

Fig 1

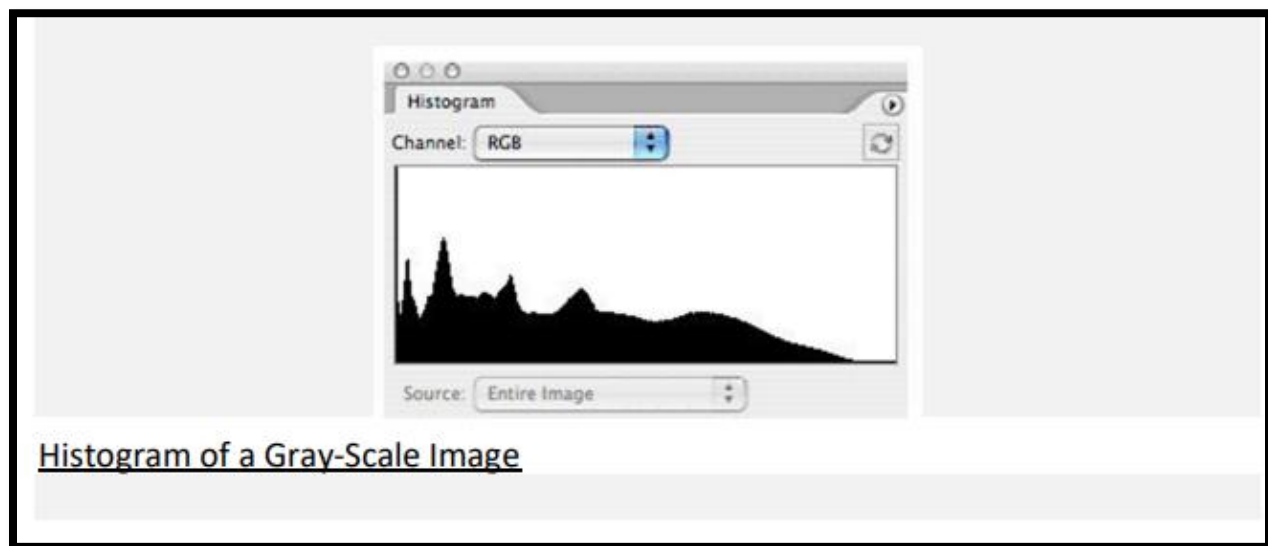


Fig 2

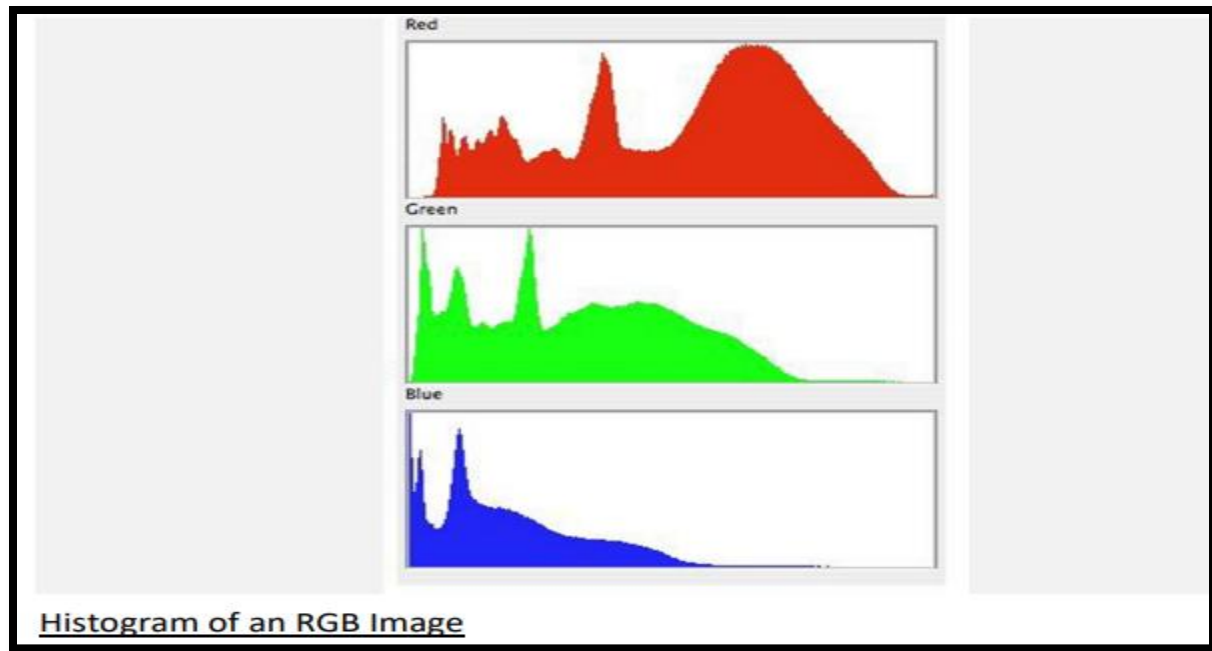


Fig 3

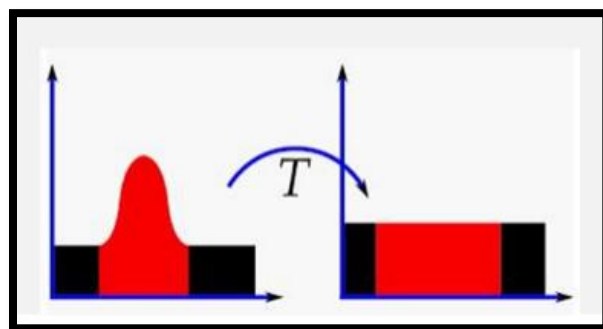


Fig 4



Fig 5

Source Code :

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

# Read the image
image = cv2.imread('practical_a2_input.jpeg', cv2.IMREAD_GRAYSCALE)

# Calculate the histogram of the original image
hist_original, bins_original = np.histogram(image.flatten(), 256, [0, 256])

# Compute the CDF of the original histogram
cdf_original = hist_original.cumsum()
cdf_normalized_original = cdf_original * hist_original.max() / cdf_original.max()

# Histogram equalization
equ = cv2.equalizeHist(image)

# Calculate the histogram of the enhanced image
hist_enhanced, bins_enhanced = np.histogram(equ.flatten(), 256, [0, 256])

# Compute the CDF of the enhanced histogram
cdf_enhanced = hist_enhanced.cumsum()
cdf_normalized_enhanced = cdf_enhanced * hist_enhanced.max() / cdf_enhanced.max()

# Plotting
plt.figure(figsize=(12, 6))

plt.subplot(2, 2, 1)
plt.imshow(image, cmap='gray')
plt.title('Original Image')

plt.subplot(2, 2, 2)
plt.plot(cdf_normalized_original, color='b')
plt.hist(image.flatten(), 256, [0, 256], color='r')
plt.title('Histogram of Original Image')

plt.subplot(2, 2, 3)
plt.imshow(equ, cmap='gray')
plt.title('Equalized Image')

plt.subplot(2, 2, 4)
plt.plot(cdf_normalized_enhanced, color='b')
plt.hist(equ.flatten(), 256, [0, 256], color='r')
plt.title('Histogram of Equalized Image')
```

`plt.show()`

Output :

