Image Transformations

Brightness & Contrast (Linear)

As you mentioned in Module 1, you can think of a (grayscale) image as a function:

$$f : \mathbb{R}^2 \to \mathbb{R}$$

where

$$f(x, y)$$

gives the intensity at position

$$(x, y)$$

So you can transform the image by adjusted the function $f(x, y)$

A common approach is to adjust the intensity levels of the image using a linear scaling function:

$$\text{output} = \alpha \times \text{input} + \beta$$

Where:

- α is the gain or contrast control. Values > 1 increase contrast, and values between 0 and 1 decrease contrast.
- β is the bias or brightness control.

Brightness: This is the adjustment of the lightness or darkness of an image. An image's brightness can be increased by adding a constant to every pixel value of the image.

Contrast: This represents the difference between the darkest and lightest regions of an image. The contrast can be increased by multiplying every pixel value of the image by a constant.

First, let's translate the brightness and contrast formula to code.

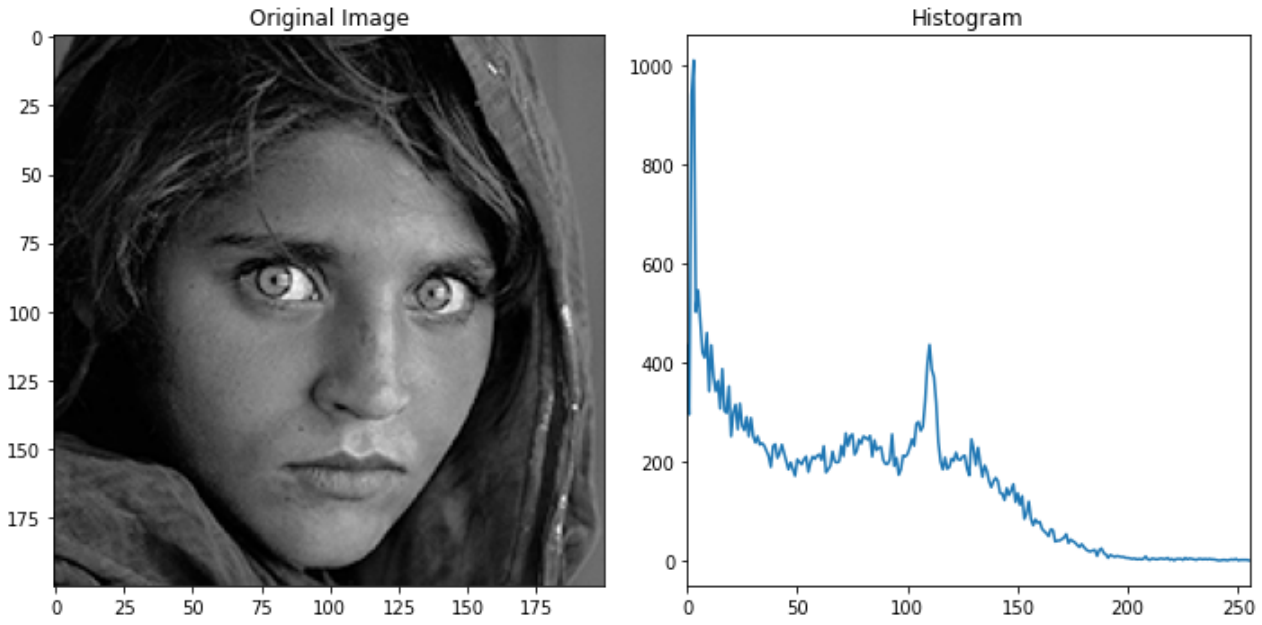
In [ ]:

```
# output = alpha * input + beta

def adjust_brightness_contrast(image_array, alpha, beta):
    # Convert to floating point to avoid overflow and underflow
    temp = image_array.astype('float32')

    # Adjust contrast
    temp *= alpha

    # Adjust brightness
    temp += beta

    # Clip values to be in the range [0, 255]
    temp = np.clip(temp, 0, 255)

    return temp.astype('uint8')
```

Next, let's import an image, convert it's colorspace to grayscale, and then plot it's histogram to analyze.

In [ ]:

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

# Load Image. 1 for color. 0 for black and white.  Let's import in color and con
img = cv2.imread("Graphics/face.png")

# Convert image_data to grayscale
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

hist = cv2.calcHist([img],[0],None,[256],[0,255])

# Plotting original and adjusted images side-by-side. plt.subplot(nrows, ncols,

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

plt.subplot(1, 2, 1)
plt.imshow(img, cmap="gray")
plt.title('Original Image')

plt.subplot(1, 2, 2)
plt.plot(hist)
plt.xlim([0,255])
plt.title('Histogram')

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



Now let's send out image to our brightness and contrast function.

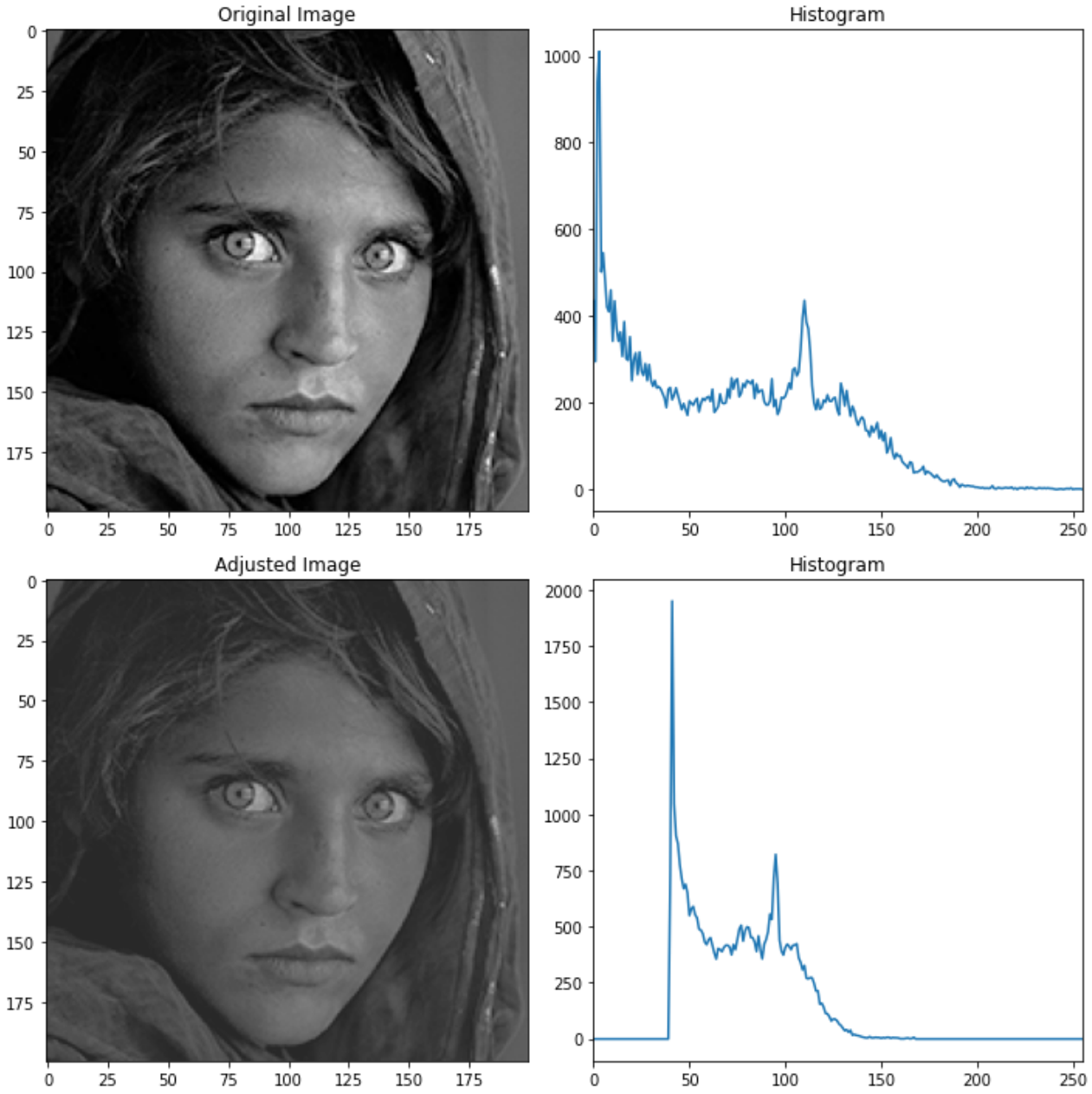
In [ ]:

```
adjusted_img = adjust_brightness_contrast(img, .5, 40)
```

Let's plot our adjusted image and the orignal image with histograms.

In [ ]:

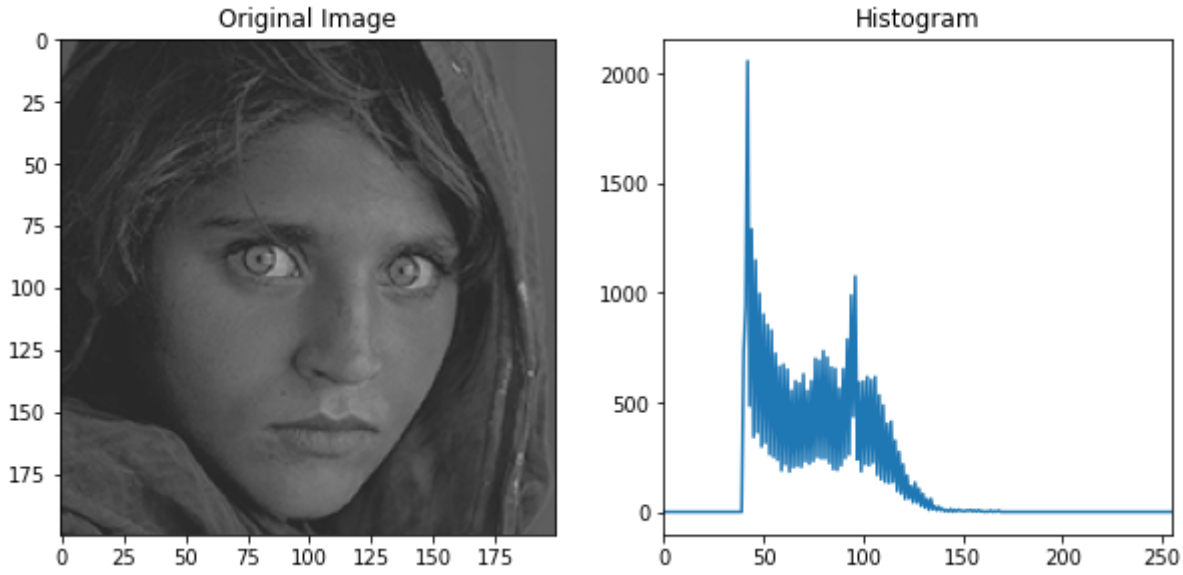
```
# Calculate histogram of adjusted image
adjusted_hist = cv2.calcHist([adjusted_img],[0],None,[256],[0,255])

# Plotting original and adjusted images side-by-side
plt.figure(figsize=(10,10))

plt.subplot(2, 2, 1)
plt.imshow(img, cmap="gray")
plt.title('Original Image')

plt.subplot(2, 2, 2)
plt.plot(hist)
plt.xlim([0,255])
plt.title('Histogram')

plt.subplot(2, 2, 3)
plt.imshow(adjusted_img, cmap="gray", vmin=0, vmax=255)
plt.title('Adjusted Image')

plt.subplot(2, 2, 4)
plt.plot(adjusted_hist)
plt.xlim([0,255])
plt.title('Histogram')

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



In [ ]:

```
# Use OpenCV to adjust brightness and contrast
openCV_adjusted_img = cv2.convertScaleAbs(img, alpha=.5, beta=40)

# Calculate histogram of adjusted image
openCV_adjusted_hist = cv2.calcHist([openCV_adjusted_img],[0],None,[256],[0,255]

# Plotting original and adjusted images side-by-side
plt.figure(figsize=(10,10))

plt.subplot(2, 2, 1)
plt.imshow(openCV_adjusted_img, cmap="gray", vmin=0, vmax=255)
plt.title('Original Image')

plt.subplot(2, 2, 2)
plt.plot(openCV_adjusted_hist)
plt.xlim([0,255])
plt.title('Histogram')
```

Out[ ]: Text(0.5, 1.0, 'Histogram')



Gamma Correction (Non-Linear)

Gamma correction is used to adjust the brightness and contrast of digital images. It's a nonlinear operation that encodes and decodes luminance values in an image.

The formula for gamma correction is:

$$O = I^{\gamma}$$

Where:

- O is the output pixel value after gamma correction.
- I is the normalized input pixel value, usually in the range [0, 1].
- γ is the gamma value.

Considering pixel values are typically in the range [0, 255], the formula can be expanded to:

$$O = 255 \times \left(\frac{I}{255}\right)^{\gamma}$$

- If $\gamma < 1$, the image is made brighter.
- If $\gamma > 1$, the image is made darker.

In [ ]:

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

def gamma_correction(img, gamma):
    # Normalize the image to [0, 1], apply gamma correction, and then scale back
    corrected = 255.0 * (img / 255.0) ** gamma
    # Return the corrected image
    return np.uint8(corrected)

# Load the image
```

```
image = cv2.imread("Graphics/face.png", cv2.IMREAD_GRAYSCALE)

# Display the original image
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.imshow(image, cmap="gray")
plt.title("Original Image")

# Apply gamma correction and display the result
gamma = 0.5
corrected_image = gamma_correction(image, gamma)

plt.subplot(1, 2, 2)
plt.imshow(corrected_image, cmap="gray")
plt.title(f"Gamma Corrected Image (gamma={gamma})")
plt.tight_layout()
plt.show()
```