**import** cv2

**import** numpy **as** np

**from** matplotlib **import** pyplot **as** plt

*# Load the image*

image\_path **=** "D:\DSIP\codes\dark.jpeg"  *# Replace with the path to your image*

image **=** cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

*# Calculate the histogram*

histogram **=** cv2.calcHist([image], [0], None, [256], [0, 256])

*# Plot the histogram*

plt.figure(*figsize***=**(8, 6))

plt.title('Histogram')

plt.xlabel('Pixel Value')

plt.ylabel('Frequency')

plt.plot(histogram)

plt.xlim([0, 256])

plt.grid(True)

plt.show()

*# Perform histogram equalization*

equalized\_image **=** cv2.equalizeHist(image)

*# Display the original and equalized images*

plt.figure(*figsize***=**(10, 5))

plt.subplot(1, 2, 1)

plt.title('Original Image')

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

plt.axis('off')

plt.subplot(1, 2, 2)

plt.title('Equalized Image')

plt.imshow(equalized\_image, *cmap***=**'gray')

plt.axis('off')

*# Calculate the histogram*

histogram **=** cv2.calcHist([equalized\_image], [0], None, [256], [0, 256])

*# Plot the histogram*

plt.figure(*figsize***=**(8, 6))

plt.title('Histogram')

plt.xlabel('Pixel Value')

plt.ylabel('Frequency')

plt.plot(histogram)

plt.xlim([0, 256])

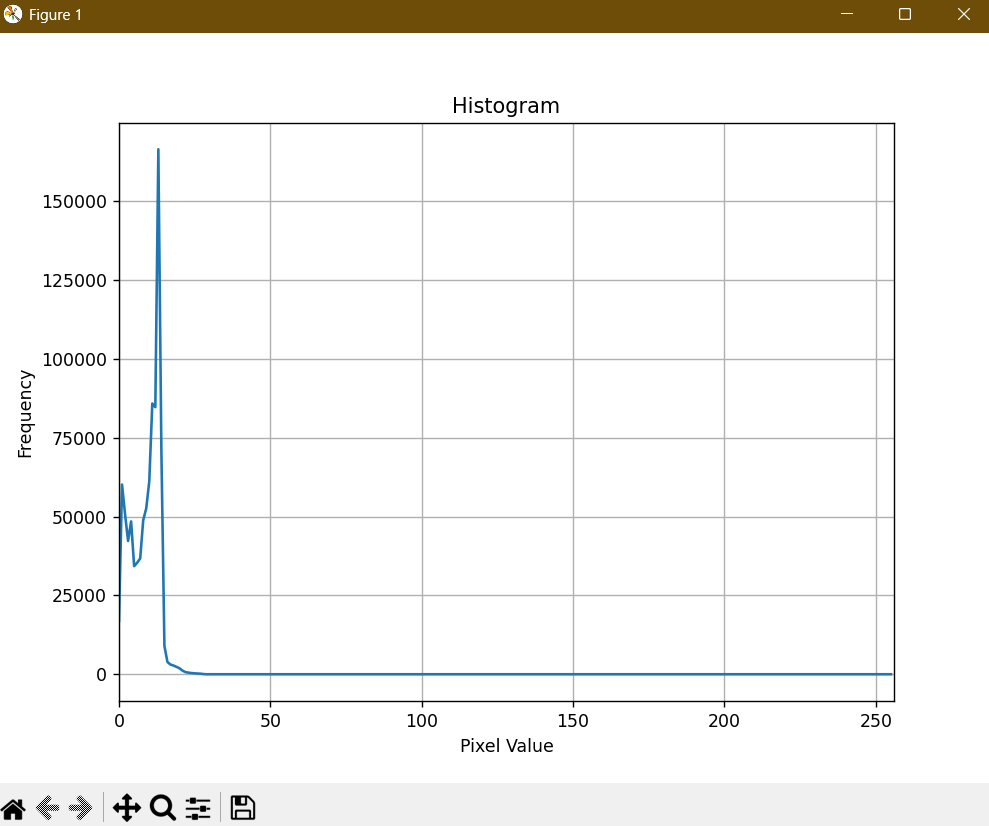
plt.grid(True)

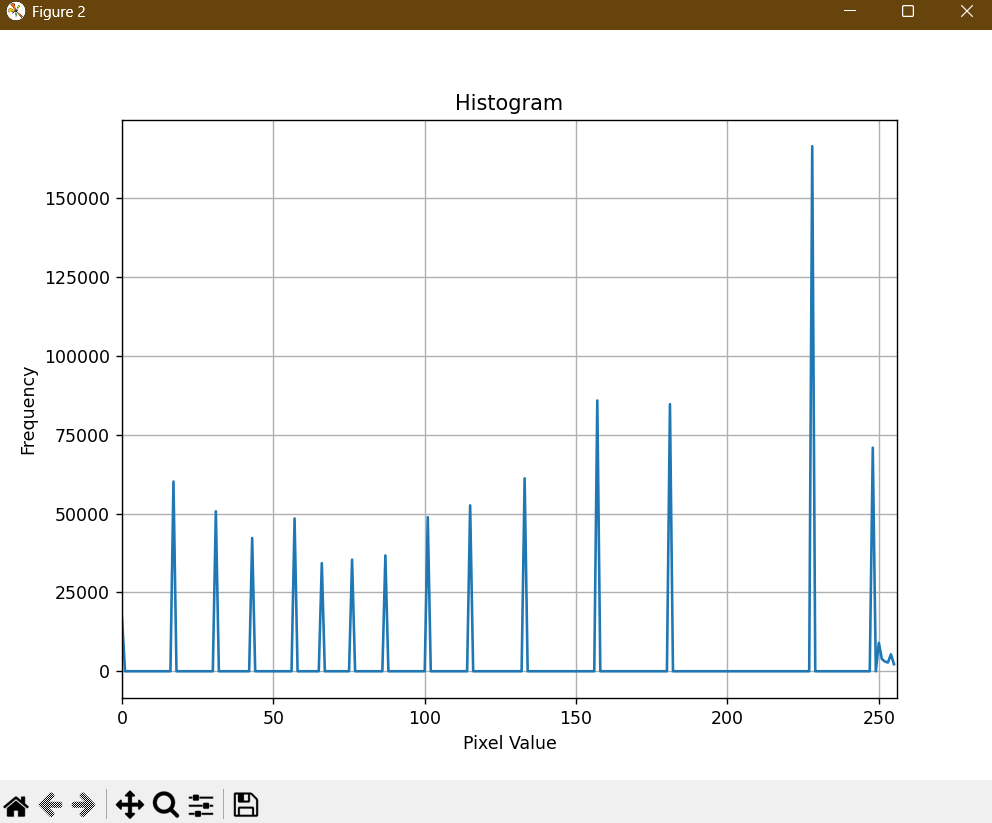
plt.show()

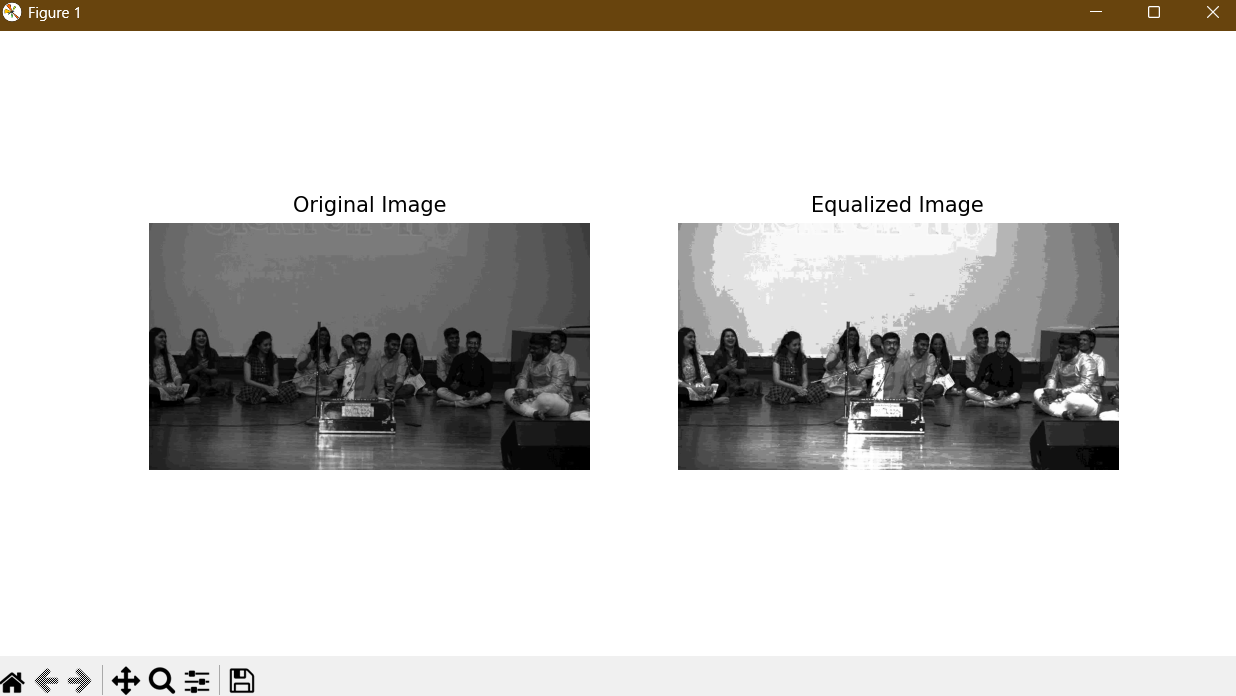
plt.tight\_layout()

plt.show()

Output:







import cv2

import numpy as np

from matplotlib import pyplot as plt

# Load the source and reference images

source\_path = '/content/IMG\_20230915\_135931789.jpg'

reference\_path = '/content/matched\_image.jpg'

source\_image = cv2.imread(source\_path, cv2.IMREAD\_GRAYSCALE)

reference\_image = cv2.imread(reference\_path, cv2.IMREAD\_GRAYSCALE)

# Calculate histograms for the source and reference images

source\_hist = cv2.calcHist([source\_image], [0], None, [256], [0, 256])

reference\_hist = cv2.calcHist([reference\_image], [0], None, [256], [0, 256])

# Normalize histograms to have sum equal to 1

source\_hist /= source\_hist.sum()

reference\_hist /= reference\_hist.sum()

# Calculate cumulative distribution functions (CDF) for histograms

source\_cdf = source\_hist.cumsum()

reference\_cdf = reference\_hist.cumsum()

# Perform histogram matching by mapping source CDF to reference CDF

mapping = np.interp(source\_cdf, reference\_cdf, range(256))

matched\_image = mapping[source\_image]

# Convert to uint8 data type

matched\_image = matched\_image.astype(np.uint8)

# Display the images using Matplotlib

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

plt.subplot(131)

plt.title('Source Image')

plt.imshow(source\_image, cmap='gray')

plt.axis('off')

plt.subplot(132)

plt.title('Reference Image')

plt.imshow(reference\_image, cmap='gray')

plt.axis('off')

plt.subplot(133)

plt.title('Matched Image')

plt.imshow(matched\_image, cmap='gray')

plt.axis('off')

plt.tight\_layout()

plt.show()