

# ICE 472: Digital Speech & Image Processing

Summer 2020

## Experiment 4: The Histogram, Cumulative Distribution Function(CDF) & Histogram Matching.

**Name:** Suraya Ahlam

**Student ID:** 2016-1-50-014

### Assignment:

(1) Running this notebook and producing the outputs.

#### ▼ Histograms

#### ▼ Library Imports

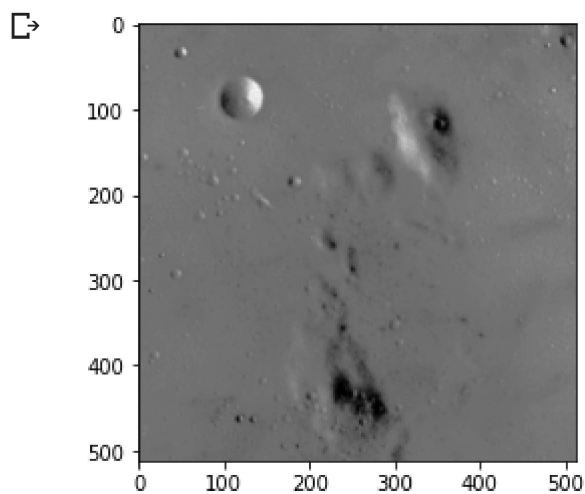
```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np

from skimage import data, img_as_float
from skimage import exposure
from skimage.exposure import match_histograms
```

#### ▼ Loading The Image

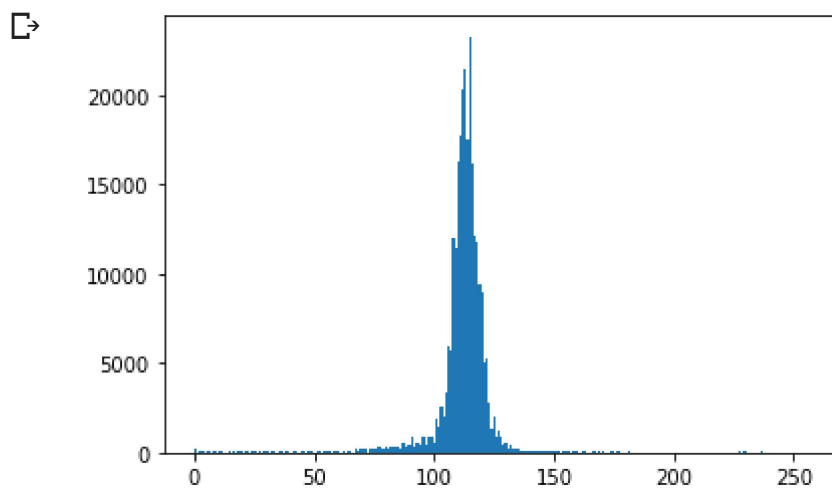
```
img = data.moon()
bins = 256
```

```
plt.imshow(img, cmap = 'gray')  
plt.show()
```

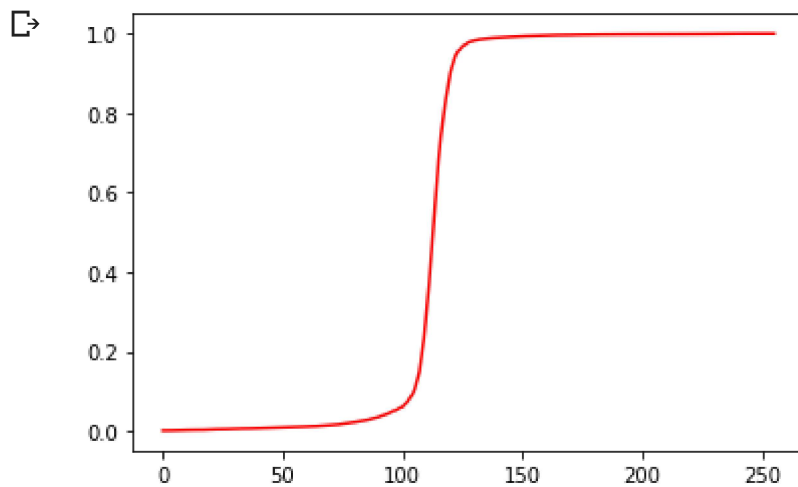


## ▼ Plotting The Histogram & CDF of the image:

```
plt.hist(img.ravel(), bins = bins)  
plt.show()
```

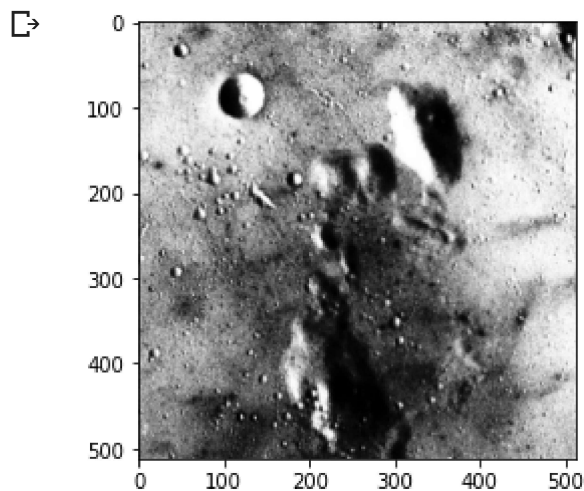


```
img_cdf, bins = exposure.cumulative_distribution(img, bins)
plt.plot(bins, img_cdf, 'r')
plt.show()
```

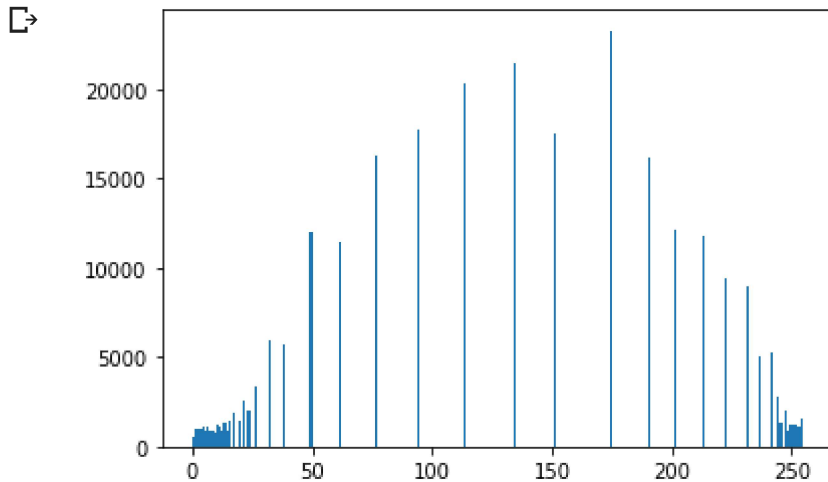


## ▼ Histogram Equalization:

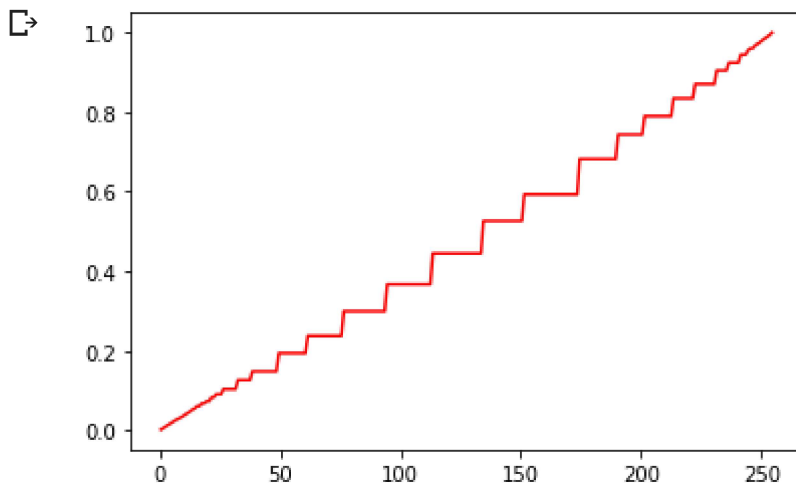
```
# Equalization
img_eq = exposure.equalize_hist(img)
img_eq = np.round(img_eq*255)
plt.imshow(img_eq, cmap = 'gray')
plt.show()
```



```
plt.hist(img_eq.ravel(), bins = bins)  
plt.show()
```



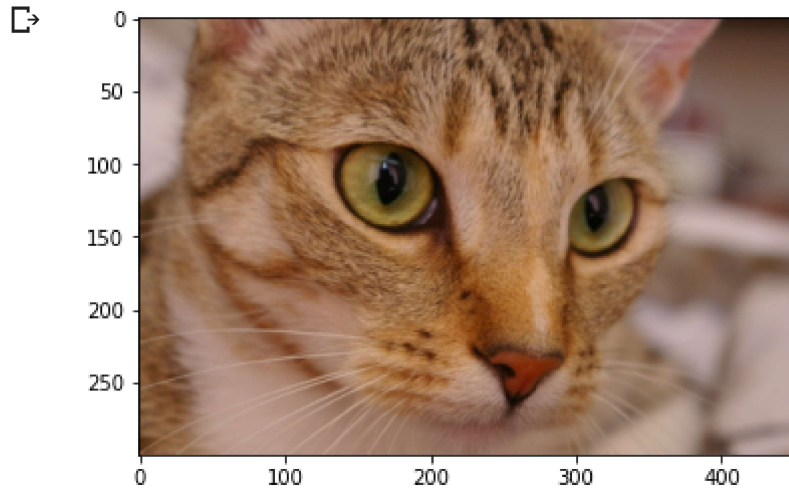
```
img_cdf, bins = exposure.cumulative_distribution(img_eq, bins)  
plt.plot(bins, img_cdf, 'r')  
plt.show()
```



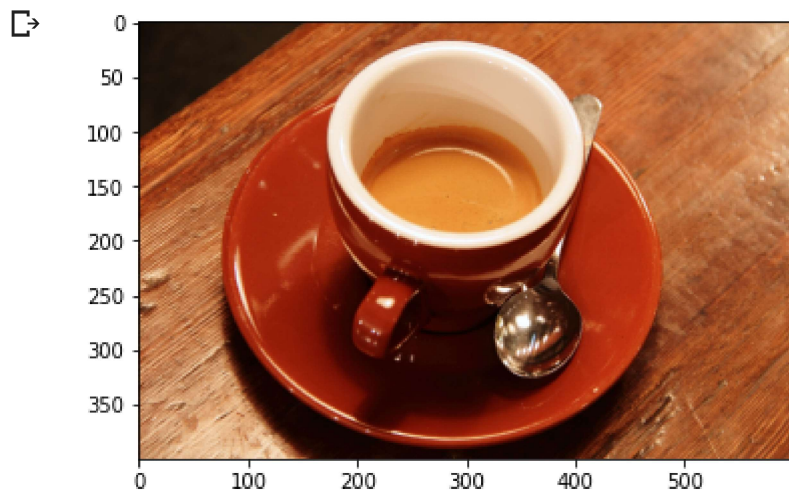
## ▼ Histogram Matching/Specification:

```
reference = data.coffee()  
image = data.chelsea()  
  
matched = match_histograms(image, reference, multichannel=True)
```

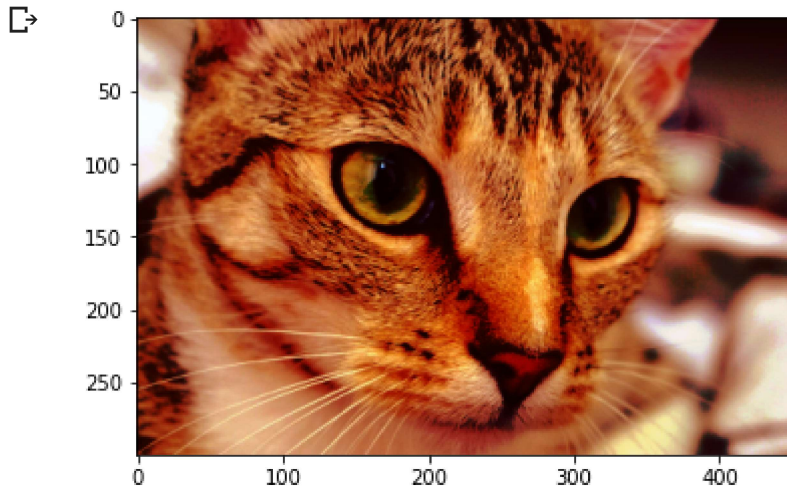
```
plt.imshow(image, cmap = 'gray')  
plt.show()
```



```
plt.imshow(reference, cmap = 'gray')  
plt.show()
```



```
plt.imshow(matched, cmap = 'gray')
plt.show()
```



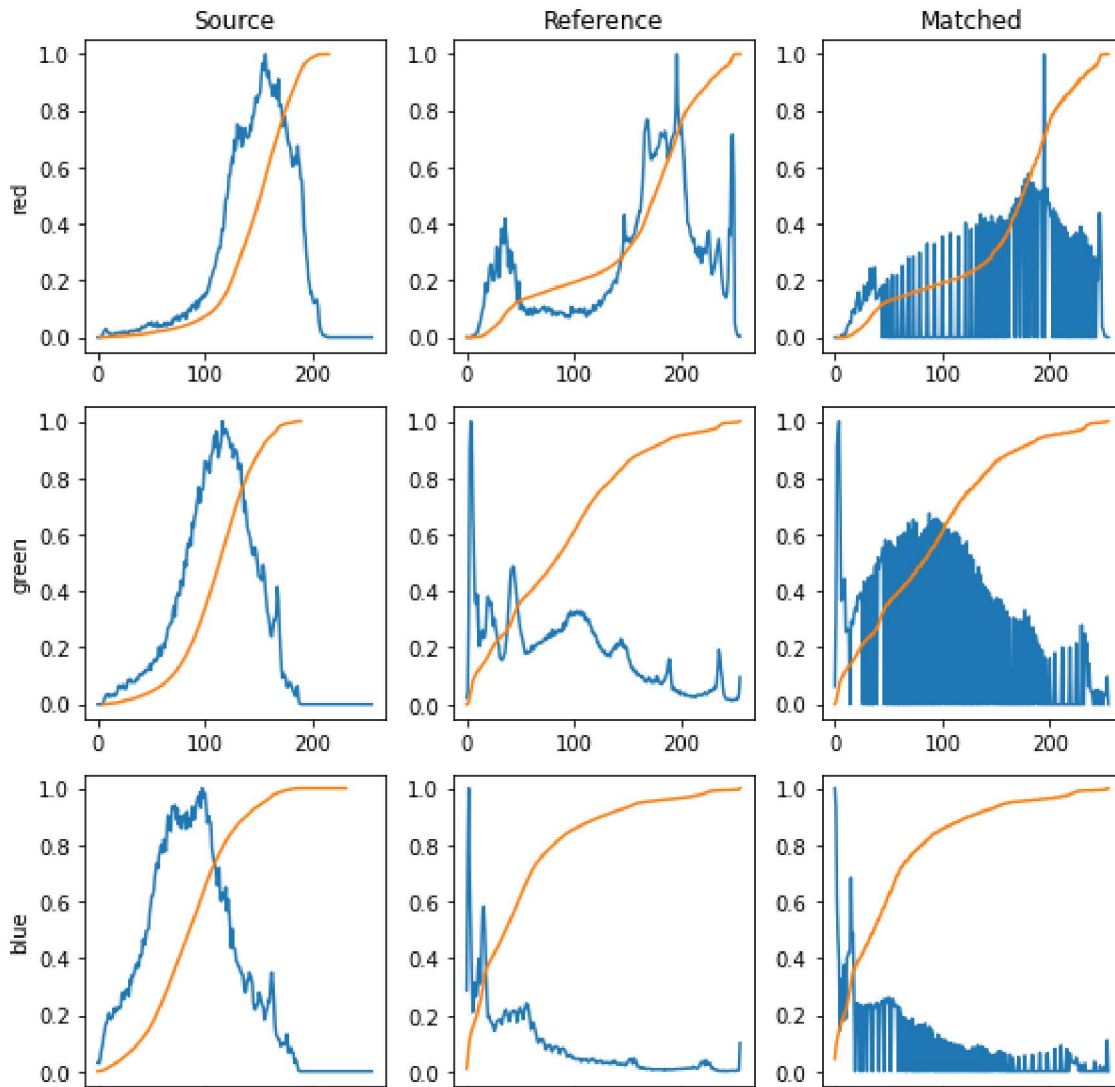
```
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(8, 8))

for i, img in enumerate((image, reference, matched)):
    for c, c_color in enumerate(('red', 'green', 'blue')):
        img_hist, bins = exposure.histogram(img[..., c], source_range='dtype')
        axes[c, i].plot(bins, img_hist / img_hist.max())
        img_cdf, bins = exposure.cumulative_distribution(img[..., c])
        axes[c, i].plot(bins, img_cdf)
        axes[c, 0].set_ylabel(c_color)

axes[0, 0].set_title('Source')
axes[0, 1].set_title('Reference')
axes[0, 2].set_title('Matched')

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





## ▼ Assignment:

1. Run this notebook and produce outputs.

Say you have two images (a) & (b). Perform the following tasks:

2. Equalize the histogram for image (a) and show the equalized histogram.
3. Apply histogram matching to image (a) so that its histogram matches image (b). Show the histogram of the image (b) and the image (c) and compare the histograms.
4. Write a brief discussion

Link to image (a): <https://raw.githubusercontent.com/suhailnajeer/ete-ice-472/master/experiment-4/image.jpg>

Link to image (b): <https://raw.githubusercontent.com/suhailnajeer/ete-ice-472/master/experiment-4/reference.jpg>

## ▼ Solution:

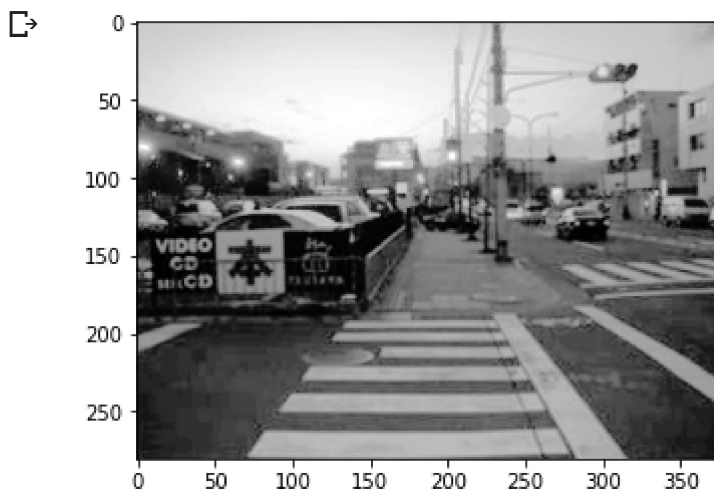
Loading the images:

```
from skimage import io
image1 = io.imread('https://raw.githubusercontent.com/suhailnajeed/ete-ice-472/master/experim
reference1 = io.imread('https://raw.githubusercontent.com/suhailnajeed/ete-ice-472/master/exp
```

## ▼ Assignment Task 2:

Equalizing the histogram for image (a):

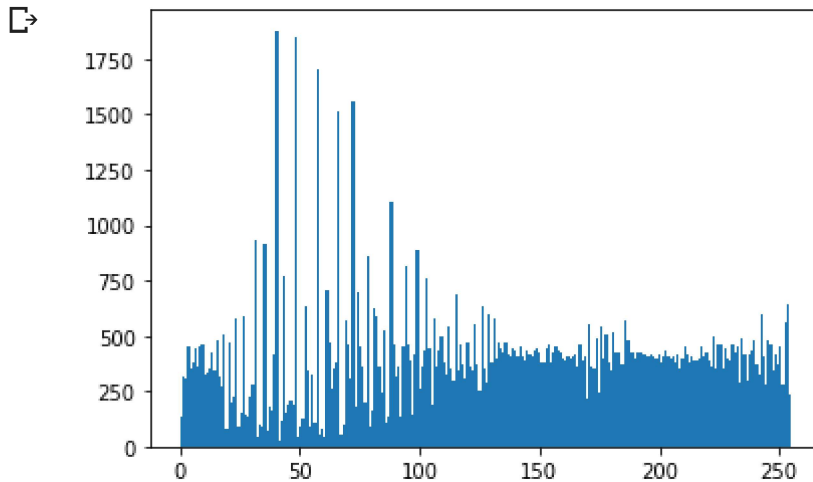
```
bins = 256
img1 = io.imread('https://raw.githubusercontent.com/suhailnajeed/ete-ice-472/master/experimen
, as_gray=True)
img1_eq = exposure.equalize_hist(img1)
img1_equalized = np.round(img1_eq*255)
plt.imshow(img1_equalized, cmap = 'gray')
plt.show()
```





Showing the equalized histogram:

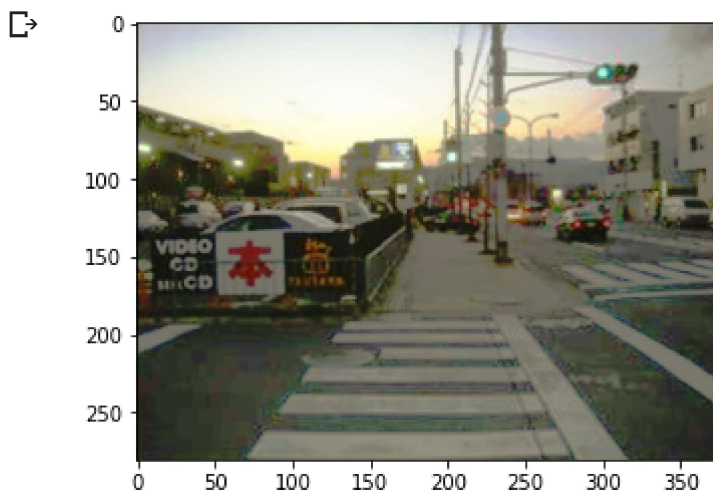
```
plt.hist(img1_equalized.ravel(), bins = bins)
plt.show()
```



## ▼ Assignment Task 3:

Applying histogram matching to image (a) so that it's histogram matches image (b):

```
matched1 = match_histograms(image1, reference1, multichannel=True)
plt.imshow(matched1, cmap = 'gray')
plt.show()
```



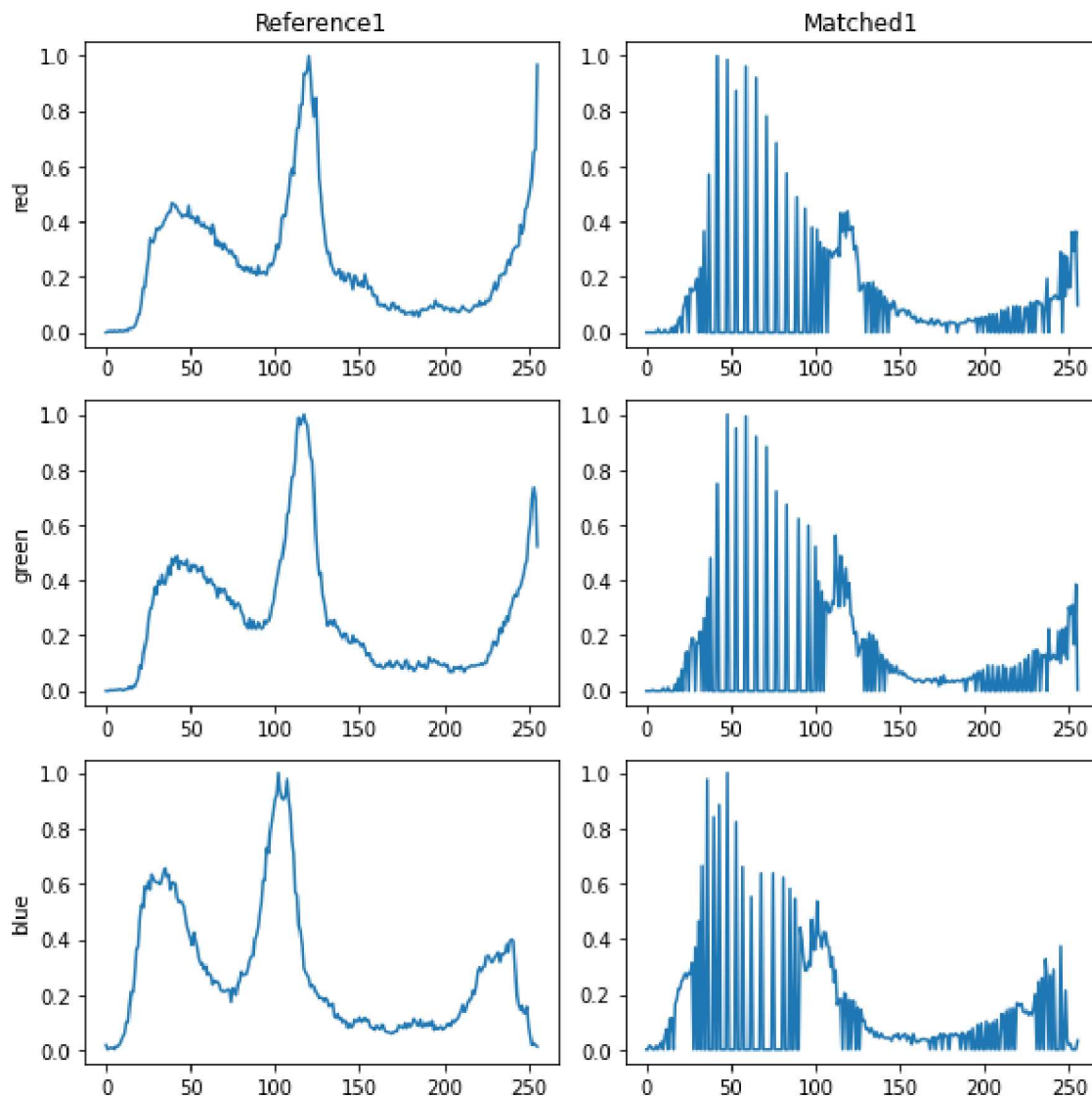
Showing the histogram of the image (b) which is **reference image** and the image (c) which is **matched image** and comparing the histograms.

```
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(8, 8))

for i, img in enumerate((reference1, matched1)):
    for c, c_color in enumerate(('red', 'green', 'blue')):
        img_hist, bins = exposure.histogram(img[..., c], source_range='dtype')
        axes[c, i].plot(bins, img_hist / img_hist.max())
        axes[c, 0].set_ylabel(c_color)

axes[0, 0].set_title('Reference1')
axes[0, 1].set_title('Matched1')

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



Here, we see the histograms of reference image and matched image. We know, Histogram matching is a way to transfer lighting of one image to another. If we see the source image histogram, we will see that, in red, green and blue channel the histogram is narrow in left side. That's why we have matched the source image to reference image. And we get matched image which convert the histogram of source image to reference image without changing the spatial arrangements of its pixel.

## ▼ Discussion:

In this experiment, I have learned about how to plot the histogram and CDF of an image. I have also learned how to make histogram equalized and plot it. Then I have learned how to make histogram matching of a image that it's histogram matches another image. For doing those, I have to make image ravel because image is 2D array but if we make image ravel the it will 1D array. For doing CDF, histogram equalization and histogram matching I have learned that, from skimage importing exposure module where `cumulative_distribution()`, `equalize_hist()` & `match_histograms()` functions are exists. Which makes it easy to get CDF, equalized and matched histogram. In assignment part, I have load the images which was given. To load image I have to import `io.imread`. Then I equalized the histogram for image (a) and showed the equalized histogram. Where the histogram was spread out. After that, I have applied histogram matching to image (a) so that it's histogram matches image (b). And showed the histogram of the image (b) and the image (c) and compare the histograms. Also showed the image. At last, now I can plot histogram and CDF, make histogram equalized and matched of any image.