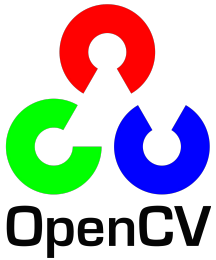
Histogram Equalization

Using Python3, Opencv2 & Matplotlib

Ghais Zaher

Tools Used

- Jupyter (Python3 Kernel)
- Opencv2 & Numpy
- Matplotlib: to draw images & charts



Algorithm & Code Steps

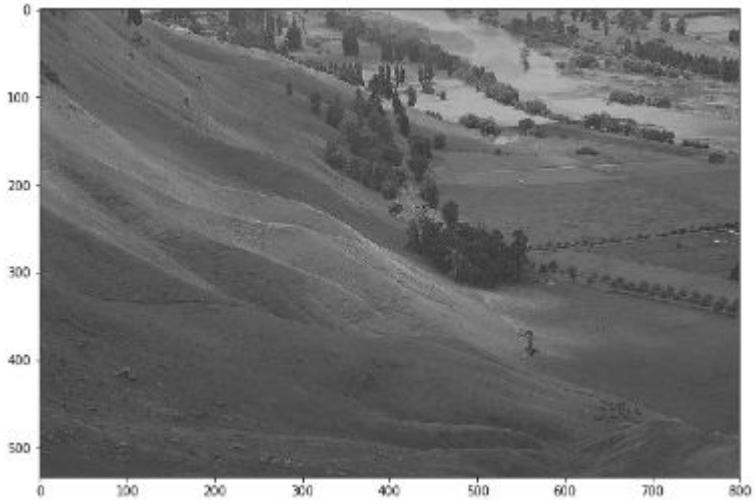
Imports

In [1]:
```
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

Algorithm & Code Steps

Read & Draw Image

```
image_path = '../images/hawkes.jpg'

img = cv2.imread(image_path)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

plt.figure(figsize=(10,10))
plt.imshow(gray, cmap='gray')
plt.show()
```



Algorithm & Code Steps

Histogram Generation Algorithm

- For an *N x M* image of *G* gray-levels create an array *H* of length *G* initialized with 0 values.
- Form the image histogram: Scan every pixel and increment the relevant member of *H*. if pixel p has intensity g_p, perform

$H[g_p]=H[g_p]+1$.

Source: Course slides (ImgProc2.pdf)

Algorithm & Code Steps

Histogram Generation Code

In [3]:

```
def histogram(gray, G):
    hist = np.zeros(G)
    for row in gray:
        for pixel in row:
            hist[pixel]+=1
    return hist
```

In [4]:

```
G = 256
hist = histogram(gray, G)
```

5	4	2	2
4	3	4	4
5	3	4	2
7	1	0	0

gray (4x4)

v	2	1	3	2	5	2	0	1
i	0	1	2	3	4	5	6	7

hist (G=8)

Algorithm & Code Steps

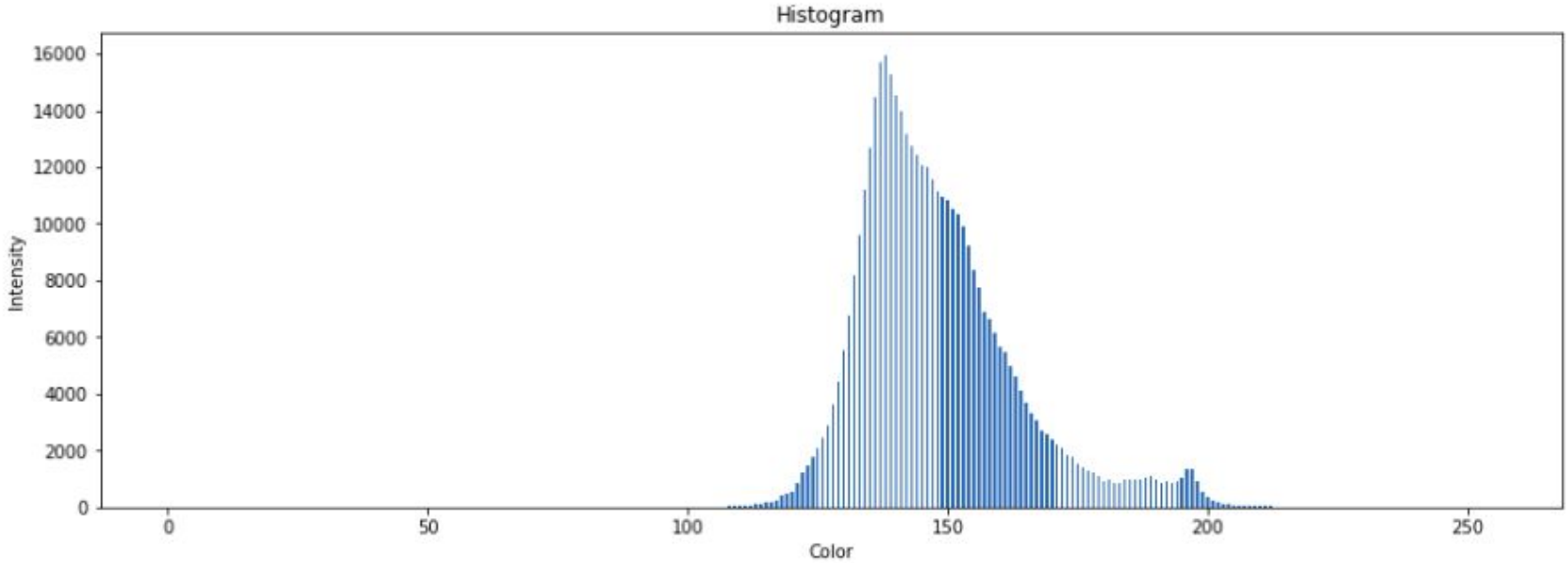
Plot Histogram (1)

```
def plot_histogram(hist, width=0.5, figsize=(15,5),
                   xlabel='Intensity', ylabel='Color',
                   title='Histogram'):
    plt.figure(figsize=figsize)
    plt.bar(list(range(len(hist))),hist,width=width)
    plt.ylabel(xlabel)
    plt.xlabel(ylabel)
    plt.title(title)

    plt.show()
```

Algorithm & Code Steps

Plot Histogram (2)

```
plot_histogram(hist)
```



Algorithm & Code Steps

Cumulative Histogram Generation Algorithm

- C = Array of length G
- C[0] = H[0]
- for i=1..G-1:
  - C[i] = sum(H[j]) , where j=0..i

Or:

- C = copy(H)
- C[i]=C[i]+C[i-1] , for all i=1..G-1

Algorithm & Code Steps

Cumulative Histogram Generation Code

```
hist_cum = hist.copy()
for i in range(1,G):
    hist_cum[i]+=hist_cum[i-1]
```

v	2	1	3	2	5	2	0	1
i	0	1	2	3	4	5	6	7

hist (G=8)

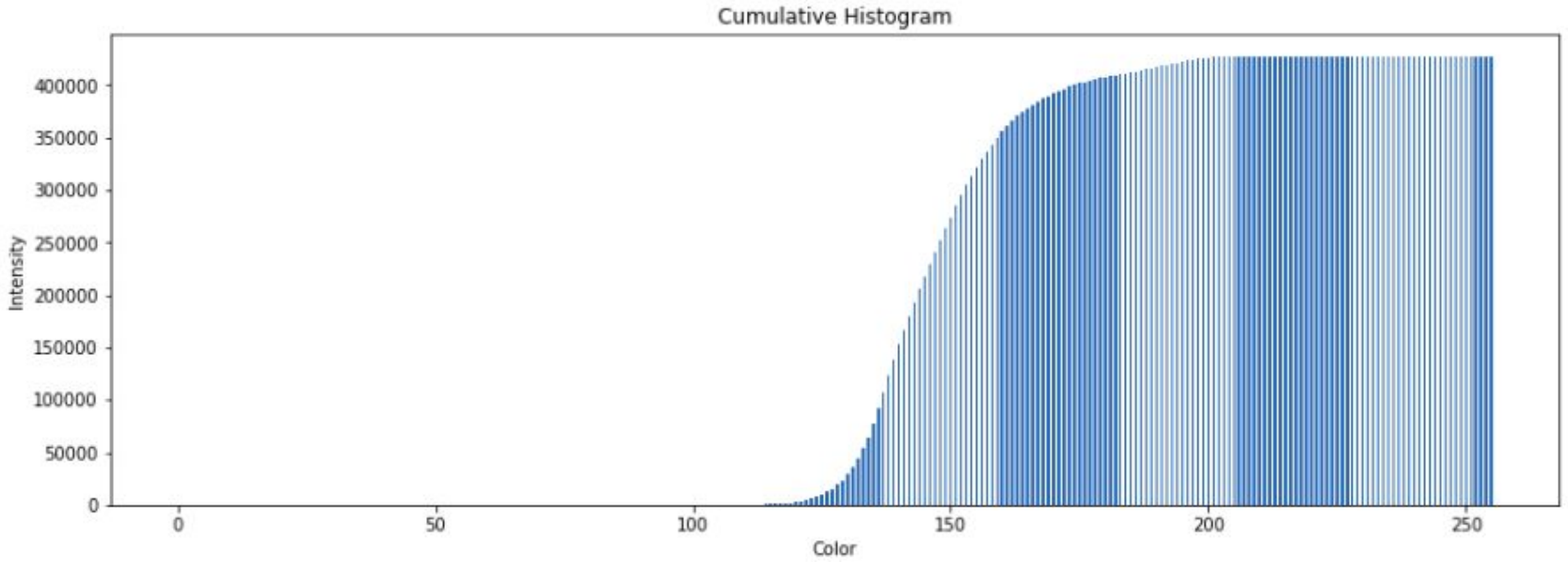
v	2	3	6	8	13	15	15	16
i	0	1	2	3	4	5	6	7

hist_cum

Algorithm & Code Steps

Plot Cumulative Histogram

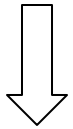
```
plot_histogram(hist_cum,title='Cumulative Histogram')
```



Algorithm & Code Steps

Equalization Algorithm & Code (1)

- Set T[p]=**round[(G-1)/(N•M)]** • $H_c[p]$.



```
N = gray.shape[0] #rows / height
M = gray.shape[1] #columns / width
t = np.round(((G-1)/(N*M))*hist_cum)
```

Source: Course slides (ImgProc2.pdf)

Algorithm & Code Steps

Equalization Algorithm & Code (2)

- Rescan the image and write an input with gray-levels g_q setting $g_q = T[g_q]$.

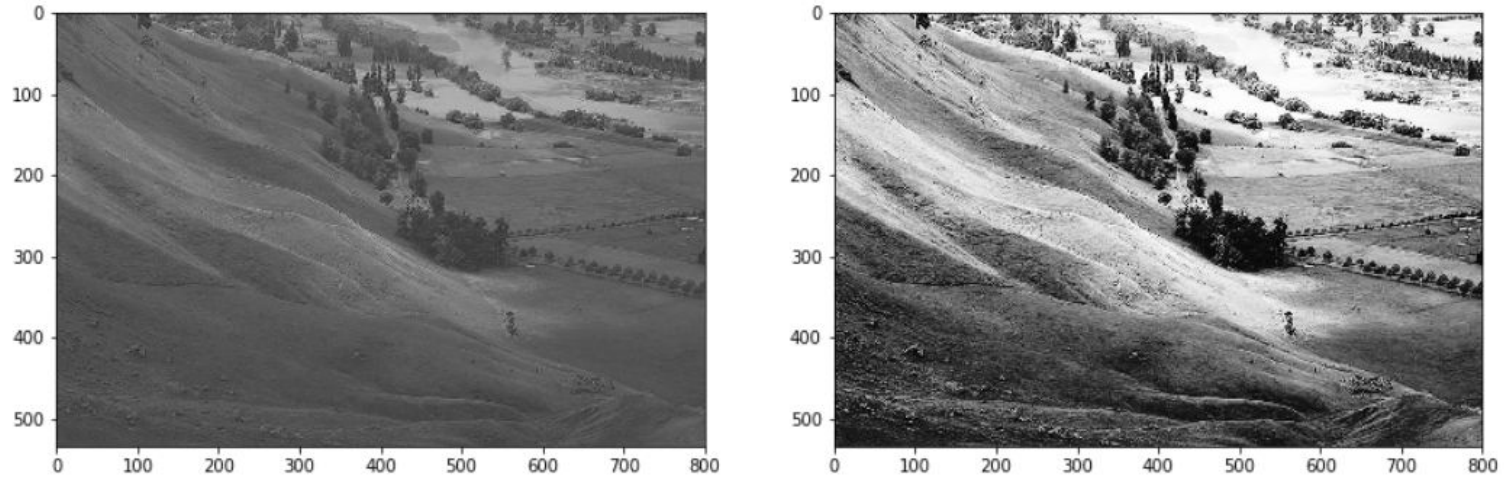
```
output = gray.copy()

for i in range(gray.shape[0]):
    for j in range(gray.shape[1]):
        output[i,j] = t[output[i,j]]
```

Source: Course slides (ImgProc2.pdf)

Algorithm & Code Steps

Results (1)

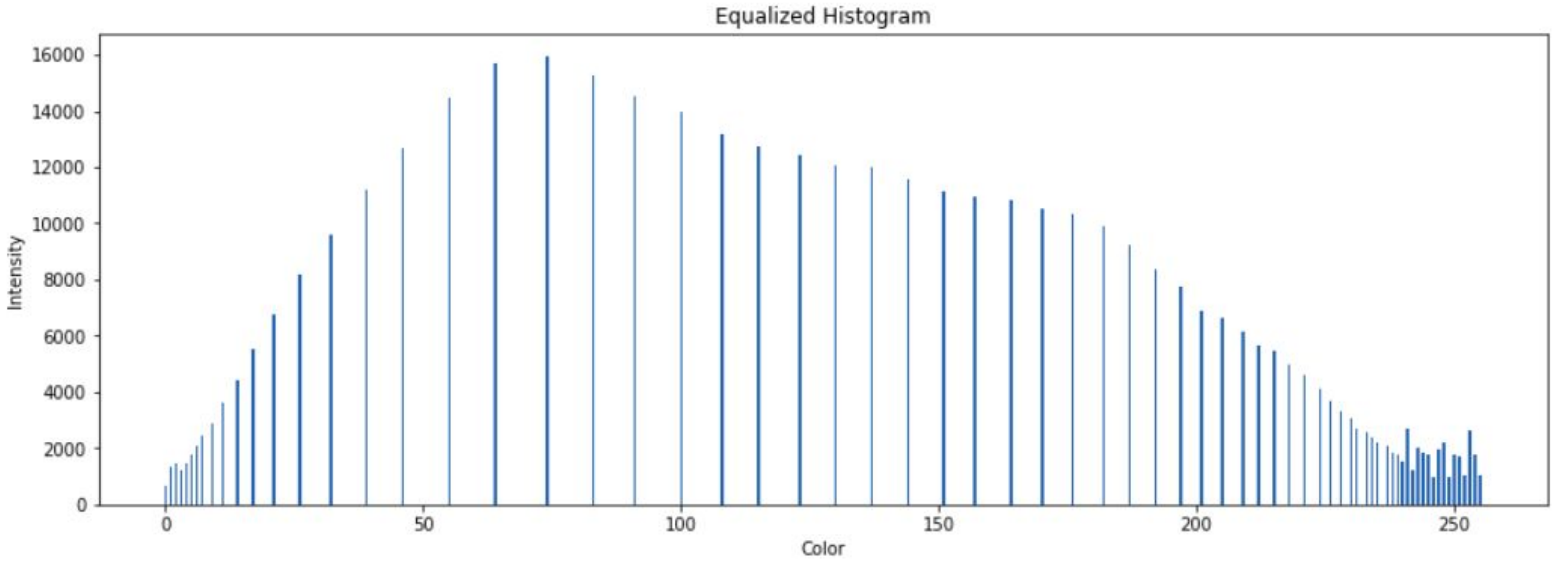
```
plt.figure(figsize=(15,15))
plt.subplot(121), plt.imshow(gray, cmap='gray')
plt.subplot(122),plt.imshow(output, cmap='gray')
plt.show()
```



Algorithm & Code Steps

Results (2)

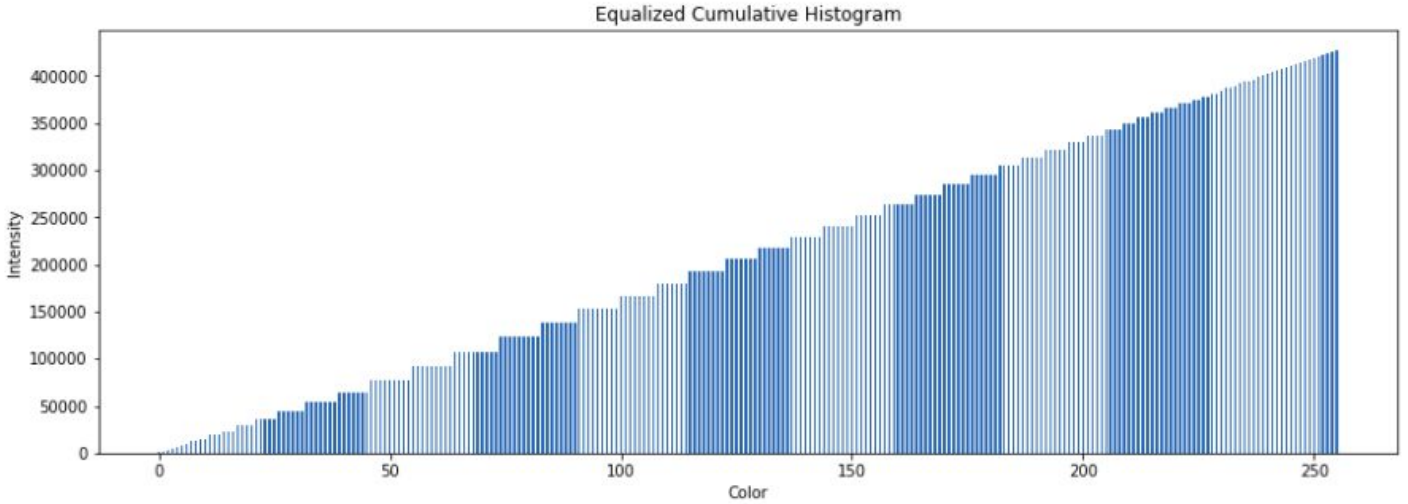
```
hist_after = histogram(output, G)
plot_histogram(hist_after, title="Equalized Histogram")
```



Algorithm & Code Steps

Results (3)

```
hist_cum_after = hist_after.copy()
for i in range(1,len(hist)):
    hist_cum_after[i]+=hist_cum_after[i-1]
plot_histogram(hist_cum_after, title="Equalized Cumulative Histogram")
```



Thanks

结束