 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: DSIP (01CT0513)</b>	<b>Aim: - Histogram for images</b>	
<b>Experiment: - 8</b>	<b>Date: -</b>	<b>Enrollment No: - 92301733041</b>

Code:

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

# Load the image
image_path = 'DSIP/ex1_4.png' # 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')
```

 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: DSIP (01CT0513)</b>	<b>Aim: - Histogram for images</b>	
<b>Experiment: - 8</b>	<b>Date: -</b>	<b>Enrollment No: - 92301733041</b>

```
plt.xlabel('Pixel Value')
plt.ylabel('Frequency')
plt.plot(histogram)
plt.xlim([0, 256])
plt.grid(True)
plt.show()
plt.tight_layout()
plt.show()

# Load the source and reference images
source_path =image_path
reference_path = 'images.jpeg' # Replace with the path to your reference
image
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)
matched_hist = cv2.calcHist([matched_image], [0], None, [256], [0, 256])

#plot histogram of source and reference images and matched image
plt.figure(figsize=(12, 6))
plt.subplot(131)
plt.title('Source Image Histogram')
```

 <b>Marwadi University</b> Marwadi Chandarana Group	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: DSIP (01CT0513)</b>	<b>Aim: - Histogram for images</b>	
<b>Experiment: - 8</b>	<b>Date: -</b>	<b>Enrollment No: - 92301733041</b>

```
plt.plot(source_hist)
plt.xlim([0, 256])
plt.subplot(132)
plt.title('Reference Image Histogram')
plt.plot(reference_hist)
plt.xlim([0, 256])
plt.subplot(133)
plt.title('Matched Image Histogram')
plt.plot(matched_hist, color='green')
plt.xlabel('Pixel Value')
plt.legend()
plt.xlim([0, 256])
plt.tight_layout()
plt.show()

# 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()
```

Output:

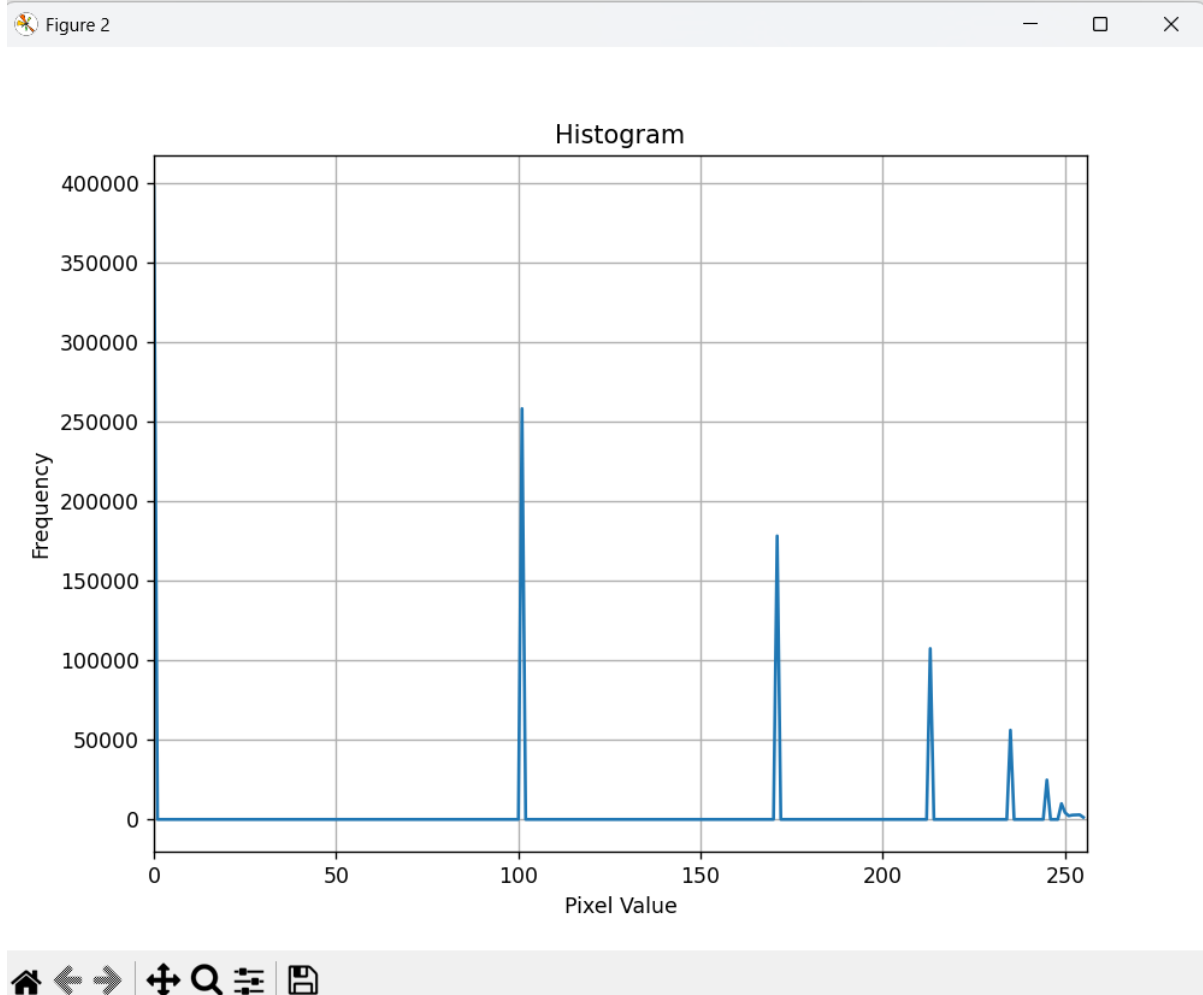
**Subject: DSIP (01CT0513)**



**Aim: - Histogram for images**

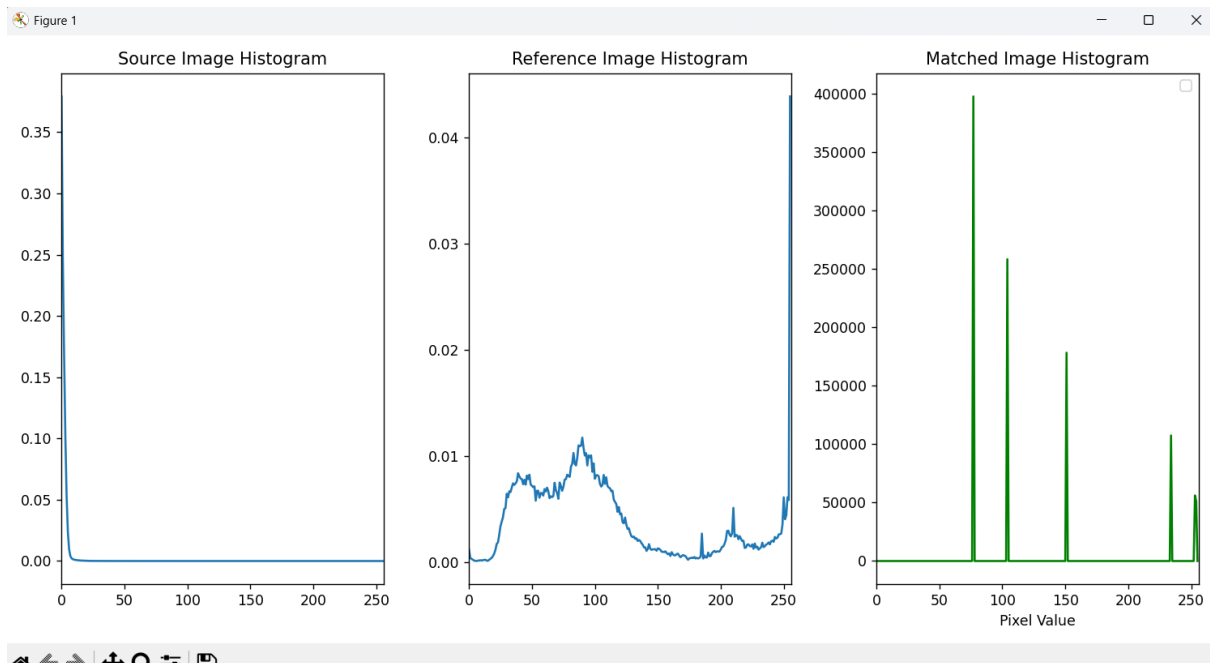
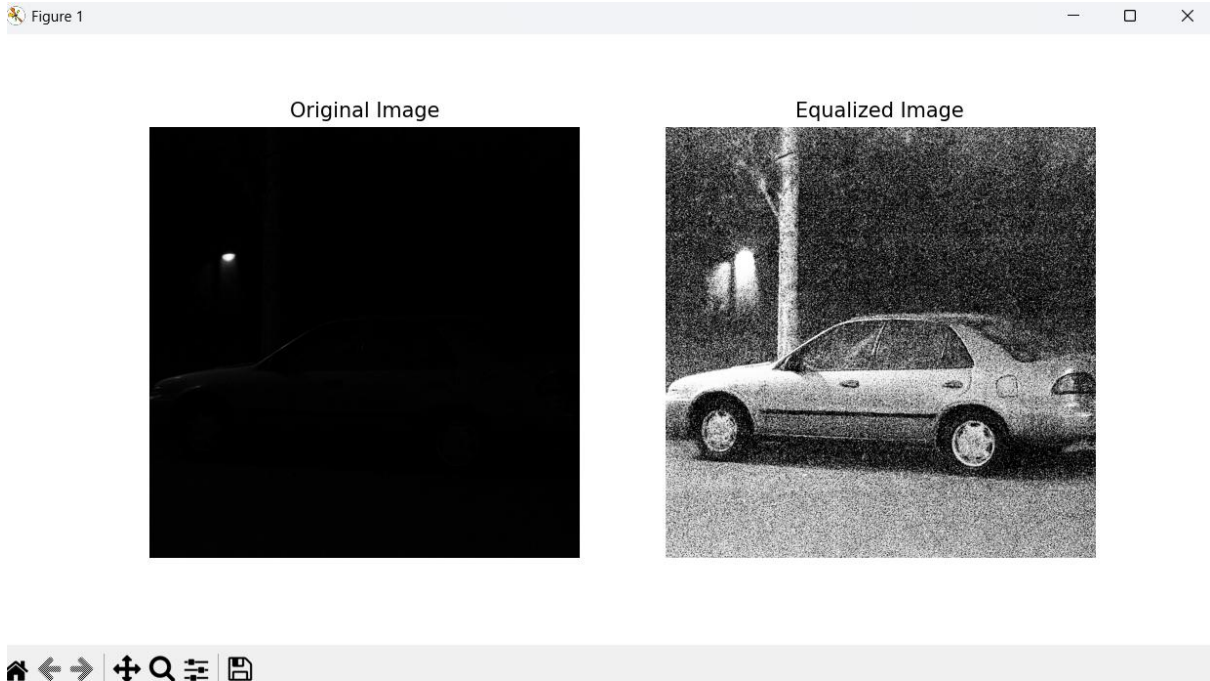
**Experiment: - 8**



**Date: -**

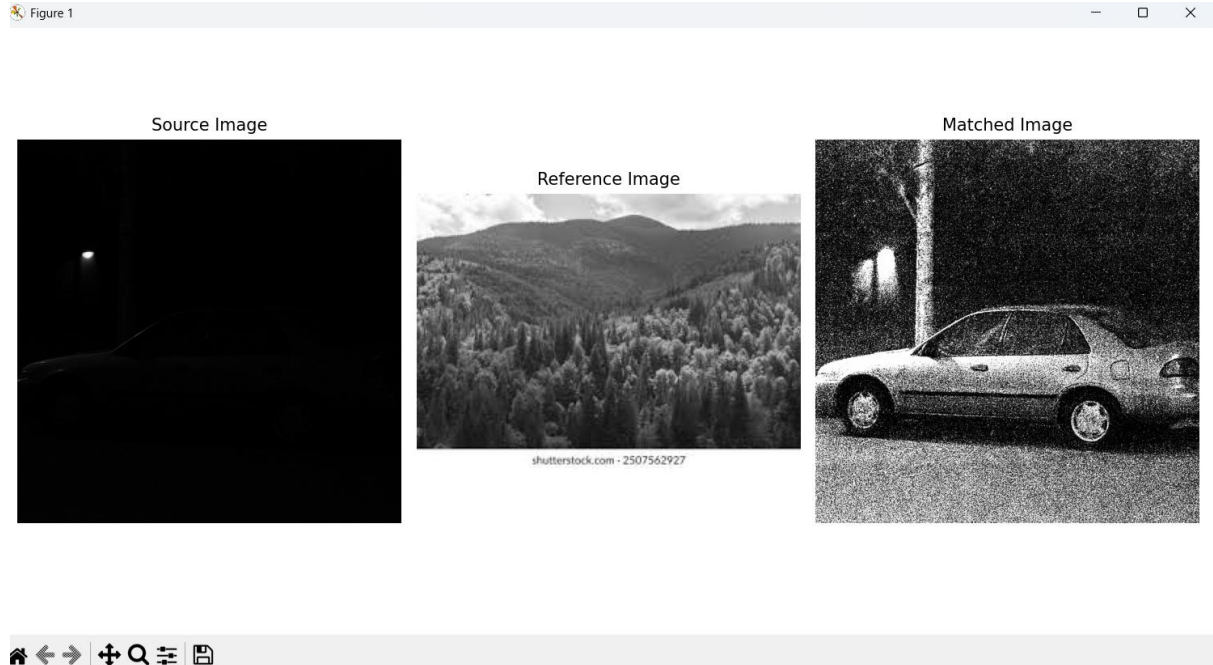
**Enrollment No: - 92301733041**



<div>  <div> <b>Marwadi University</b>  Marwadi Chandarana Group </div> <div> NAAC   </div> </div>		<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: DSIP (01CT0513)</b>		<b>Aim: - Histogram for images</b>	
<b>Experiment: - 8</b>		<b>Date: -</b>	<b>Enrollment No: - 92301733041</b>



 <b>Marwadi University</b> Marwadi Chandarana Group 	<b>Marwadi University</b> <b>Faculty of Engineering &amp; Technology</b> <b>Department of Information and Communication Technology</b>	
<b>Subject: DSIP (01CT0513)</b>	<b>Aim: - Histogram for images</b>	
<b>Experiment: - 8</b>	<b>Date: -</b>	<b>Enrollment No: - 92301733041</b>



**Conclusion:** In this exp I learnt about histogram for images , how to calculate it using pdf and cdf functions also in programming by available cv2 direct functions also got to know about match by referencing image where I can update the blurry or fade image to clear or enhance by referencing or matching with the bright image of the same scene.