In [65]:

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

In [106…

```
# Load the image using OpenCV
image_path1 = "dark.jpg"
image_path2 = "bright.jpg"
image_path3 = "lowcontrast.jpg"
image_path4 = "highcontrast.jpg"
```

In [107…

```
dark = cv.imread(image_path1)
bright = cv.imread(image_path2)
lowcontrast = cv.imread(image_path3)
highcontrast = cv.imread(image_path4)
```

In [79]:

```
# Function to plot histogram for grayscale image using OpenCV
def plot_gray_histogram_opencv(image_cv):
    plt.figure(figsize=(10, 5))
    plt.hist(image_cv.ravel(), bins=256, color='gray', alpha=0.7)
    plt.title('Grayscale Histogram')
    plt.xlabel('Pixel Intensity')
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.show()
```

In [80]:

```
# Function to plot histograms for each color channel using OpenCV
def plot_color_histogram_opencv(image_cv):
    colors = ('blue', 'green', 'red')
    plt.figure(figsize=(10, 5))
    for i, color in enumerate(colors):
        plt.hist(image_cv[:, :, i].ravel(), bins=256, color=color, alpha=0.7)
    plt.title('Color Histogram')
    plt.xlabel('Pixel Intensity')
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.show()
```

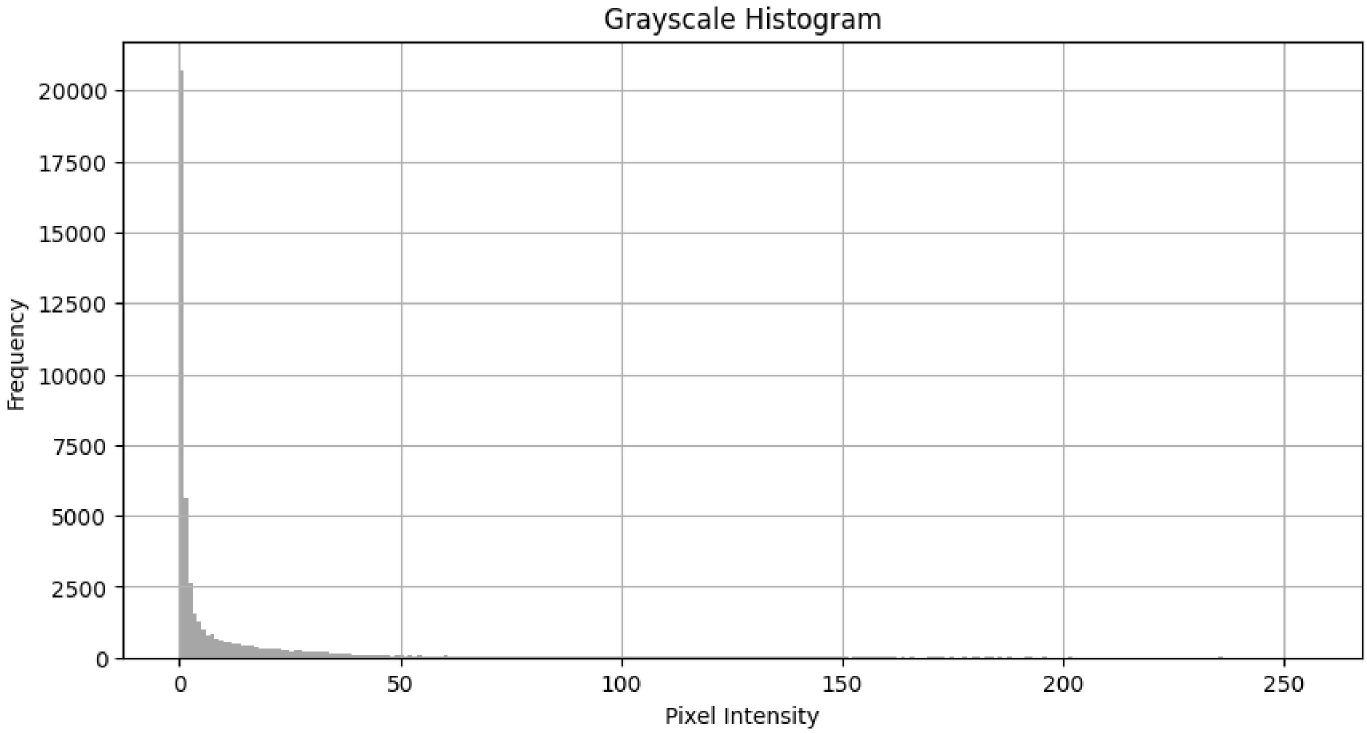
In [81]:

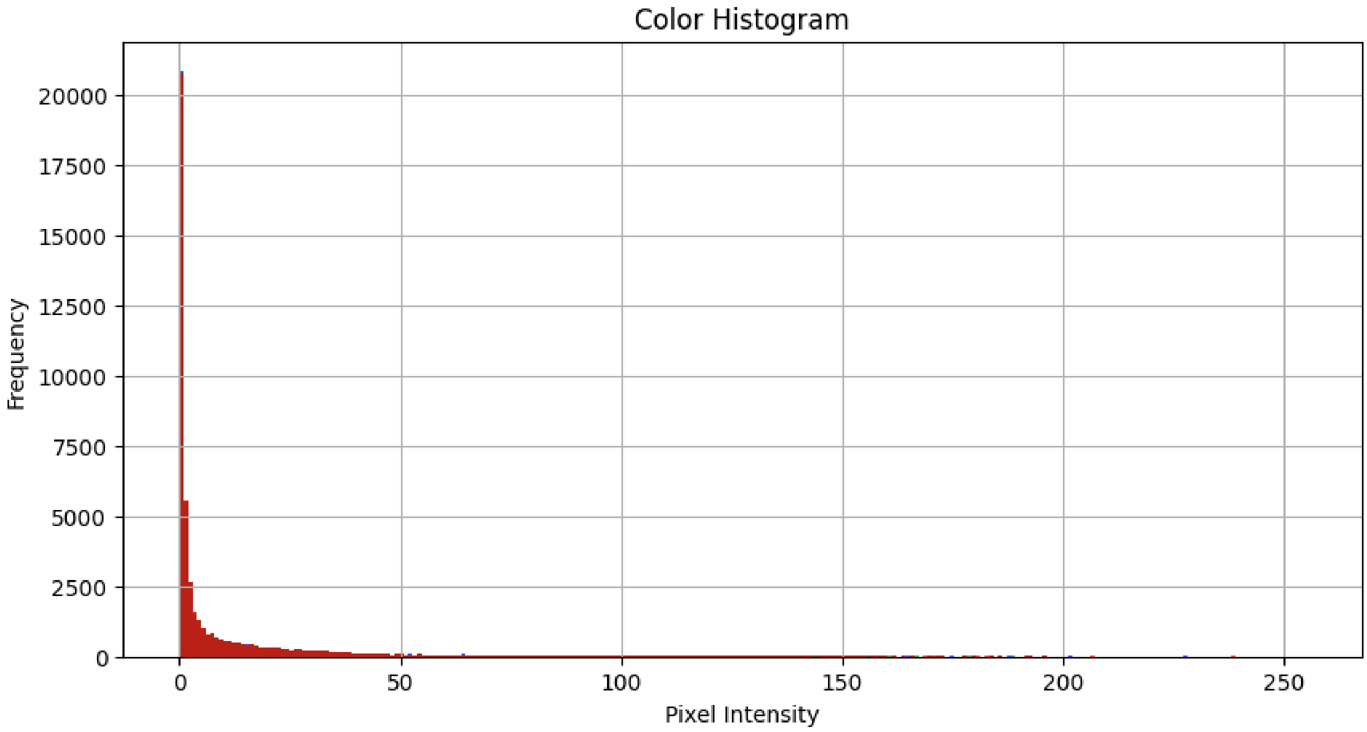
```
# Function to plot histograms for each individual color channel using OpenCV
def plot_individual_color_histogram(image_cv):
    colors = ('blue', 'green', 'red')
    plt.figure(figsize=(15, 5))
    for i, color in enumerate(colors):
        plt.subplot(1, 3, i+1)
        plt.hist(image_cv[:, :, i].ravel(), bins=256, color=color, alpha=0.7)
        plt.title(f'{color.capitalize()} Channel Histogram')
        plt.xlabel('Pixel Intensity')
        plt.ylabel('Frequency')
        plt.grid(True)
    plt.tight_layout()
    plt.show()
```

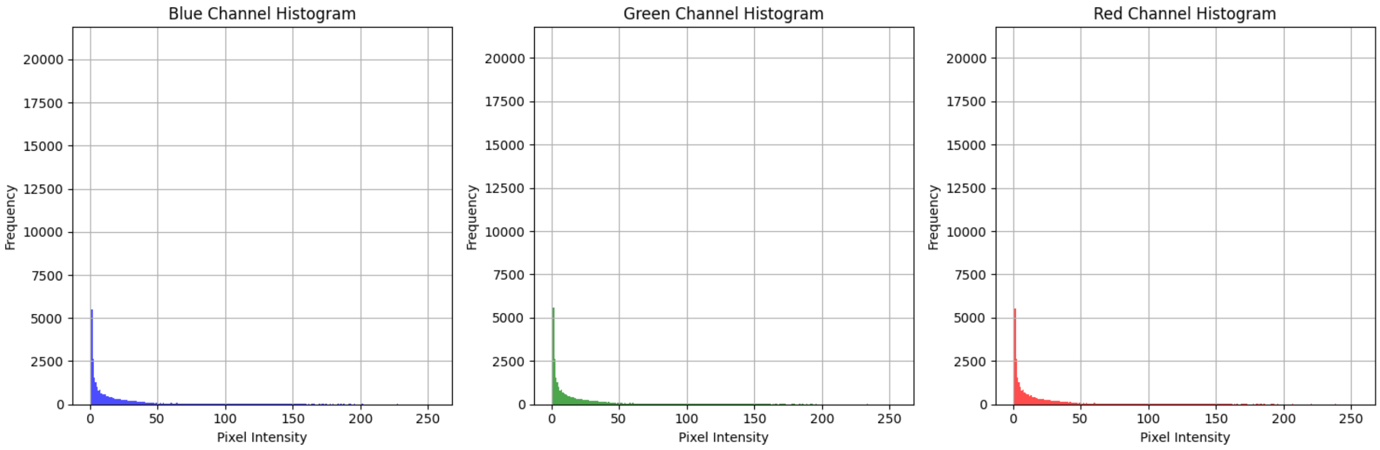
In [82]:

```
print("DARK IMAGE")
gray_dark = cv.cvtColor(dark, cv.COLOR_BGR2GRAY)
plot_gray_histogram_opencv(gray_dark)
plot_color_histogram_opencv(dark)
plot_individual_color_histogram(dark)
```

```
DARK IMAGE
```





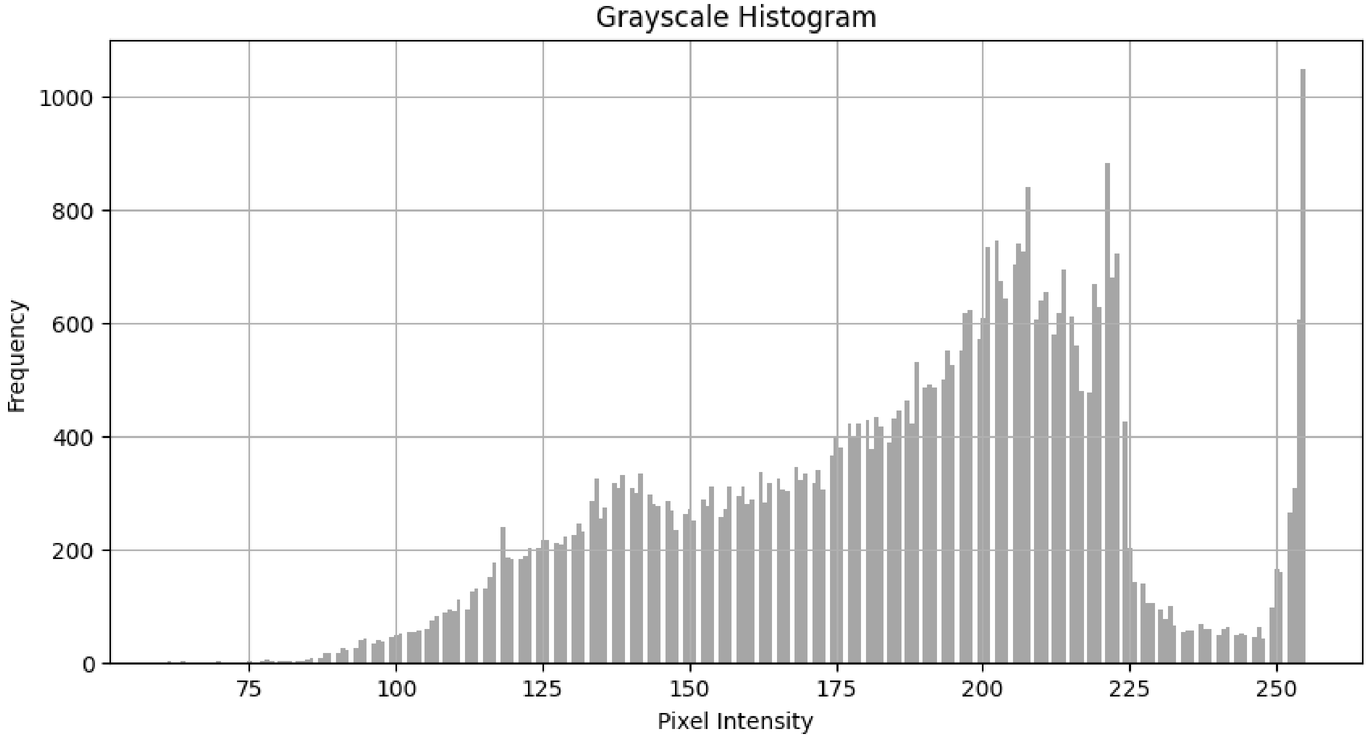


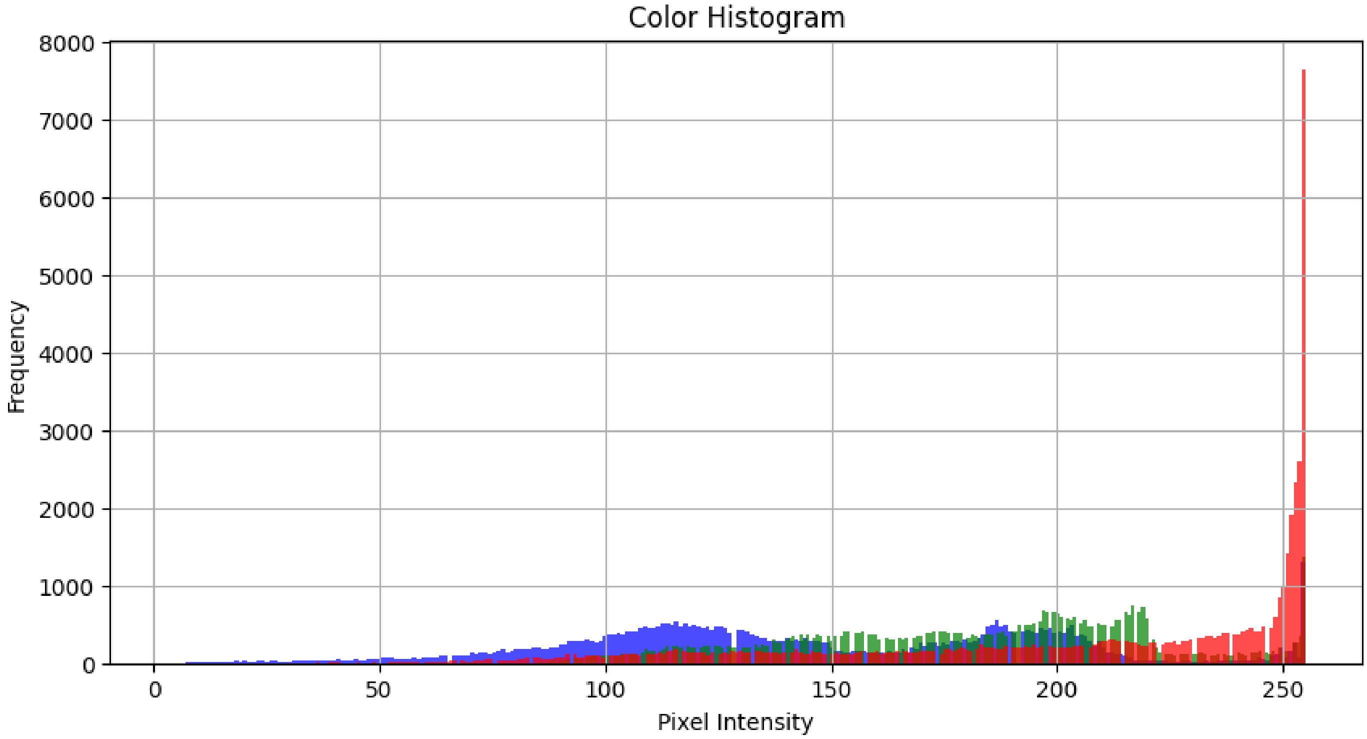
In [108…

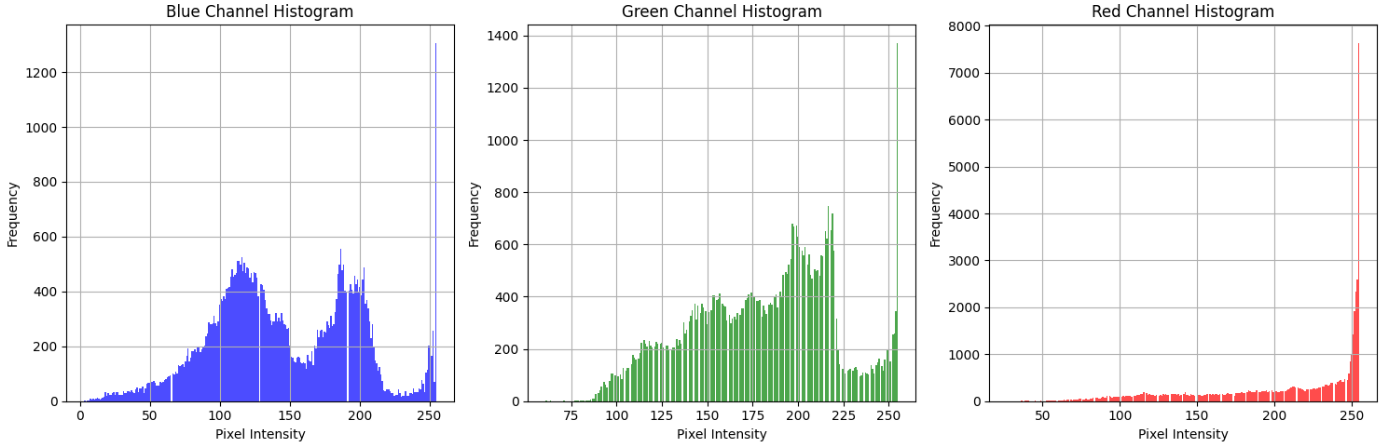
```
print("BRIGHT")
gray_bright = cv.cvtColor(bright, cv.COLOR_BGR2GRAY)
plot_gray_histogram_opencv(gray_bright)
```

```
plot_color_histogram_opencv(bright)
plot_individual_color_histogram(bright)
```

```
BRIGHT
```



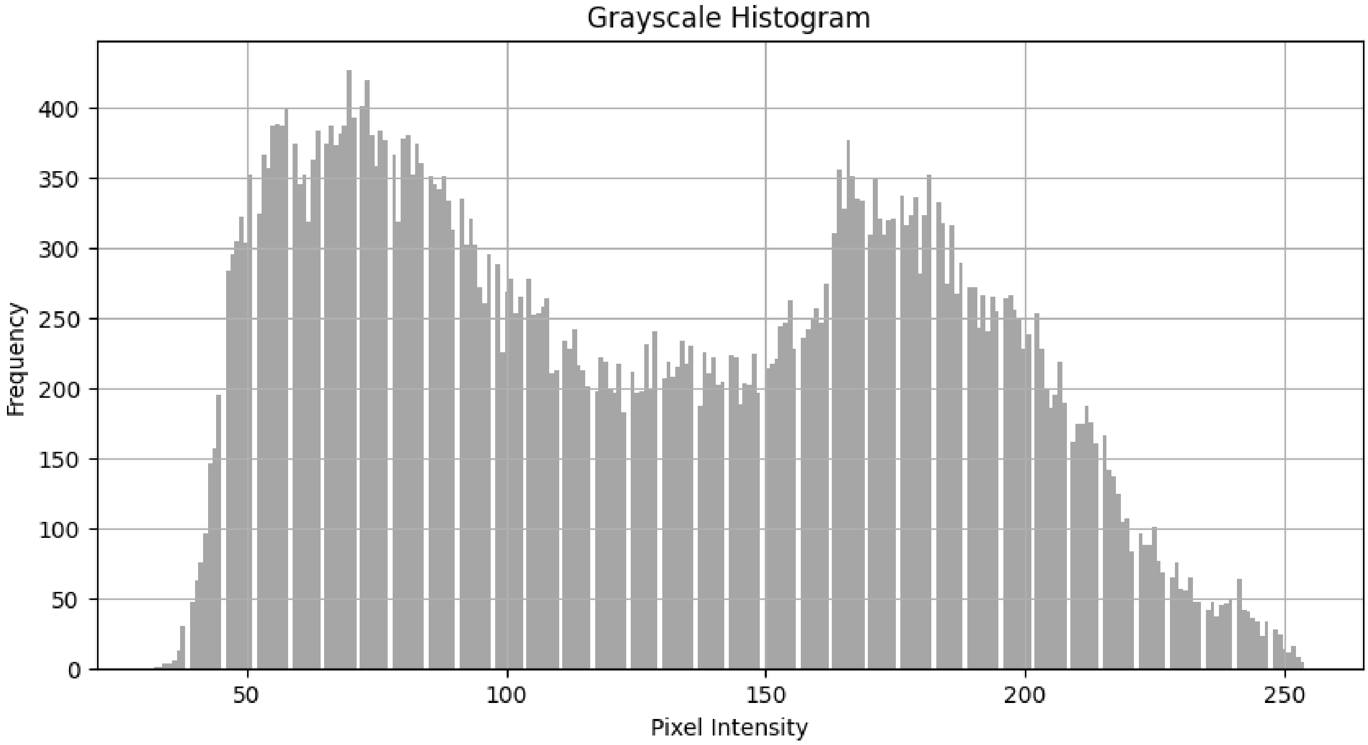


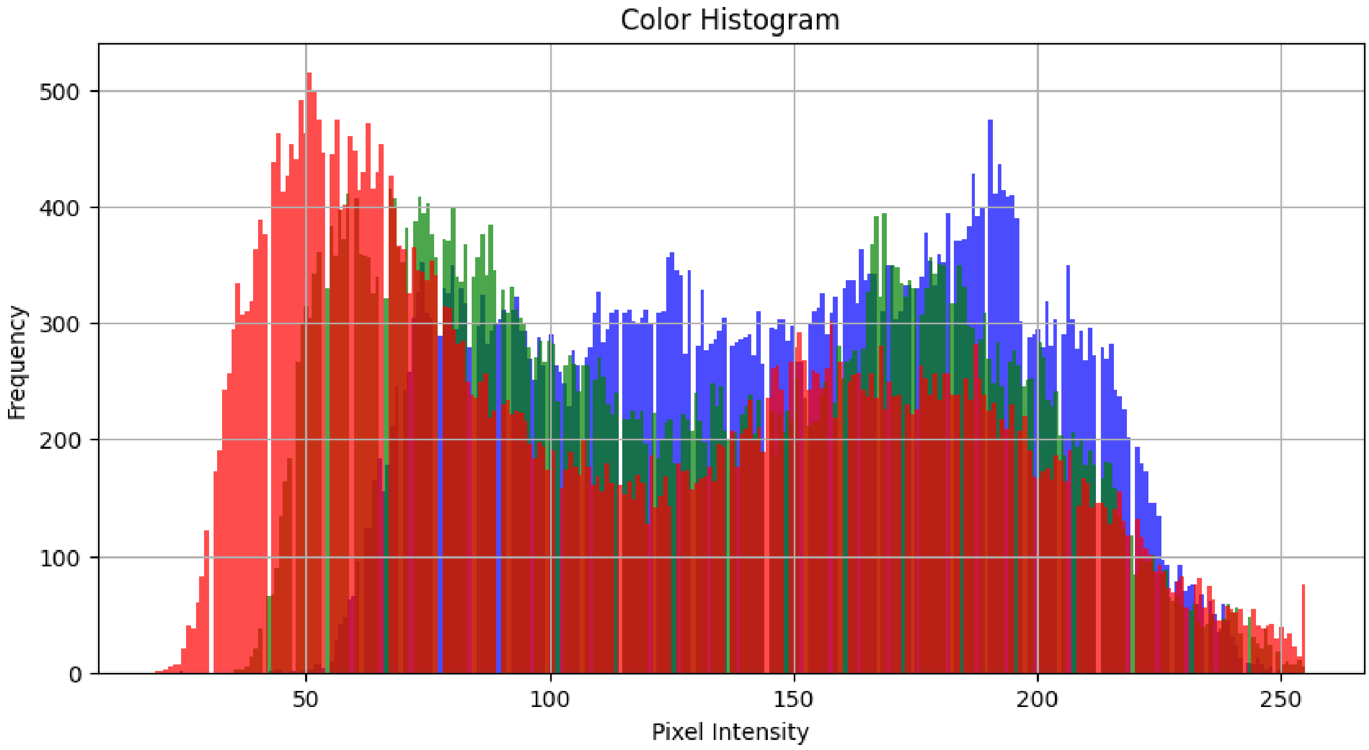


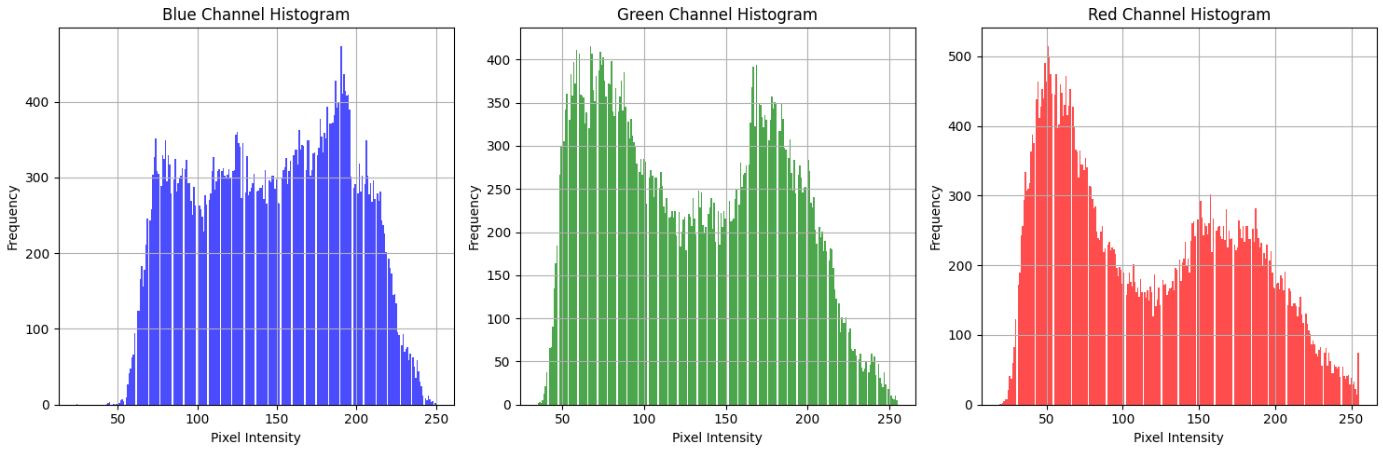
In [84]:

```
print("LOW CONTRAST IMAGE")
gray_lowcontrast = cv.cvtColor(lowcontrast, cv.COLOR_BGR2GRAY)
plot_gray_histogram_opencv(gray_lowcontrast)
plot_color_histogram_opencv(lowcontrast)
plot_individual_color_histogram(lowcontrast)
```

```
LOW CONTRAST IMAGE
```



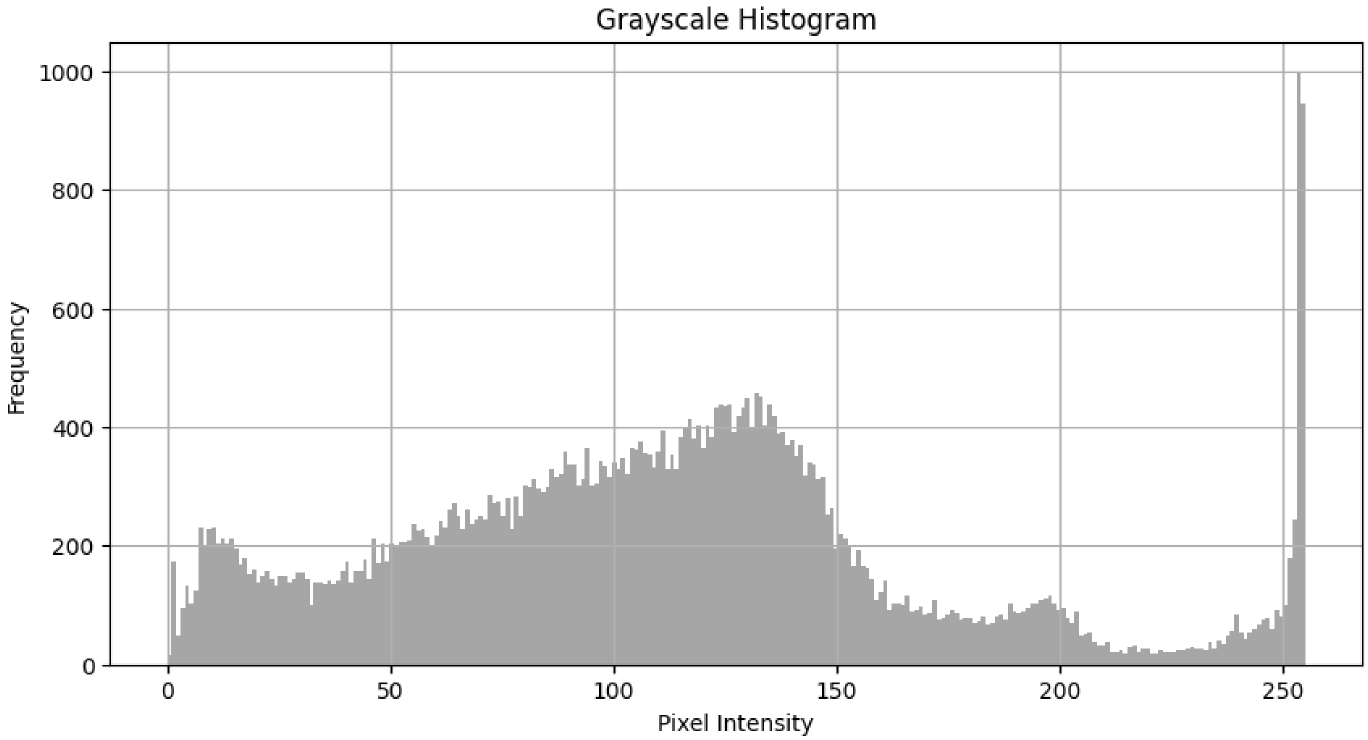


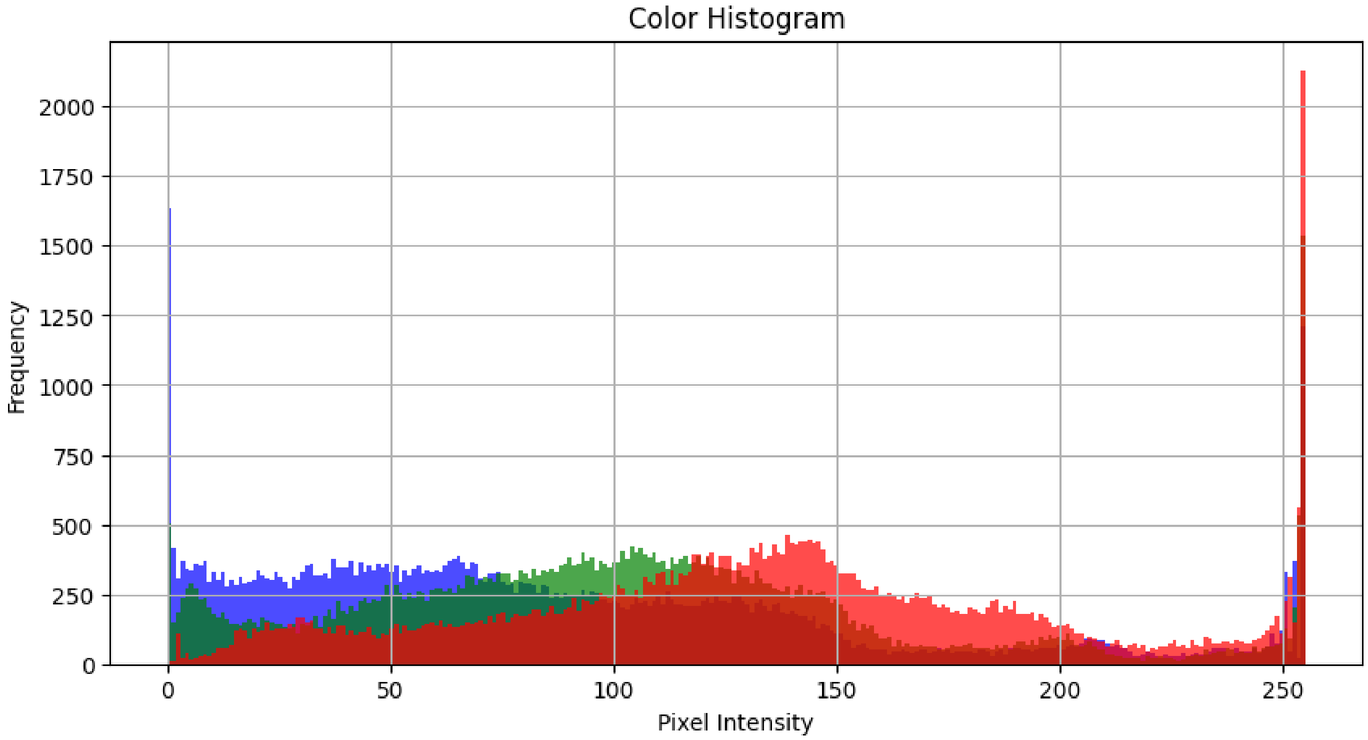


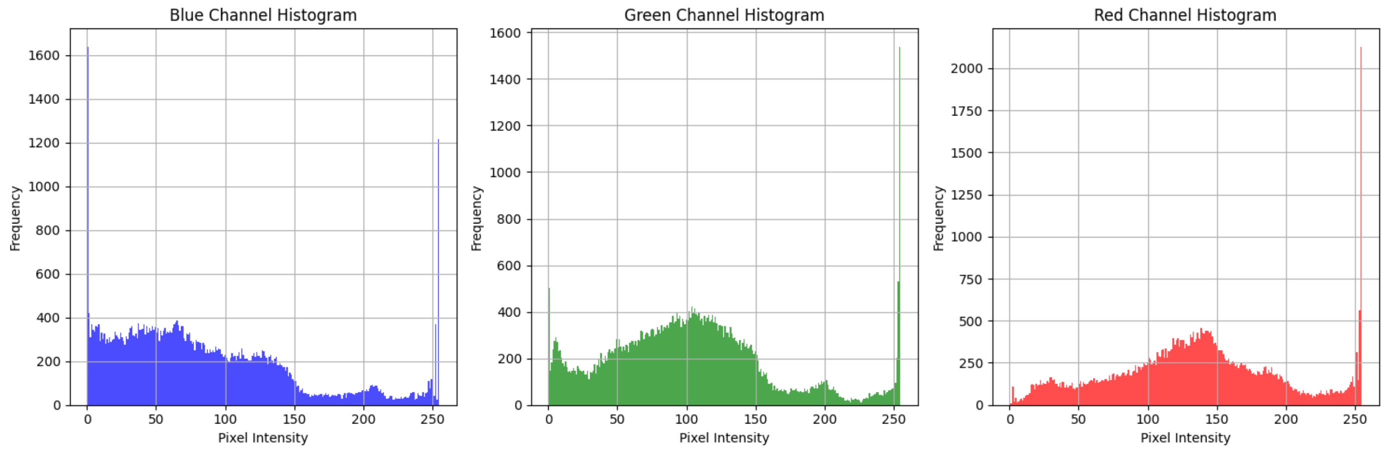
In [85]:

```
print("HIGH CONTRAST IMAGE")
gray_highcontrast = cv.cvtColor(highcontrast, cv.COLOR_BGR2GRAY)
plot_gray_histogram_opencv(gray_highcontrast)
plot_color_histogram_opencv(highcontrast)
plot_individual_color_histogram(highcontrast)
```

```
HIGH CONTRAST IMAGE
```



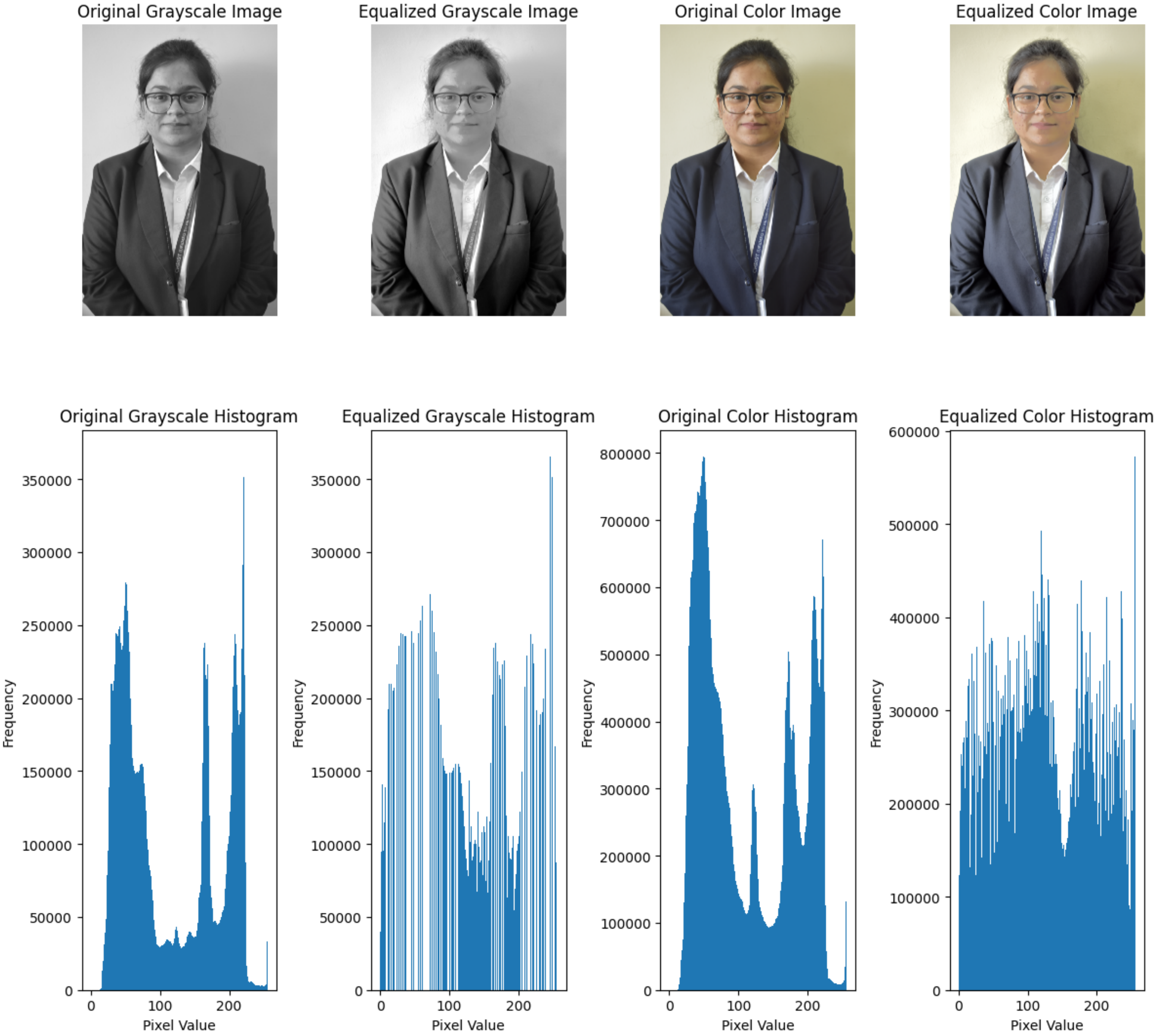




In [119…

```
import cv2
import numpy as np
```

```python
import matplotlib.pyplot as plt

# Load the image
image = cv2.imread("myphoto.JPG")

# Convert to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Apply histogram equalization to grayscale image
equalized_gray_image = cv2.equalizeHist(gray_image)

# Apply histogram equalization to color image
# Convert to YUV color space
yuv_image = cv2.cvtColor(image, cv2.COLOR_BGR2YUV)
# Equalize the histogram of the Y channel
yuv_image[:, :, 0] = cv2.equalizeHist(yuv_image[:, :, 0])
# Convert back to BGR color space
equalized_color_image = cv2.cvtColor(yuv_image, cv2.COLOR_YUV2BGR)

# Function to plot images and histograms
def plot_image_histogram(image, title, subplot):
    plt.subplot(subplot)
    plt.title(title)
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
    plt.axis('off')

def plot_histogram(image, title, subplot):
    plt.subplot(subplot)
    plt.title(title)
    plt.hist(image.ravel(), 256, [0, 256])
    plt.xlabel('Pixel Value')
    plt.ylabel('Frequency')

# Plot the original and equalized images with histograms
plt.figure(figsize=(12, 12))

# Original Grayscale Image and Histogram
plot_image_histogram(gray_image, 'Original Grayscale Image', 241)
plot_histogram(gray_image, 'Original Grayscale Histogram', 245)

# Equalized Grayscale Image and Histogram
plot_image_histogram(equalized_gray_image, 'Equalized Grayscale Image', 242)
plot_histogram(equalized_gray_image, 'Equalized Grayscale Histogram', 246)

# Original Color Image and Histogram
plot_image_histogram(image, 'Original Color Image', 243)
plot_histogram(image, 'Original Color Histogram', 247)

# Equalized Color Image and Histogram
plot_image_histogram(equalized_color_image, 'Equalized Color Image', 244)
plot_histogram(equalized_color_image, 'Equalized Color Histogram', 248)

plt.tight_layout()
plt.show()
```


Original Grayscale Image
Equalized Grayscale Image
Original Color Image
Equalized Color Image
Original Grayscale Histogram
Equalized Grayscale Histogram
Original Color Histogram
Equalized Color Histogram
Frequency
Pixel Value
0
100
200
0
50000
100000
150000
200000
250000
300000
350000
400000
500000
600000
700000
800000