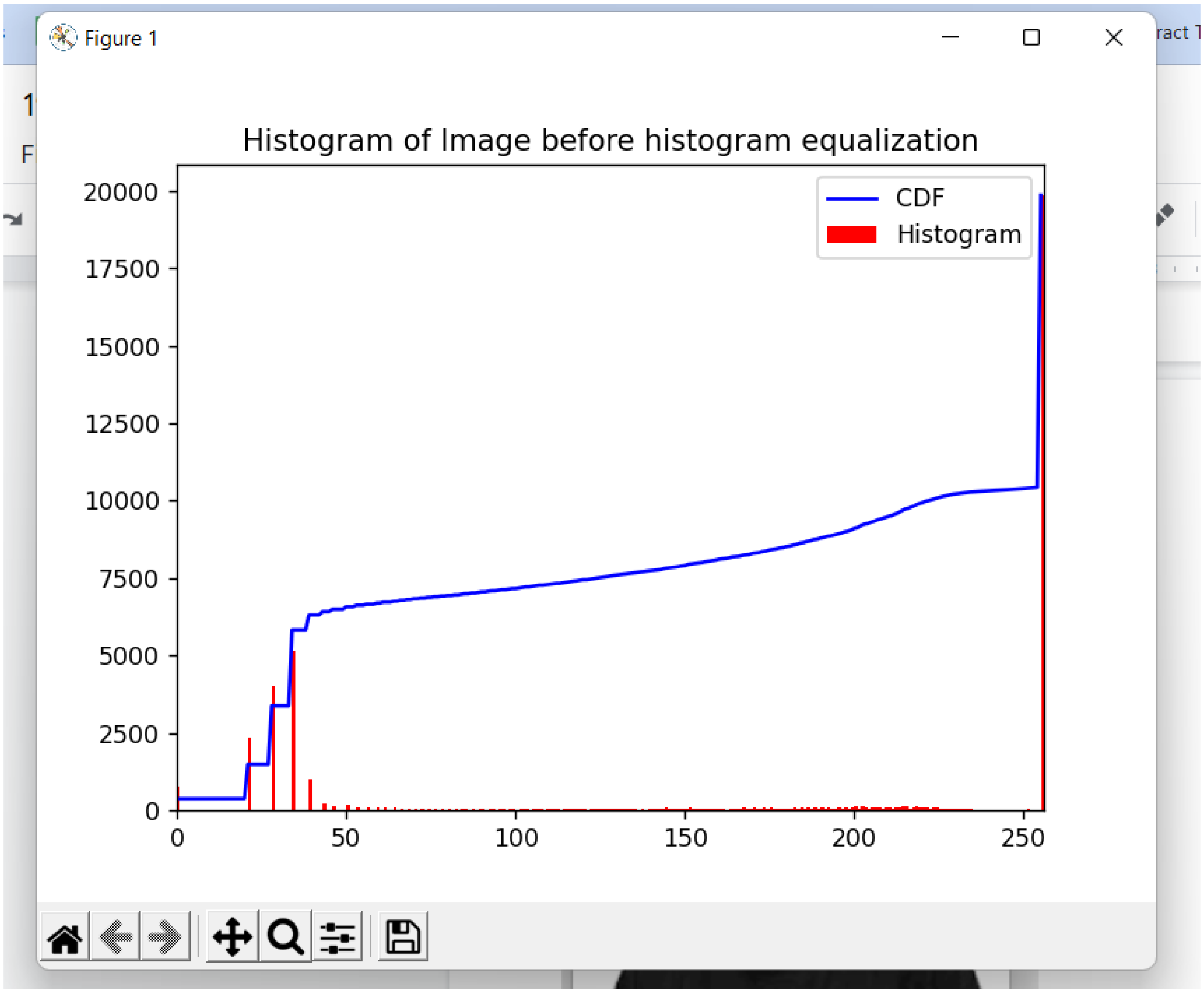
```
hist_enhanced, _ = np.histogram(enhanced_image.flatten(), 256, [0, 256])
cdf_enhanced = hist_enhanced.cumsum()
cdf_normalized = cdf_enhanced * float(hist_enhanced.max()) / cdf_enhanced.max()
plt.plot(cdf_normalized, color="b")
plt.hist(enhanced_image.flatten(), 256, [0, 256], color="r")
plt.legend(("CDF", "Histogram"))
plt.xlim([0, 256])
plt.title("Histogram of Image  after histogram equalization")
plt.show()

# Applying CLAHE to the grayscale image

clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
clahe_ehnanced_image = clahe.apply(grayscale_image)
cv2.imshow("CLAHE Enhanced Image", clahe_ehnanced_image)
cv2.waitKey()


# Calculating the equalized histogram

hist_clahe, _ = np.histogram(clahe_ehnanced_image.flatten(), 256, [0, 256])
cdf_clahe = hist_clahe.cumsum()
cdf_normalized = cdf_clahe * float(hist_clahe.max()) / cdf_clahe.max()
plt.plot(cdf_normalized, color="b")
plt.hist(clahe_ehnanced_image.flatten(), 256, [0, 256], color="r")
plt.legend(("CDF", "Histogram"))
plt.xlim([0, 256])
plt.title("Histogram of Image after CLAHE")
plt.show()
```

Output :

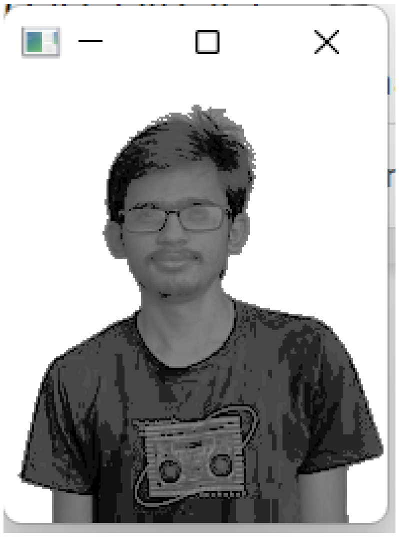
1. Base image converted to grayscale



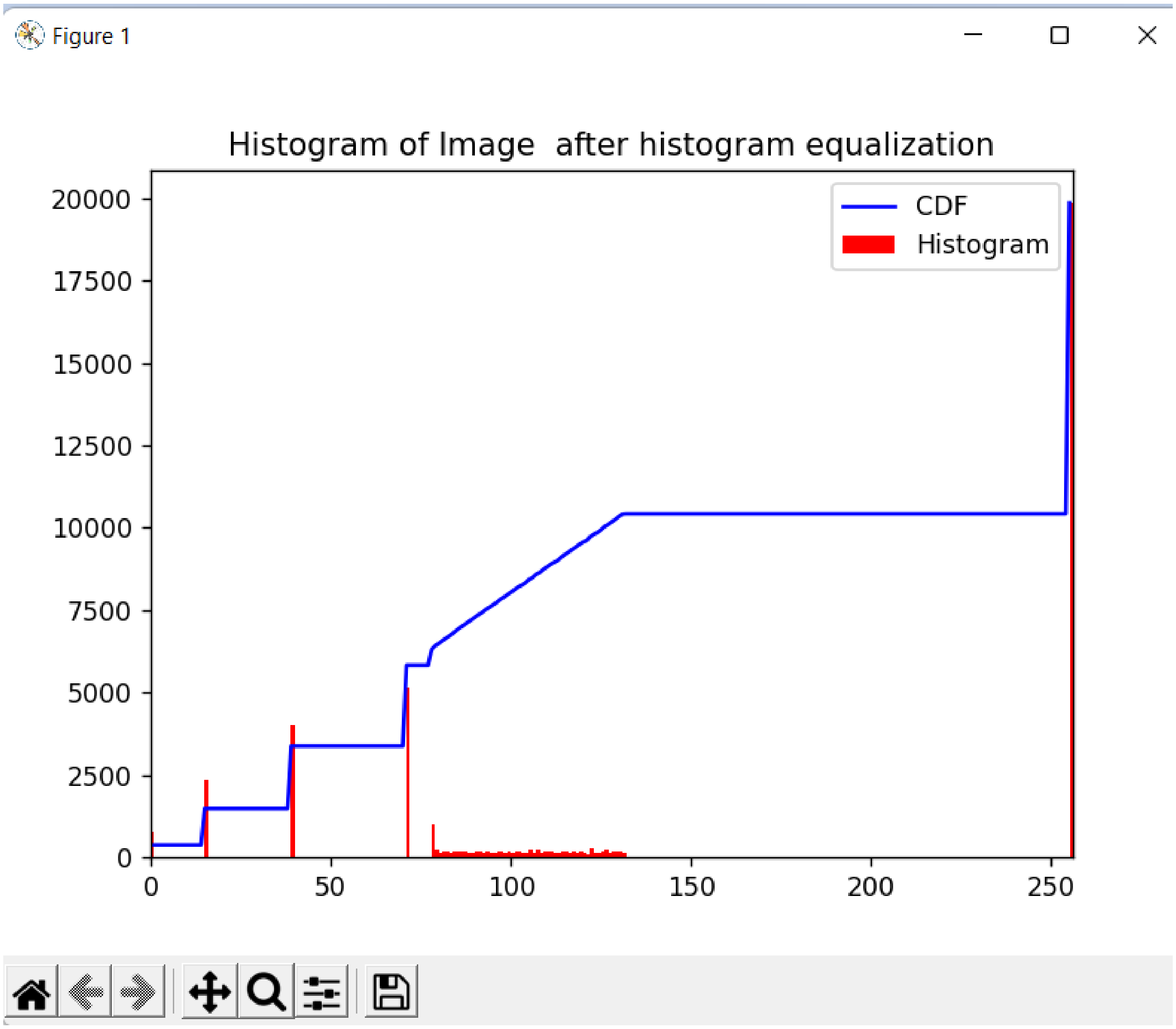
2. Image histogram of grayscale image before histogram equalization shown along with the cumulative distribution function of the pixels



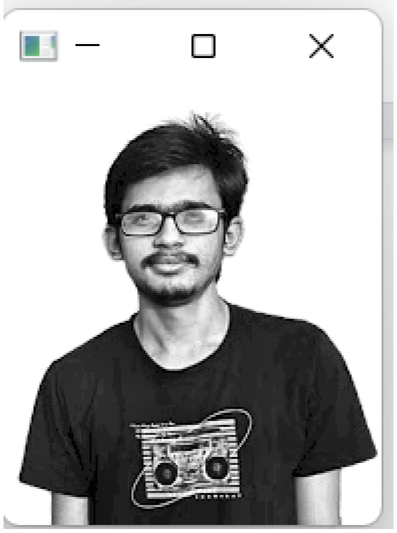
3. Image after histogram equalization



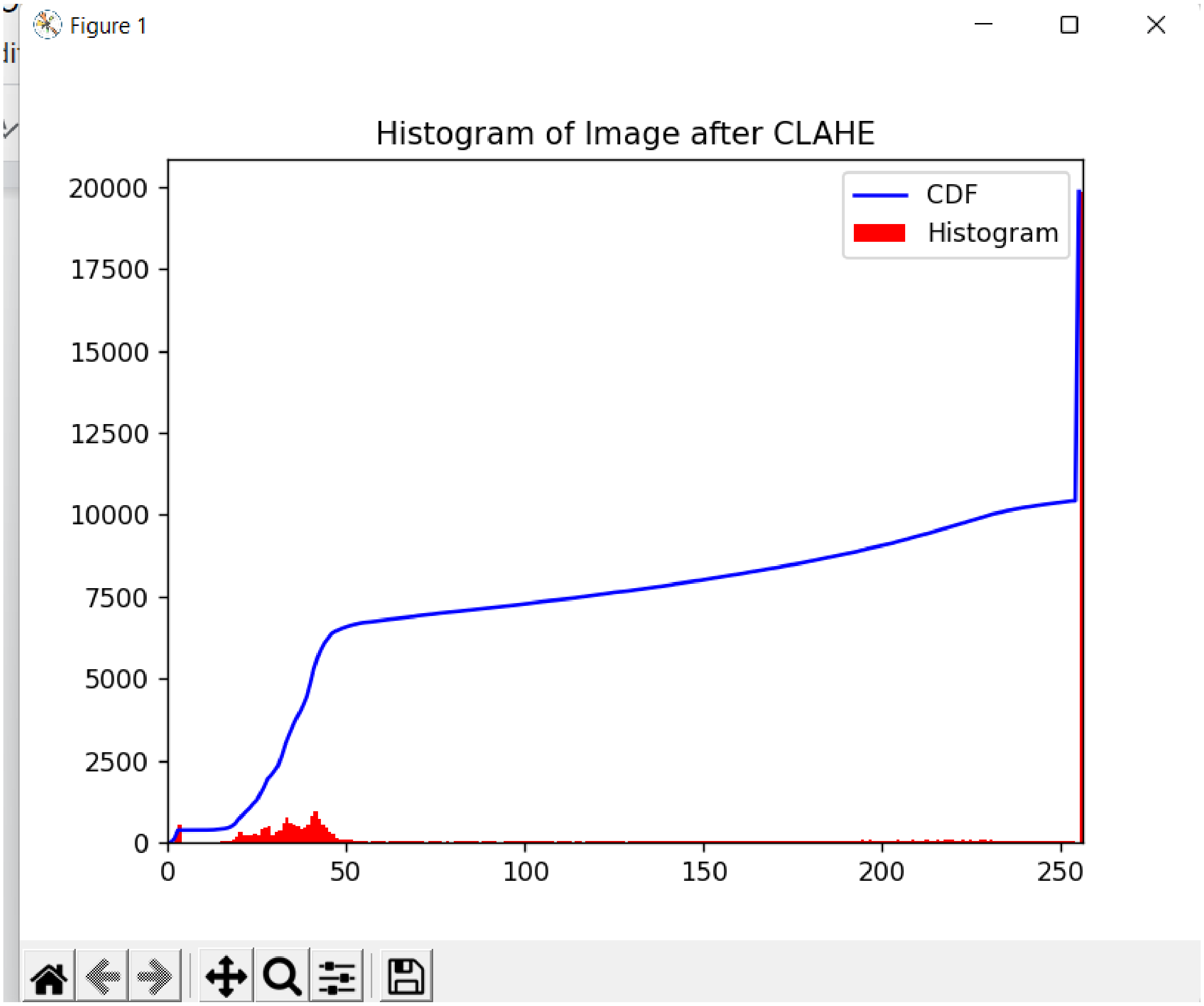
4. Image histogram of grayscale image after histogram equalization shown along with the cumulative distribution function of the pixels



5.CLAHE enhanced image



6. Image histogram after applying CLAHE



CONCLUSION:

- In this Experiment, We Studied Histogram Equalization. Also We came to know that it is especially effective in improving the visual quality of grayscale images

*********End of the Assignment*********