Academic Year 2024-25

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJS22ITL603

COURSE NAME: Image Processing and Computer Vision Laboratory **CLASS: T Y B. TECH**

NAME: Anish Sharma

Sap ID: 60003220045

EXPERIMENT NO. 4 CO/LO:

Apply Image Enhancement Techniques.

AIM / OBJECTIVE: To perform Histogram Transformations using OpenCV/Pillow Library

EXERCISE

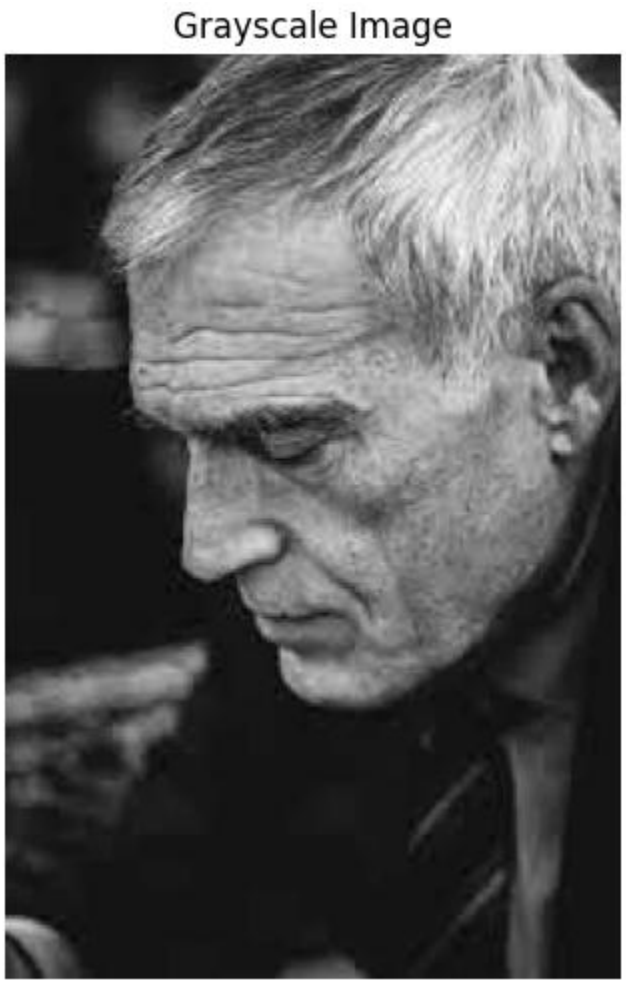
Perform histogram transformations on any Image for optimizing the image characteristics using PIL/OpenCV:

1. Read a grayscale Image
2. Display the intensity values in an Image
3. Perform Intensity transformation for image processing (vary the brightness/contrast levels in image)
4. Compare the histogram generated for normal, bright, low, and High contrast images.
5. Perform histogram equalization on an Image.
6. Apply the same to a colored image too.

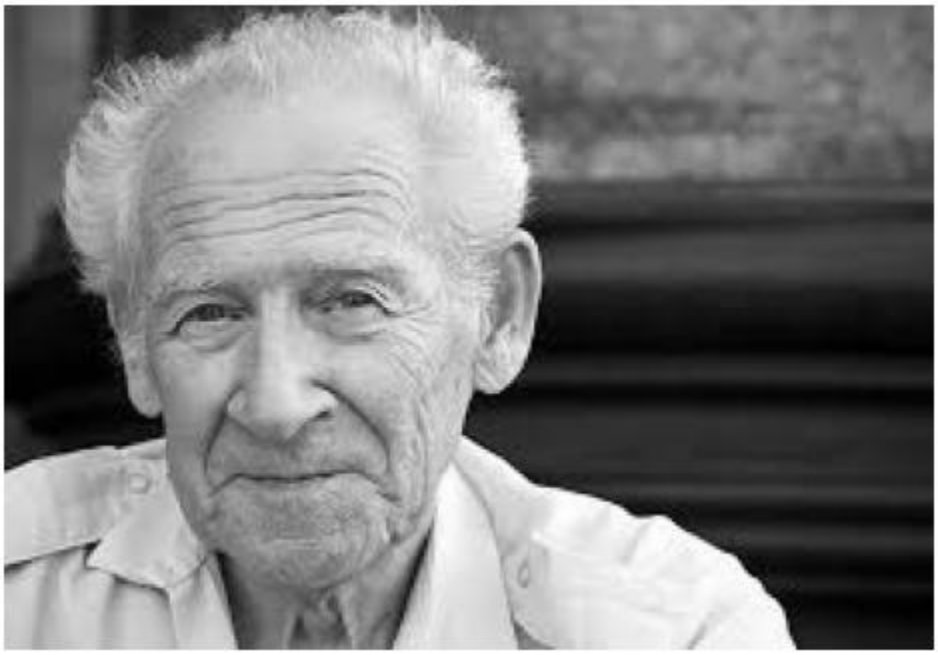
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def display_image(title, img):
    """Displays an image using matplotlib."""
    plt.figure(figsize=(6,6))
    plt.title(title)
    plt.axis('off')
    plt.imshow(img, cmap='gray')
    plt.show()
```

```
def plot_histogram(image, title):
    """Plots histogram of an image."""
    plt.figure(figsize=(6,4))
    plt.hist(image.ravel(), bins=256, range=[0,256], color='black',
alpha=0.7)
    plt.title(title)
    plt.xlabel('Pixel Intensity')
    plt.ylabel('Frequency')
    plt.show()

def adjust_brightness_contrast(image, alpha, beta):
    """Adjusts brightness and contrast of an image."""
    adjusted = cv2.convertScaleAbs(image, alpha=alpha, beta=beta)
    return adjusted

def histogram_equalization(image):
    """Performs histogram equalization."""
    return cv2.equalizeHist(image)

# Load images
gray_image = cv2.imread('/content/oldman.jpg', cv2.IMREAD_GRAYSCALE)
color_image = cv2.imread('/content/oldmancolored.jpg')
gray_from_color = cv2.cvtColor(color_image, cv2.COLOR_BGR2GRAY)

# Display original images
display_image('Grayscale Image', gray_image)
display_image('Converted Grayscale from Color', gray_from_color)

# Adjust brightness and contrast
bright_image = adjust_brightness_contrast(gray_image, 1.2, 50)
dark_image = adjust_brightness_contrast(gray_image, 0.8, -50)
high_contrast = adjust_brightness_contrast(gray_image, 2.0, 0)
low_contrast = adjust_brightness_contrast(gray_image, 0.5, 0)

display_image('Brightened Image', bright_image)
display_image('Darkened Image', dark_image)
display_image('High Contrast Image', high_contrast)
display_image('Low Contrast Image', low_contrast)

# Plot histograms
plot_histogram(gray_image, 'Histogram - Original')
plot_histogram(bright_image, 'Histogram - Bright Image')
plot_histogram(dark_image, 'Histogram - Dark Image')
plot_histogram(high_contrast, 'Histogram - High Contrast')
plot_histogram(low_contrast, 'Histogram - Low Contrast')
```

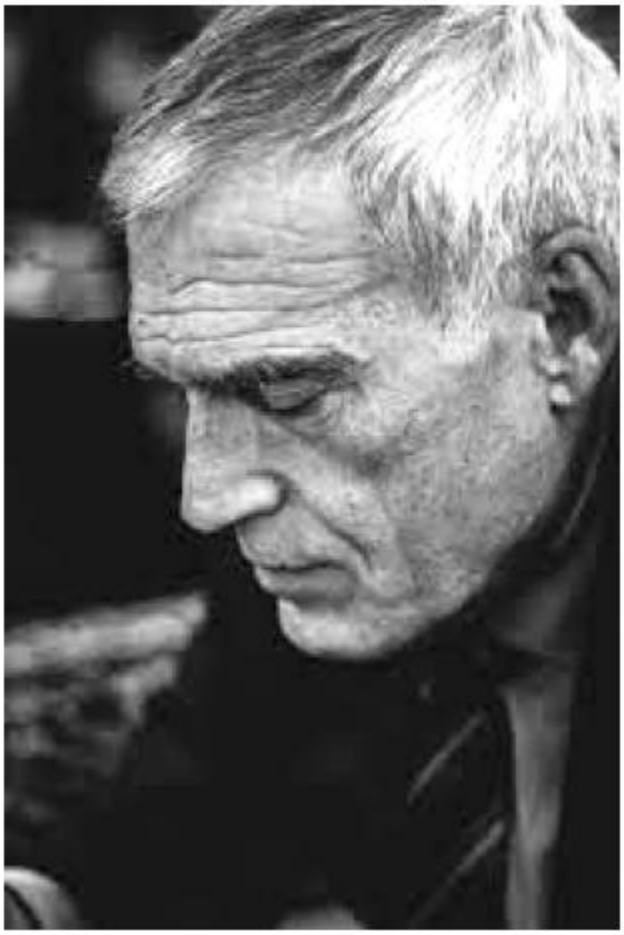
```
# Perform histogram equalization
equalized_image = histogram_equalization(gray_image)
display_image('Histogram Equalized Image', equalized_image)
plot_histogram(equalized_image, 'Histogram - Equalized Image')

# Apply histogram equalization to a colored image
color_yuv = cv2.cvtColor(color_image, cv2.COLOR_BGR2YUV)
color_yuv[:,:,0] = cv2.equalizeHist(color_yuv[:,:,0])
equalized_color = cv2.cvtColor(color_yuv, cv2.COLOR_YUV2BGR)
display_image('Histogram Equalized Color Image',
cv2.cvtColor(equalized_color, cv2.COLOR_BGR2RGB))
```



Converted Grayscale from Color



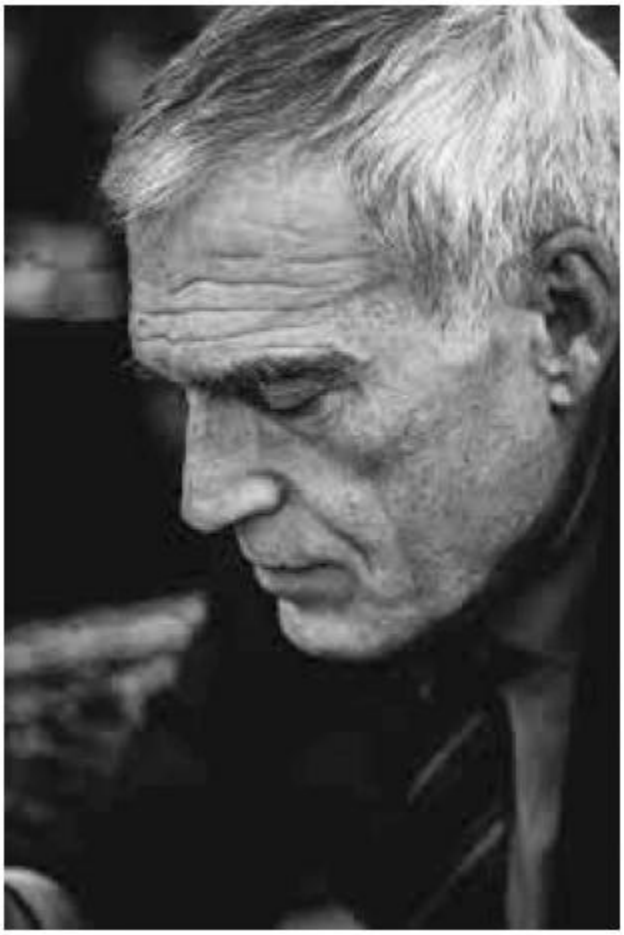
Brightened Image

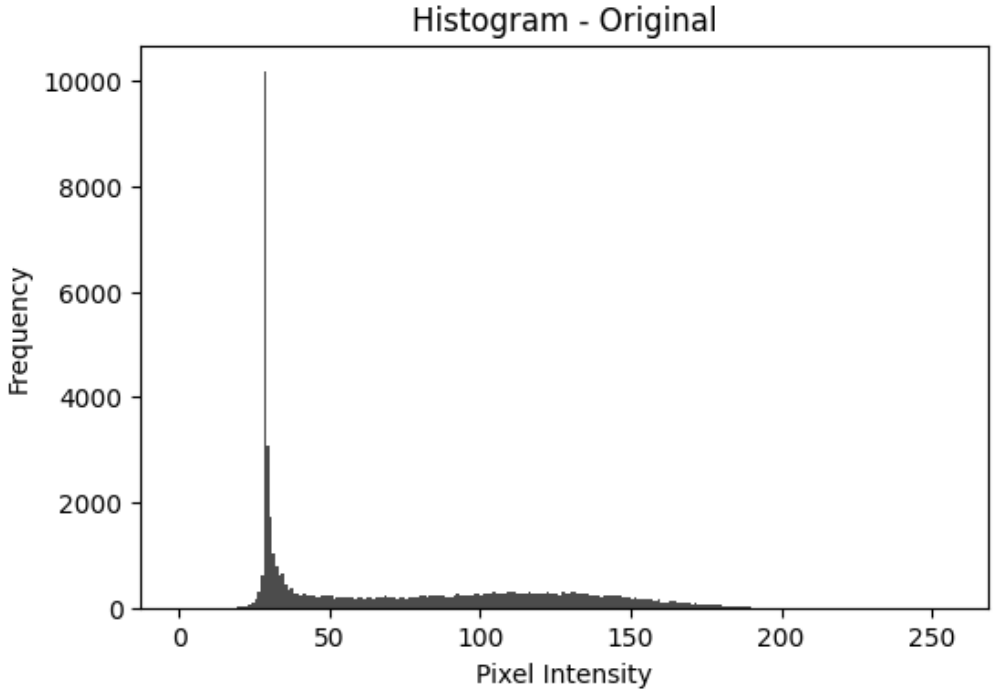


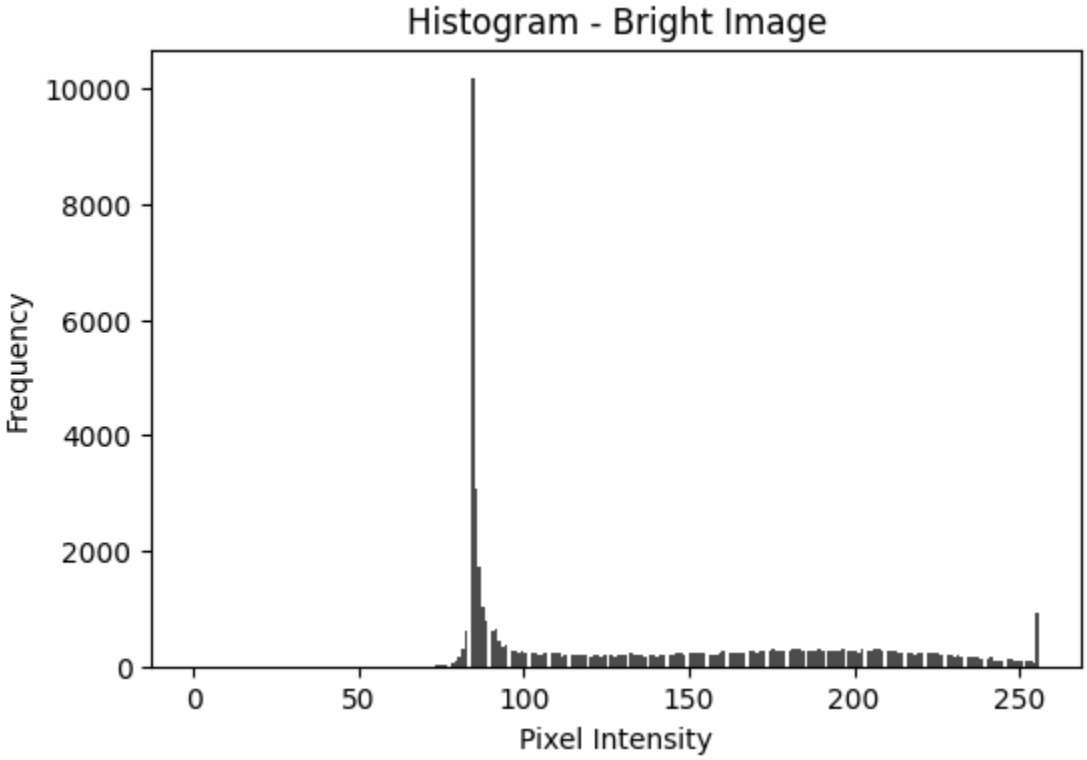
Darkened Image

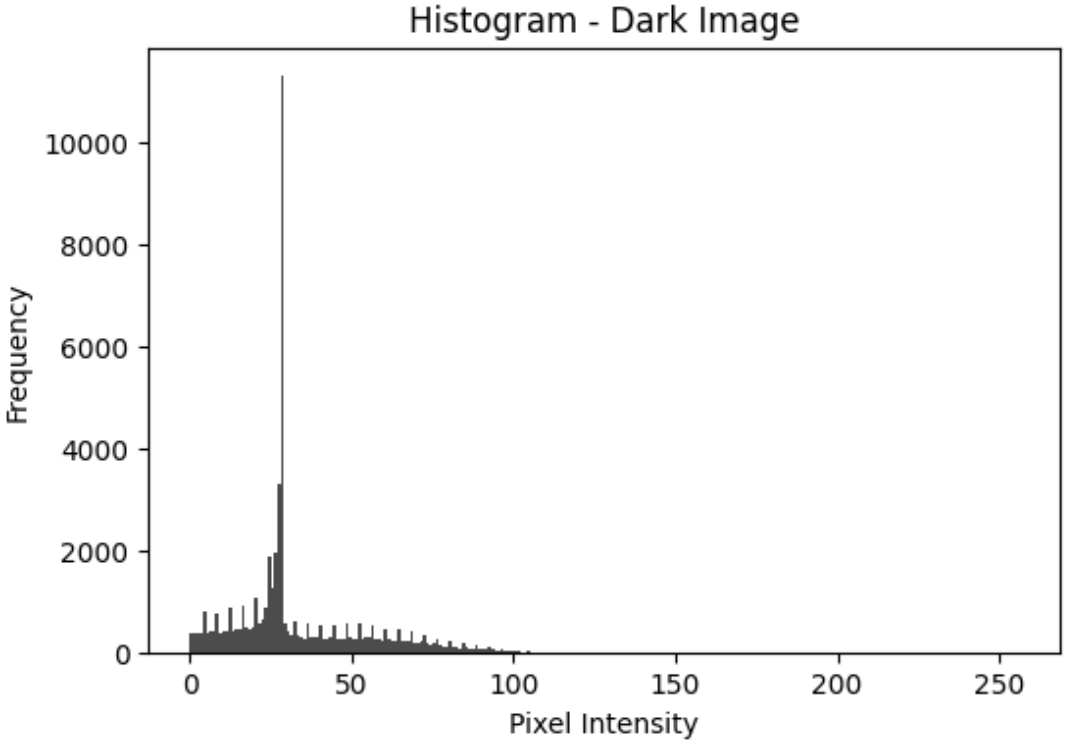
High Contrast Image

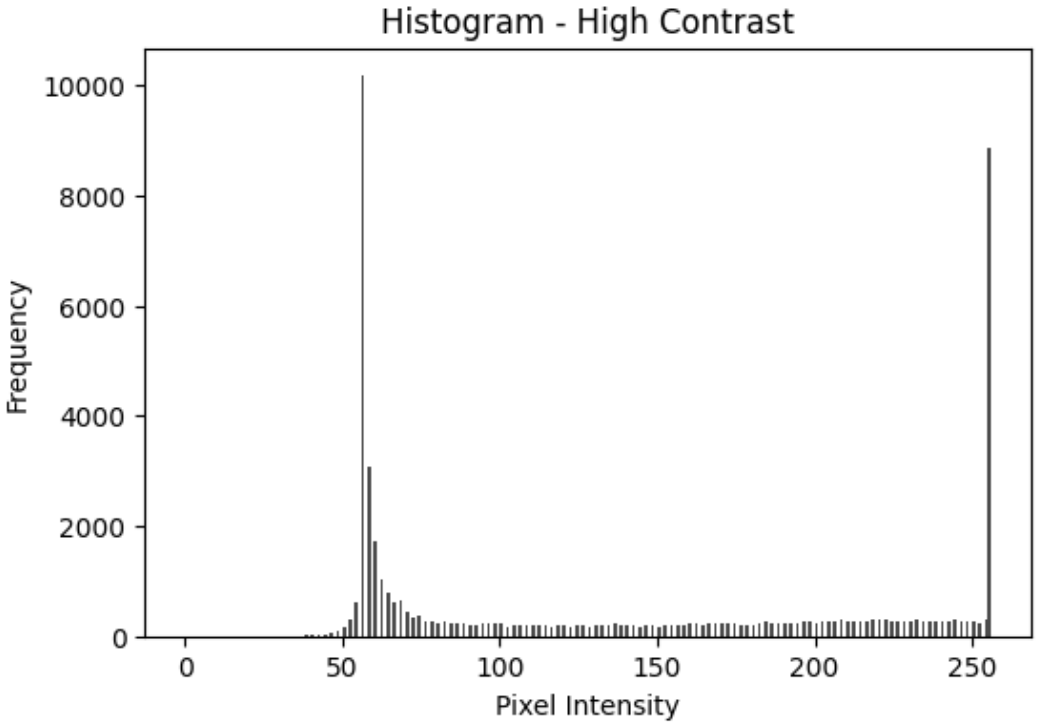
Low Contrast Image

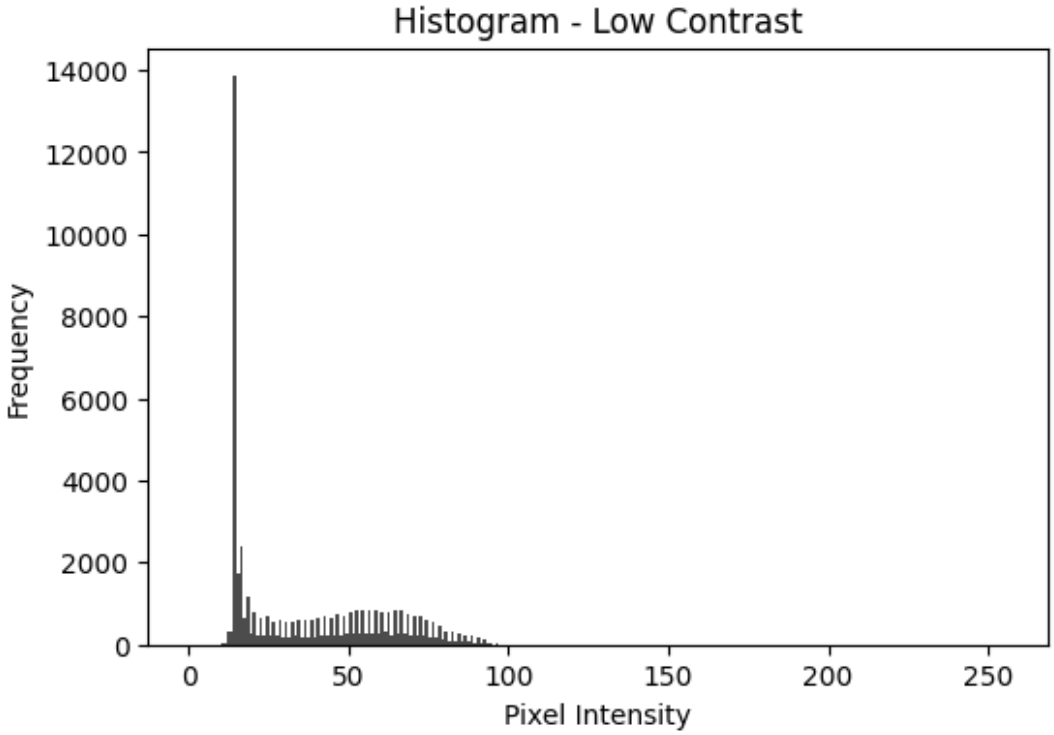




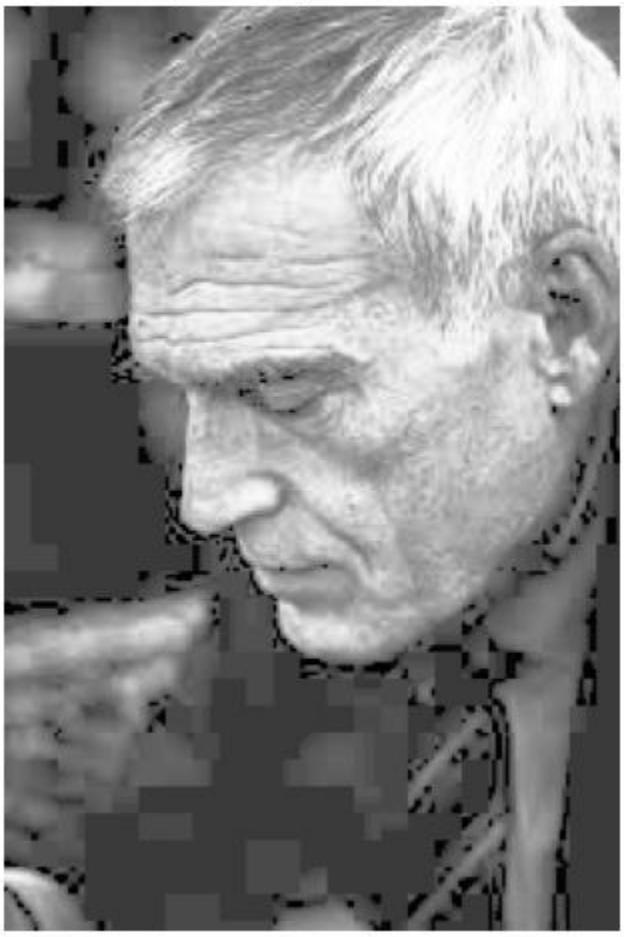

Histogram - Bright Image
Frequency
Pixel Intensity
0
2000
4000
6000
8000
10000
0
50
100
150
200
250

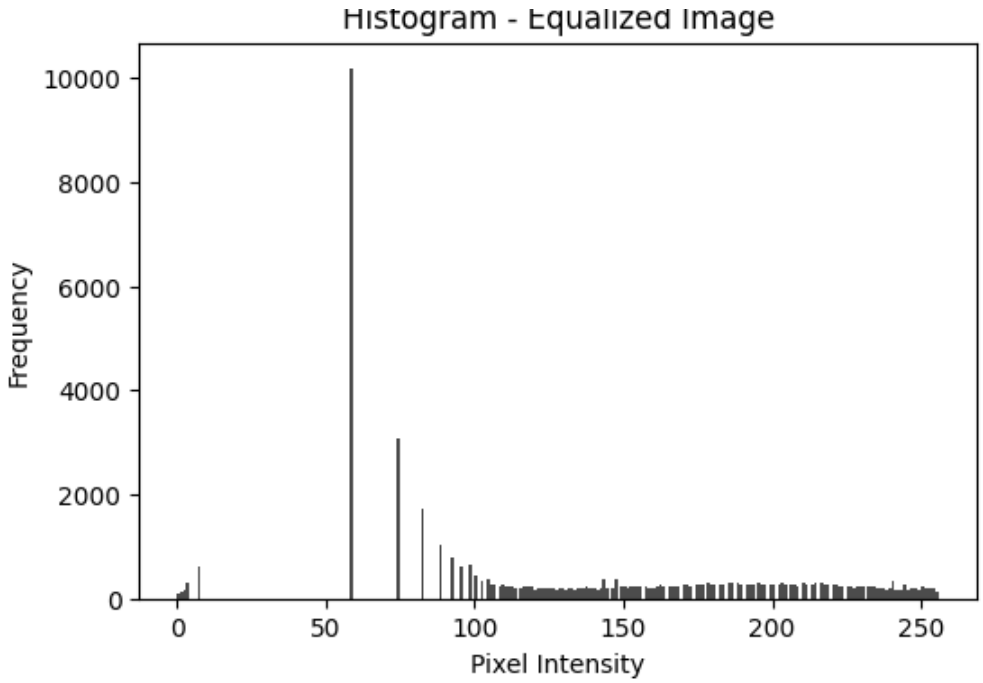

Histogram - Dark Image
Frequency
Pixel Intensity
0
2000
4000
6000
8000
10000
0
50
100
150
200
250

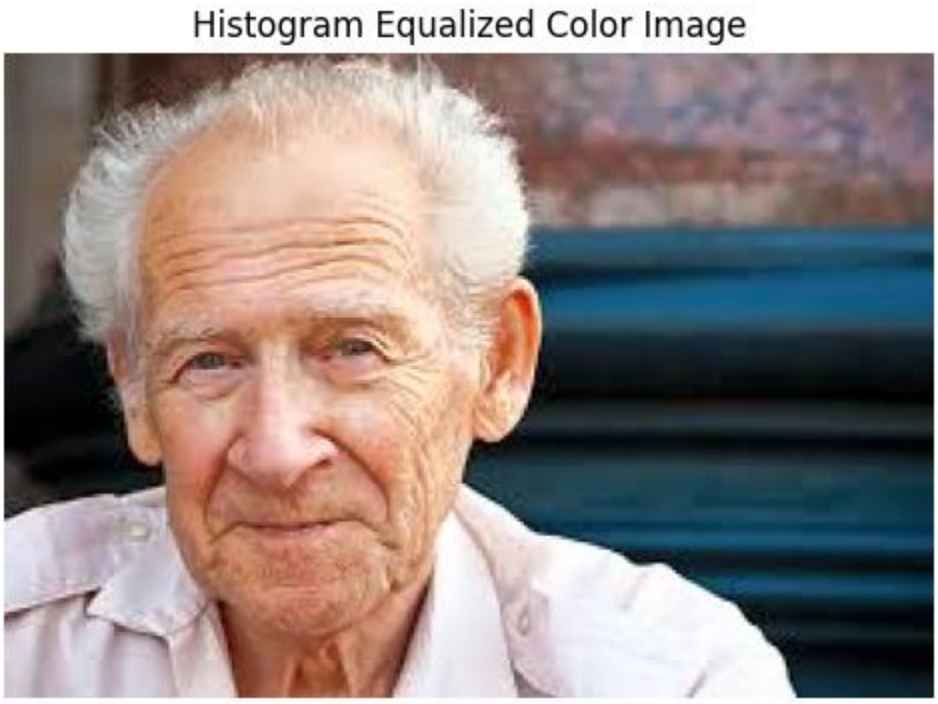

Histogram - High Contrast
Frequency
Pixel Intensity
0
2000
4000
6000
8000
10000
0
50
100
150
200
250


Histogram - Low Contrast
Frequency
Pixel Intensity
0
2000
4000
6000
8000
10000
12000
14000
0
50
100
150
200
250

Histogram Equalized Image







QUESTIONS:

- What happens if histogram equalization is applied twice?
- Describe what you can infer from the Histograms generated for different images.
- Describe the shortcomings (if any) in histogram equalization techniques.

REFERENCES:

Website References:

1. Towards Data Science, "Histogram Equalization: A Simple Way to Improve the Contrast of Your Image," *Towards Data Science*. Available: https://towardsdatascience.com/histogramequalization-a-simple-way-to-improve-the-contrast-of-your-image-bcd66596d815.
2. OpenCV, "Histogram Equalization," *OpenCV Documentation*. Available: https://docs.opencv.org/3.4/d4/d1b/tutorial_histogram_equalization.html.