

**Aim:**

To study and perform Histogram processing in image enhancement.

- Plot a Histogram of Grayscale and Color Image
- Contrast and Brightness stretching of Grayscale image
- Perform Histogram Equalization on Grayscale image

**1) Plot a Histogram of Grayscale and Color Image****Input Image:****Code:**

```
import cv2
import matplotlib.pyplot as plt

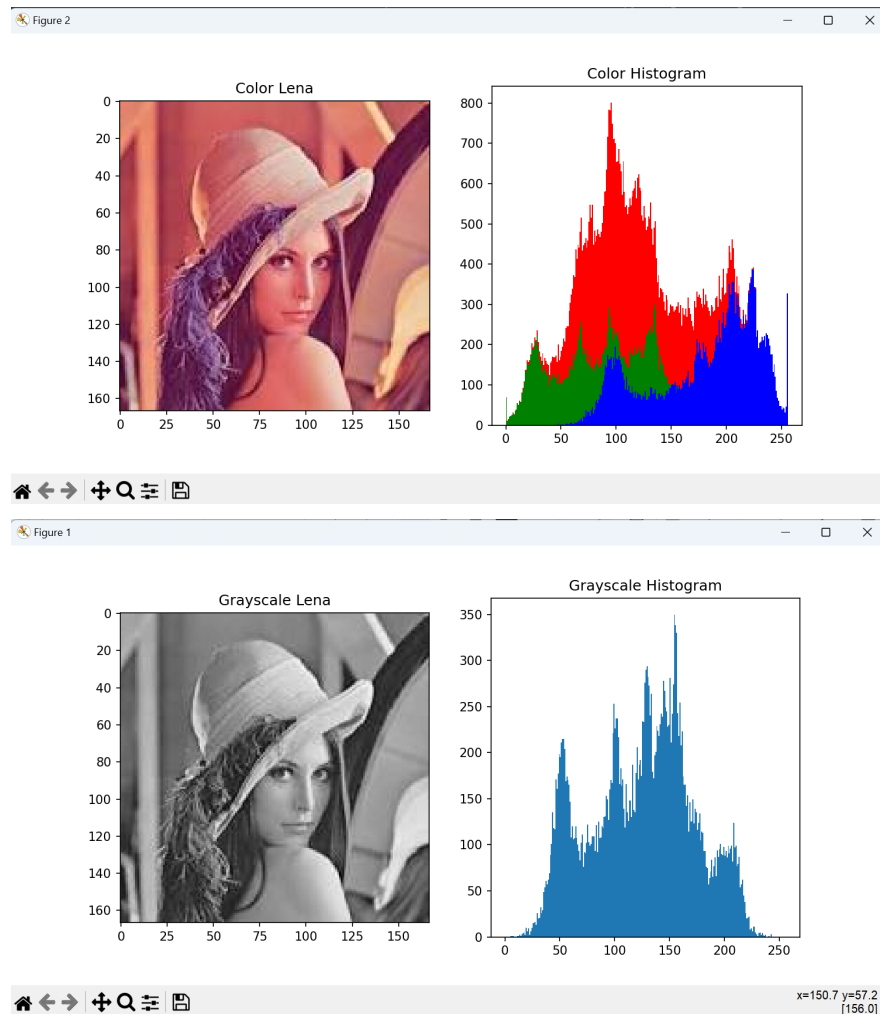
# Load the grayscale and color Lena images
lena_gray = cv2.imread('lena.jpg', cv2.IMREAD_GRAYSCALE)
lena_color = cv2.imread('lena.jpg', cv2.IMREAD_COLOR)

# Plot the grayscale image and its histogram
plt.figure(figsize=(10, 5))
plt.subplot(121)
plt.imshow(lena_gray, cmap='gray')
plt.title('Grayscale Lena')
plt.subplot(122)
plt.hist(lena_gray.ravel(), bins=256, range=(0, 256))
plt.title('Grayscale Histogram')

# Plot the color image and its histogram
plt.figure(figsize=(10, 5))
plt.subplot(121)
```

```
plt.imshow(cv2.cvtColor(lena_color, cv2.COLOR_BGR2RGB))
plt.title('Color Lena')
plt.subplot(122)
plt.hist(lena_color.ravel(), bins=256, range=(0, 256), color='r')
plt.hist(lena_color[:, :, 1].ravel(), bins=256, range=(0, 256), color='g')
plt.hist(lena_color[:, :, 2].ravel(), bins=256, range=(0, 256), color='b')
plt.title('Color Histogram')
plt.show()
```

**Output:**



## 2) Contrast and Brightness Stretching of grayscale Image

**Input Image:**

**Code:**

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

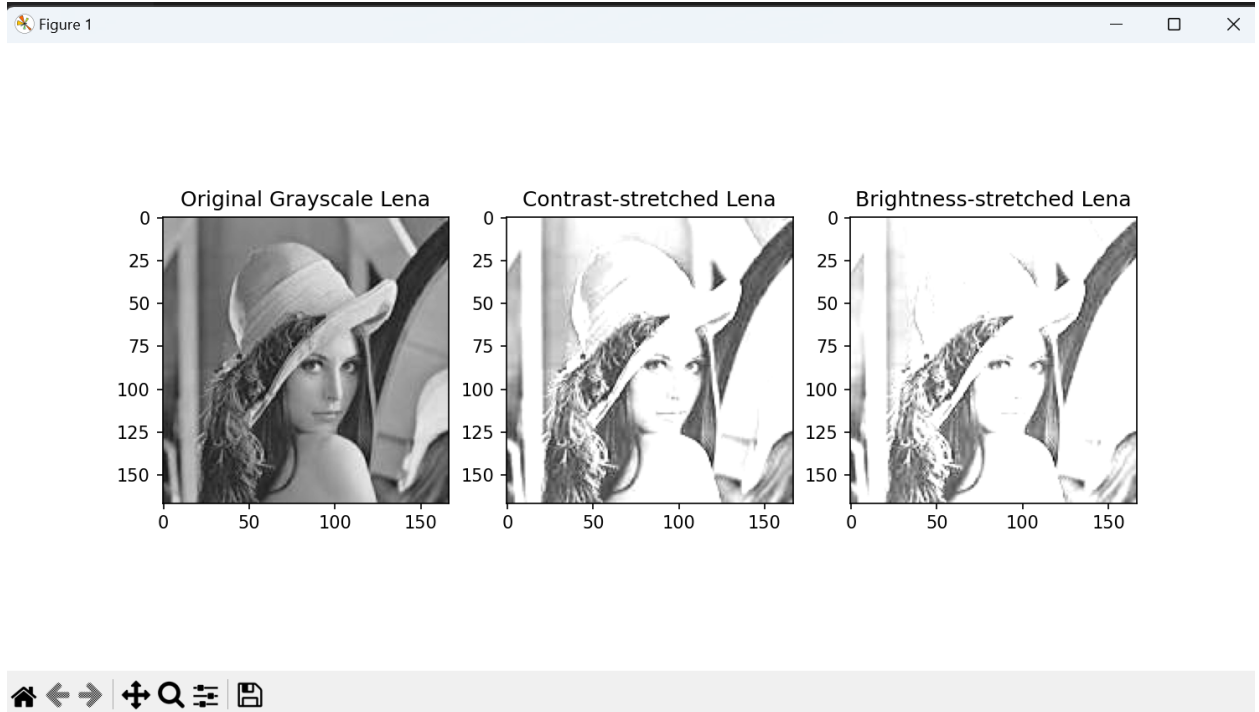
# Load the grayscale Lena image
lena_gray = cv2.imread('lena.jpg', cv2.IMREAD_GRAYSCALE)

# Set the contrast and brightness stretching parameters
alpha = 2.0 # contrast control
beta = 50 # brightness control

# Apply contrast and brightness stretching
lena_contrast = cv2.convertScaleAbs(lena_gray, alpha=alpha)
lena_brightness = cv2.convertScaleAbs(lena_contrast, beta=beta)

# Display the original and modified images
plt.figure(figsize=(10, 5))
plt.subplot(131)
plt.imshow(lena_gray, cmap='gray')
plt.title('Original Grayscale Lena')
plt.subplot(132)
plt.imshow(lena_contrast, cmap='gray')
plt.title('Contrast-stretched Lena')
plt.subplot(133)
plt.imshow(lena_brightness, cmap='gray')
plt.title('Brightness-stretched Lena')
plt.show()
```

**Output:**



### 3) Perform Histogram Equalization on Grey Scale Image

**Input Image:**



**Code:**

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

# Load the Lena image
img = cv2.imread('lena.jpg',0)
# In greyscale pixel is minimum
```

```

#The calcHist function calculates a histogram for a single-channel (grayscale) image.
# Calculate the histogram
hist = cv2.calcHist([img],[0],None,[256],[0,256])

# Apply histogram equalization
equ = cv2.equalizeHist(img)

# Calculate the histogram of the equalized image
hist_equ = cv2.calcHist([equ],[0],None,[256],[0,256])

# Plot the original and equalized images and their histograms
plt.subplot(221), plt.imshow(img, cmap='gray')
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(222), plt.imshow(equ, cmap='gray')
plt.title('Equalized Image'), plt.xticks([]), plt.yticks([])
plt.subplot(223), plt.plot(hist)
plt.title('Original Histogram'), plt.xlim([0,256])
plt.subplot(224), plt.plot(hist_equ)
plt.title('Equalized Histogram'), plt.xlim([0,256])
plt.show()

```

**Output:**

