

Histogram-of-an-images

Aim

To obtain a histogram for finding the frequency of pixels in an Image with pixel values ranging from 0 to 255. Also write the code using OpenCV to perform histogram equalization.

Software Required:

Anaconda - Python 3.7

Algorithm:

Step1:

Read the gray imread()

Step2:

Print the image using imshow().

Step3:

Use calcHist() function to mark the image in graph frequency for gray and color image.

step4:

Use calcHist() function to mark the image in graph frequency for gray and color image.

Step5:

The Histogram of gray scale image and color image is shown.

Program:

```
python
```

Developed By: MOHAMED RASHITH S

Register Number: 212223243003

Input Grayscale Image

```
import cv2
from matplotlib import pyplot as plt
# Load the color image
image = cv2.imread('image.jpg')
# Convert the image to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.imshow(gray_image, cmap='gray')
plt.title('Original Grayscale Image')
plt.axis('off')
```



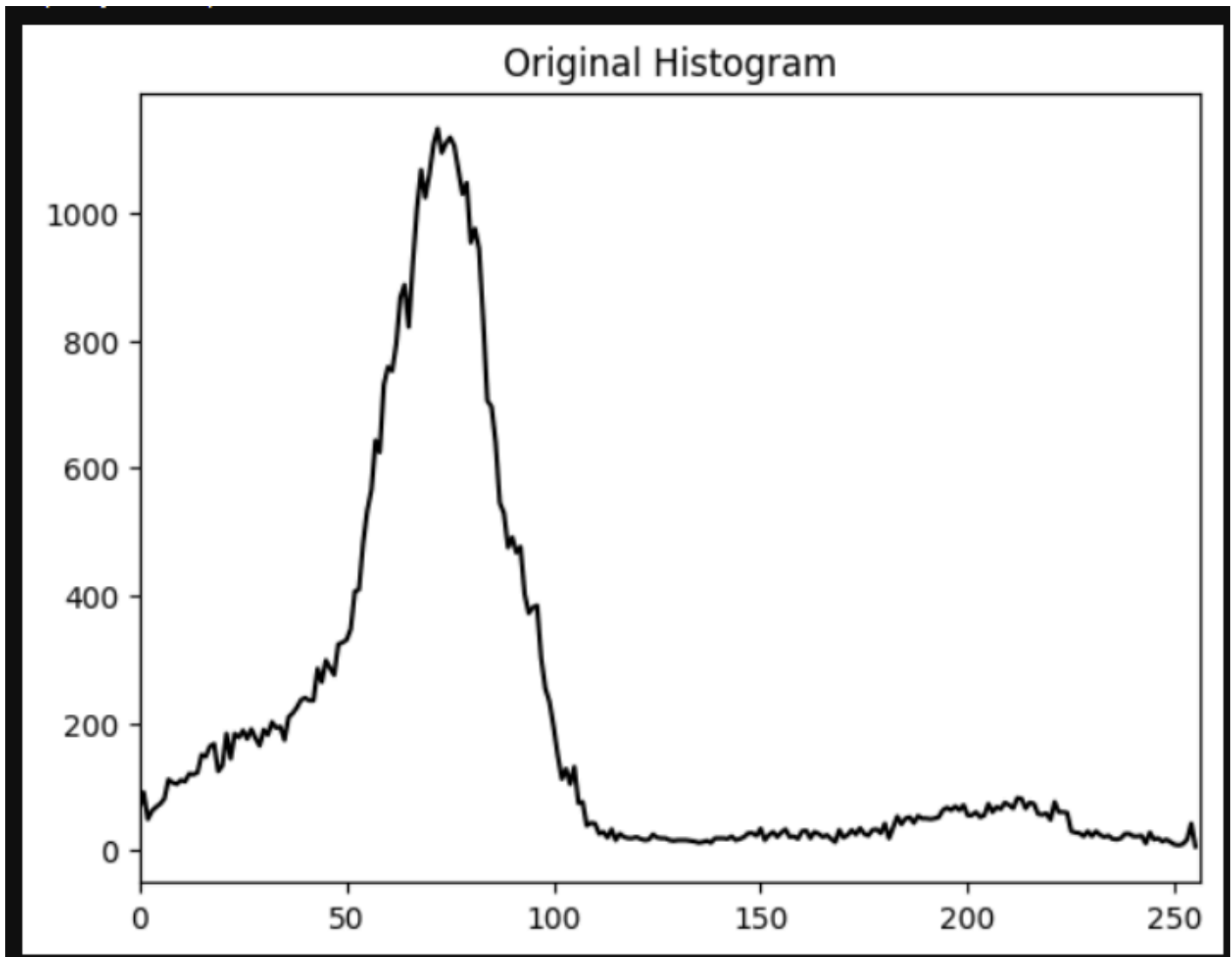
Original Grayscale Image



Histogram of Grayscale Image

```
hist_original = cv2.calcHist([gray_image], [0], None, [256], [0, 256])
plt.plot(hist_original, color='black')
plt.title('Original Histogram')
plt.xlim([0, 256])
```





Equalization

```
# Apply histogram equalization
equalized_image = cv2.equalizeHist(gray_image)
plt.imshow(equalized_image, cmap='grey')
plt.title('Equalized Image')
plt.axis('off')
```



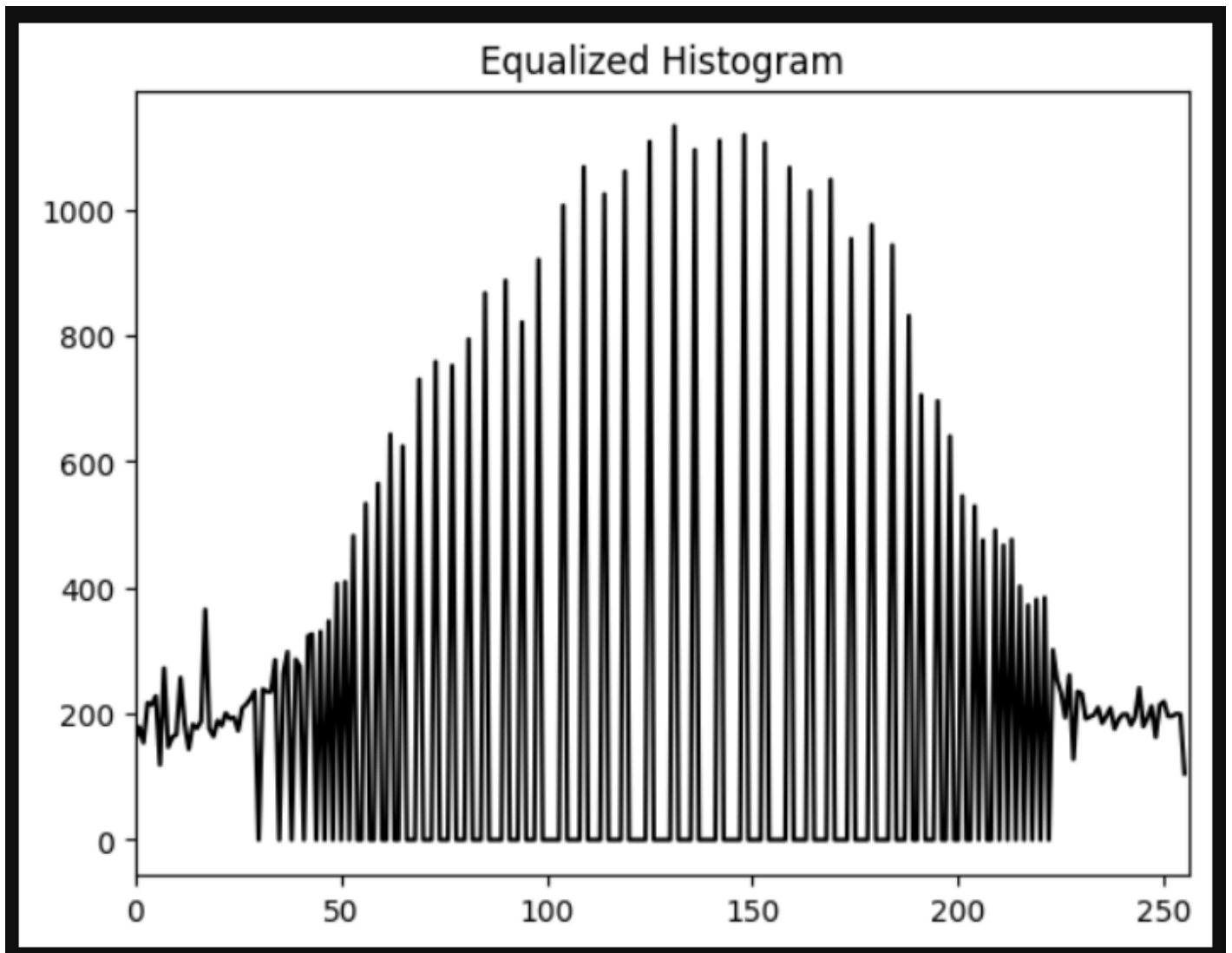
Equalized Image



Equalized Histogram

```
hist_original = cv2.calcHist([equalized_image], [0], None, [256], [0, 256])  
plt.plot(hist_original, color='black')  
plt.title('Equalized Histogram')  
plt.xlim([0, 256])
```





Result:

Thus the histogram for finding the frequency of pixels in an image with pixel values ranging from 0 to 255 is obtained. Also, histogram equalization is done for the gray scale image using OpenCV.