|  |  |
| --- | --- |
| **EXERCISE NO:1**  **DATE:09/01/2025** | **RGB TO GRAY**  **HISTOGRAM EQUILIZATION**  **IMAGE RESIZING** |

**AIM:**

To view the rgb image to grayscale image and histogram equilization of the grayscale image and the image resizing

**LENA IMAGE**

**ALGORITHM:**

1. Importing the library function of the open cv for image processing
2. Read the image using the imread function
3. Convert the image using function of the cvtColor in BGR to RGB
4. Execute the image
5. After the image is convert into the GRAY color using the function cvt Color in RGB to GRAY
6. Manuval code of the RGB to gray scale
7. Histogram equilization and using the RAVEL function and calculating the histogram using the CALC HIST and also the manuval code of the histogram equilization using the table
8. After that the resizing the image using the function resize

**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

img1 = cv2.imread(r"lena.png")

img3 = cv2.cvtColor(img1,cv2.COLOR\_BGR2RGB)

img\_2 = cv2.cvtColor(img3,cv2.COLOR\_RGB2GRAY)

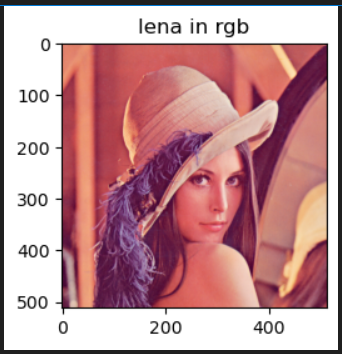
plt.figure(figsize=(6,5))

plt.subplot(121)

plt.title("lena in rgb")

plt.imshow(img3)

**OUTPUT:**



**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

plt.figure(figsize=(4,2))

plt.subplot(121)

plt.title("grayimage")

plt.imshow(img\_2 , cmap='gray')

red,green,blue = img1[:,:,0],img1[:,:,1],img1[:,:,2]

gamma=1

gray=0.299\*red\*\*gamma+0.587\*green\*\*gamma+0.114\*blue\*\*gamma

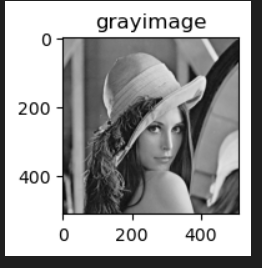
print(gray)

plt.figure(figsize=(4,5))

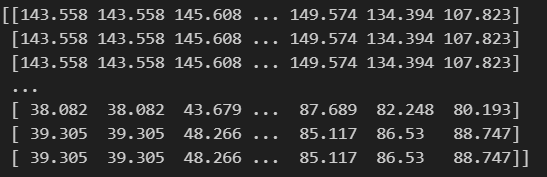
plt.subplot(121)

plt.imshow(gray,cmap='gray')

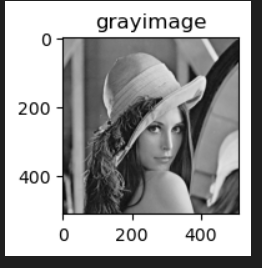
**OUTPUT:**

f

The grayscale image output



The manuval code of the the grayscale image



**HISTOGRAM EQUALIZATION PLOT USING THE LIBRARIES AND THE MANUVAL**

**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

img\_hist=cv2.calcHist([img1],[0],None,[256],[0,256])

plt.subplot(222)

plt.title("histogram")

plt.plot(img\_hist)

final\_image =cv2.equalizeHist(img\_2)

res=np.hstack((gray,final\_image))

plt.imshow(res,cmap='gray')

histg = cv2.calcHist([img\_2],[0],None,[256],[0,256])

plt.plot(histg)

plt.hist(img\_2.ravel(),256,[0,256])

plt.hist(img1.ravel(),256,[0,256])

**MANUVAL CODE:**

def manual\_histogram\_equalization(image):

    if len(image.shape) == 3:

        image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

    hist, bins = np.histogram(image.flatten(), 256, [0, 256])

    cdf = hist.cumsum()

    cdf\_normalized = cdf \* hist.max() / cdf.max()

    cdf\_m = np.ma.masked\_equal(cdf, 0)

    cdf\_m = (cdf\_m - cdf\_m.min()) \* 255 / (cdf\_m.max() - cdf\_m.min())

    cdf = np.ma.filled(cdf\_m, 0).astype('uint8')

    img\_equalized = cdf[image]  #

    return img\_equalized

img1 = cv2.imread(r"lena.png")

if img1 is None:

    print("Error: Image not loaded. Check the file path.")

else:

    img\_equalized = manual\_histogram\_equalization(img1)

    plt.figure(figsize=(10, 5))

    # Original Image

    plt.subplot(1, 2, 1)

    plt.imshow(cv2.cvtColor(img1, cv2.COLOR\_BGR2RGB))

    plt.title('Original Image')

    plt.axis('off')

    # Equalized Image

    plt.subplot(1, 2, 2)

    plt.imshow(cv2.cvtColor(img\_equalized, cv2.COLOR\_BGR2RGB))

    plt.title('Histogram Equalized Image')

    plt.axis('off')

    plt.show()\

**HISTOGRAM CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread(r"lena.png")

if img is None:

    print("Error: Image not loaded. Check the file path.")

else:

    gray\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

    hist = cv2.calcHist([gray\_img], [0], None, [256], [0, 256])

    plt.figure(figsize=(10, 5))

    plt.title('Histogram of Grayscale Image')

    plt.xlabel('Pixel Intensity')

    plt.ylabel('Frequency')

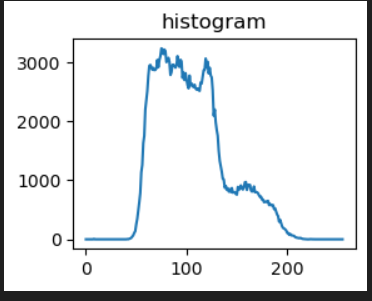
    plt.plot(hist)

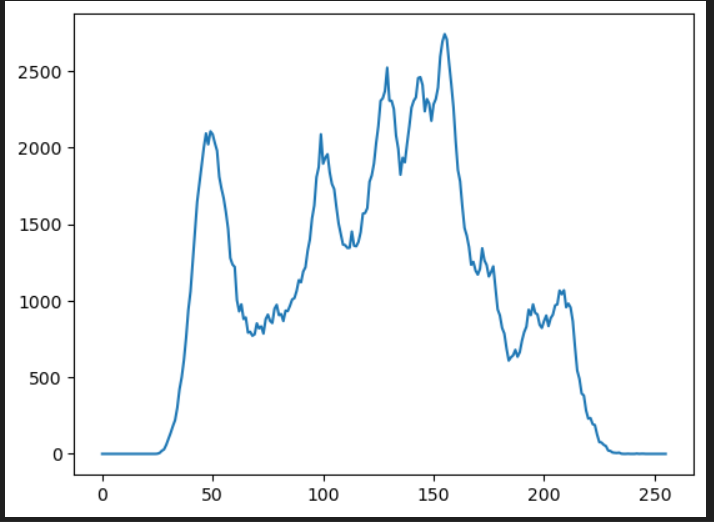
    plt.xlim([0, 256])

    plt.show()

**OUTPUT:**

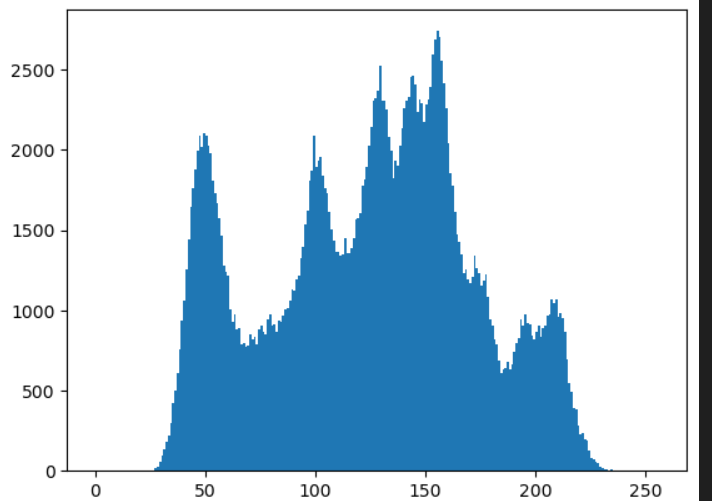
**Using libraries output**

****

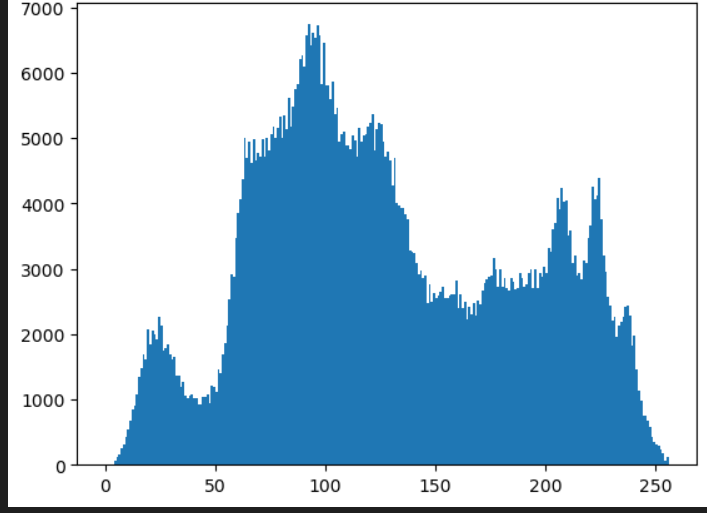


**Histogram equalize output**

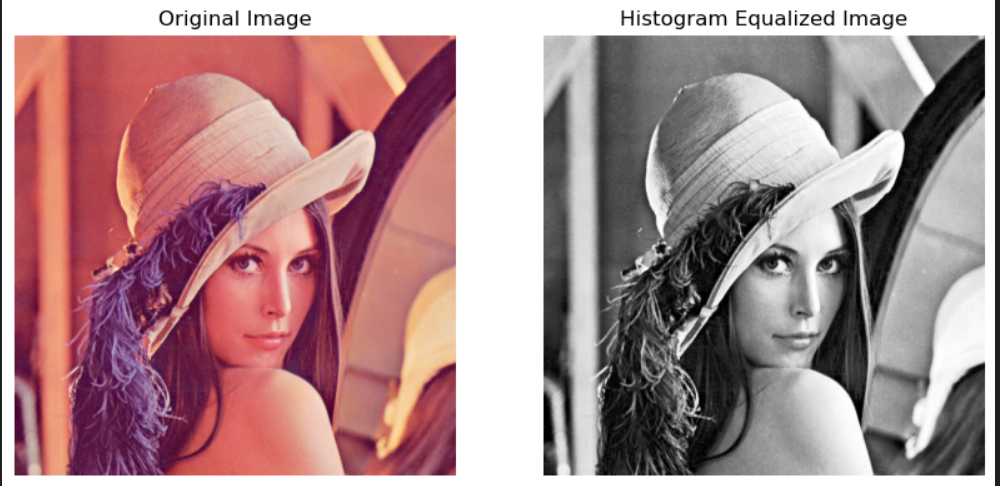


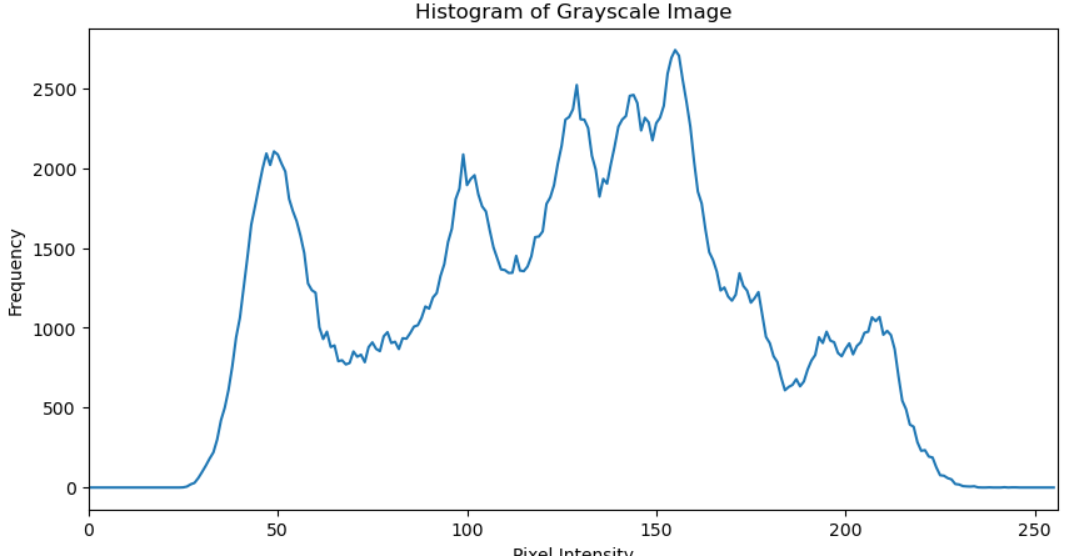


**Histogram output of example 1 :**

****

**Histogram equalize output of manuval :**





**RESIZING THE IMAGE**

**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

resized\_image = cv2.resize(img\_2, (1000, 1000))

plt.imshow(img\_2,cmap='grey')

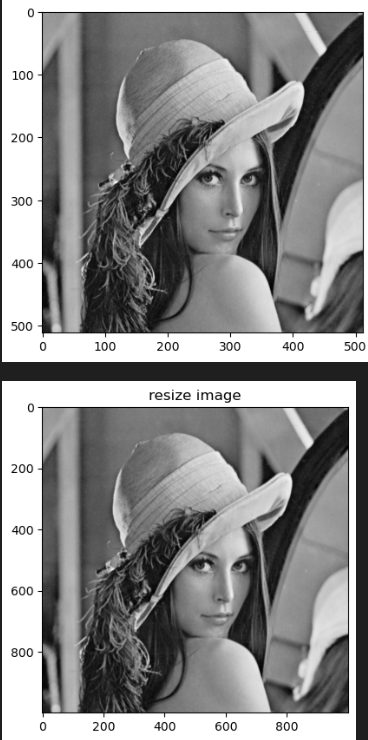
plt.figure(figsize=(10,10))

plt.subplot(121)

plt.imshow(resized\_image,cmap='grey')

plt.title("resize image")

**OUTPUT:**

****

**SUNG IMAGE**

**ALGORITHM:**

1. Importing the library function of the open cv for image processing
2. Read the image using the imread function
3. Convert the image using function of the cvtColor in BGR to RGB
4. Execute the image
5. After the image is convert into the GRAY color using the function cvt Color in RGB to GRAY
6. Manuval code of the RGB to gray scale
7. Histogram equilization and using the RAVEL function and calculating the histogram using the CALC HIST and also the manuval code of the histogram equilization using the table
8. After that the resizing the image using the function resize

**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

img1 = cv2.imread(r"SUNG.png")

img3 = cv2.cvtColor(img1,cv2.COLOR\_BGR2RGB)

img\_2 = cv2.cvtColor(img3,cv2.COLOR\_RGB2GRAY)

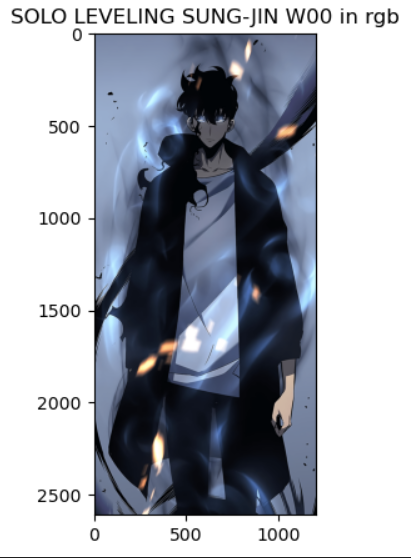
plt.figure(figsize=(6,5))

plt.subplot(121)

plt.title("lena in rgb")

plt.imshow(img3)

**OUTPUT:**



**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

plt.figure(figsize=(4,2))

plt.subplot(121)

plt.title("grayimage")

plt.imshow(img\_2 , cmap='gray')

red,green,blue = img1[:,:,0],img1[:,:,1],img1[:,:,2]

gamma=1

gray=0.299\*red\*\*gamma+0.587\*green\*\*gamma+0.114\*blue\*\*gamma

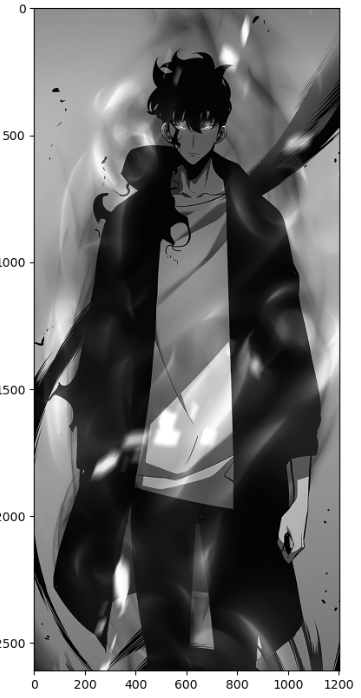
print(gray)

plt.figure(figsize=(4,5))

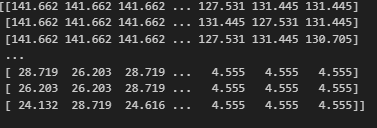
plt.subplot(121)

plt.imshow(gray,cmap='gray')

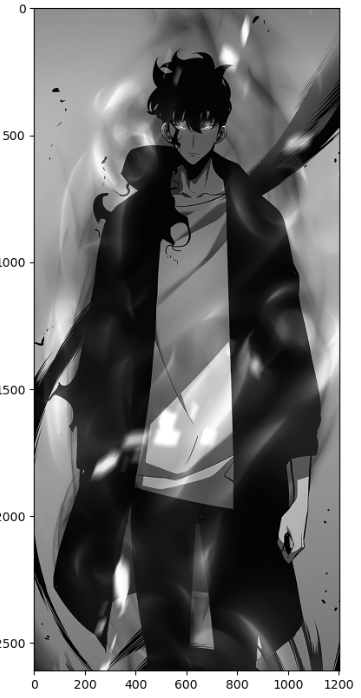
**OUTPUT:**



The grayscale image output



The manuval code of the the grayscale image



**HISTOGRAM EQUALIZATION PLOT USING THE LIBRARIES AND THE MANUVAL**

**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

img\_hist=cv2.calcHist([img1],[0],None,[256],[0,256])

plt.subplot(222)

plt.title("histogram")

plt.plot(img\_hist)

final\_image =cv2.equalizeHist(img\_2)

res=np.hstack((gray,final\_image))

plt.imshow(res,cmap='gray')

histg = cv2.calcHist([img\_2],[0],None,[256],[0,256])

plt.plot(histg)

plt.hist(img\_2.ravel(),256,[0,256])

plt.hist(img1.ravel(),256,[0,256])

**MANUVAL CODE:**

def manual\_histogram\_equalization(image):

    if len(image.shape) == 3:

        image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

    hist, bins = np.histogram(image.flatten(), 256, [0, 256])

    cdf = hist.cumsum()

    cdf\_normalized = cdf \* hist.max() / cdf.max()

    cdf\_m = np.ma.masked\_equal(cdf, 0)

    cdf\_m = (cdf\_m - cdf\_m.min()) \* 255 / (cdf\_m.max() - cdf\_m.min())

    cdf = np.ma.filled(cdf\_m, 0).astype('uint8')

    img\_equalized = cdf[image]  #

    return img\_equalized

img1 = cv2.imread(r"SUNG.png")

if img1 is None:

    print("Error: Image not loaded. Check the file path.")

else:

    img\_equalized = manual\_histogram\_equalization(img1)

    plt.figure(figsize=(10, 5))

    # Original Image

    plt.subplot(1, 2, 1)

    plt.imshow(cv2.cvtColor(img1, cv2.COLOR\_BGR2RGB))

    plt.title('Original Image')

    plt.axis('off')

    # Equalized Image

    plt.subplot(1, 2, 2)

    plt.imshow(cv2.cvtColor(img\_equalized, cv2.COLOR\_BGR2RGB))

    plt.title('Histogram Equalized Image')

    plt.axis('off')

    plt.show()\

**HISTOGRAM CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread(r"SUNG.png")

if img is None:

    print("Error: Image not loaded. Check the file path.")

else:

    gray\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

    hist = cv2.calcHist([gray\_img], [0], None, [256], [0, 256])

    plt.figure(figsize=(10, 5))

    plt.title('Histogram of Grayscale Image')

    plt.xlabel('Pixel Intensity')

    plt.ylabel('Frequency')

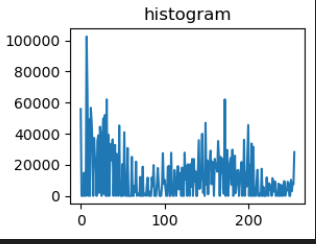
    plt.plot(hist)

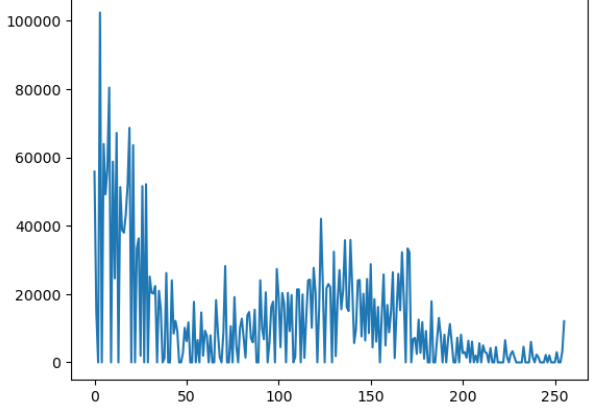
    plt.xlim([0, 256])

    plt.show()

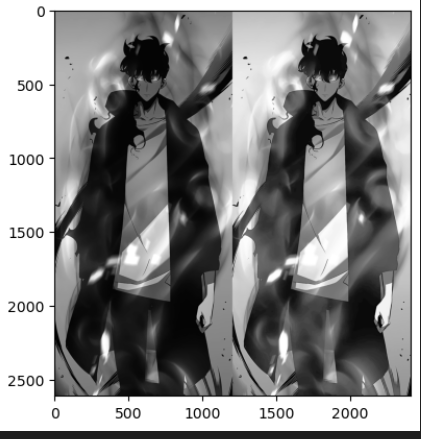
**OUTPUT:**

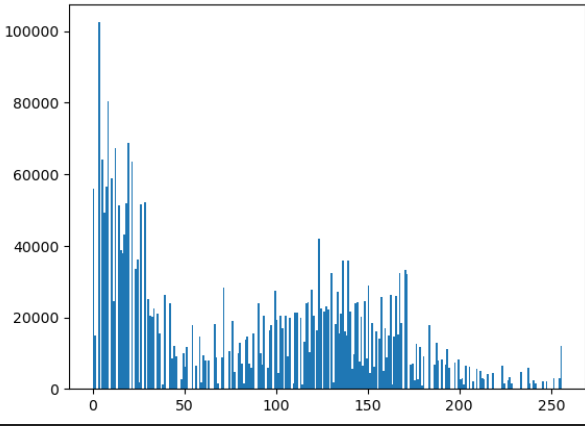
**Using libraries output**

****

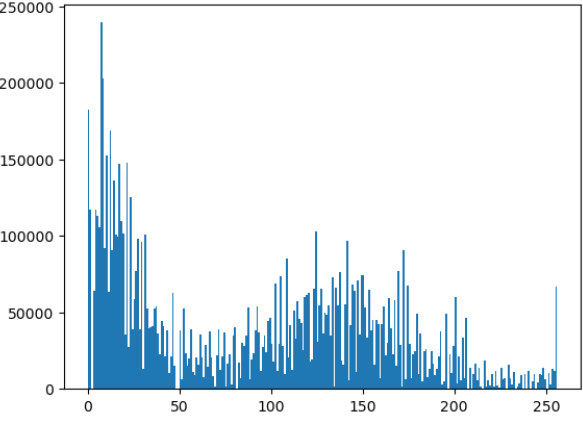


**Histogram equalize output**

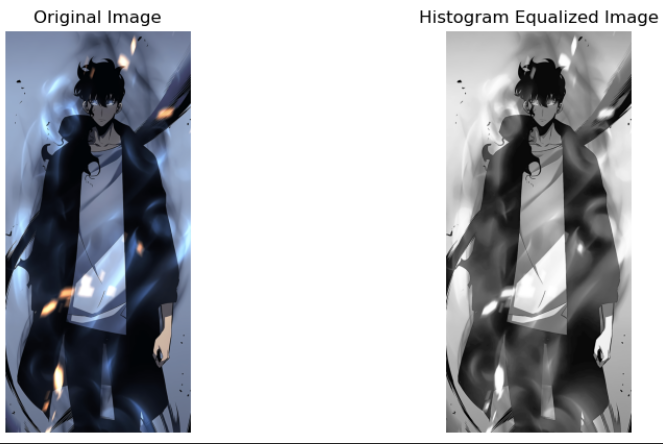


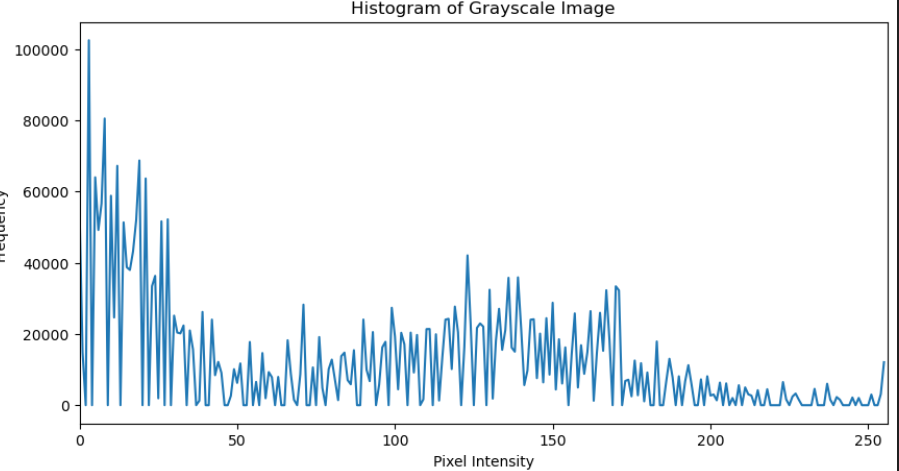


**Histogram output of example 1 :**

****

**Histogram equalize output of manuval :**





**RESIZING THE IMAGE**

**CODE:**

import cv2

import numpy as np

import matplotlib.pyplot as plt

resized\_image = cv2.resize(img\_2, (3000, 5000))

plt.imshow(img\_2,cmap='grey')

plt.figure(figsize=(10,10))

plt.subplot(121)

plt.imshow(resized\_image,cmap='grey')

plt.title("resize image")

**OUTPUT:**

****

**RESULT:**

**We have successfully executed basic image and functions to reize image histogram equilization and RGB to GRAY for lena and sung image**