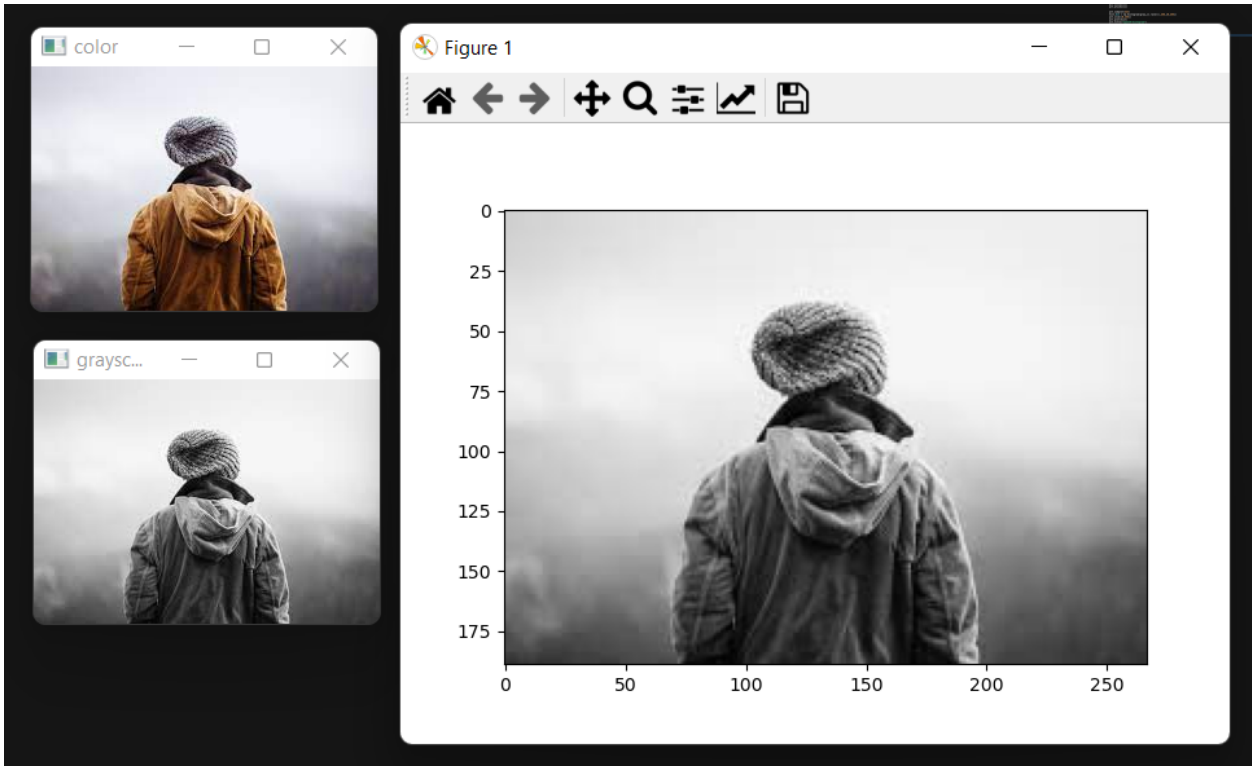
Assignment2 : Grayscale image and Image histogram

Change the BGR image to grayscale by using tools “openCV” and “matplotlib”. then , plot the images for comparison between 2 methods. The code and results are shown below.

histogram.py > ...

```python
from matplotlib.image import imread
import matplotlib.pyplot as plt
import numpy as np
import cv2 as cv

# use openCV
img = cv.imread('images.jpg')
cv.imshow("color", img)

gray_cv = cv.cvtColor(img,cv.COLOR_BGR2GRAY)
cv.imshow("grayscale from openCV", gray_cv)

# use matplotlib
def rgb_to_gray(img):
    gray_mat = np.zeros(img.shape)

    blue_ch = np.array(img[:,:,0])
    green_ch = np.array(img[:,:,1])
    red_ch = np.array(img[:,:,2])

    avg = ((blue_ch * 0.114) + (green_ch * 0.587) + (red_ch * 0.299))
    gray_mat = img.copy()

    for i in range(3):
        gray_mat[:,:,i] = avg
    return gray_mat

gray_mat = rgb_to_gray(img)
plt.imshow(gray_mat)
plt.show()
```


color
graysc...
Figure 1
0
25
50
75
100
125
150
175
0
50
100
150
200
250

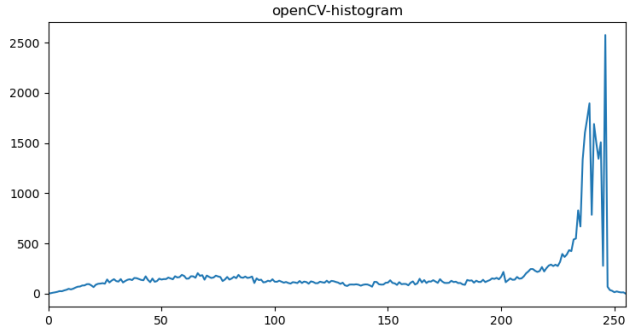
Plot the histogram of both grayscale images. Histogram was calculated using class in openCV and numpy.

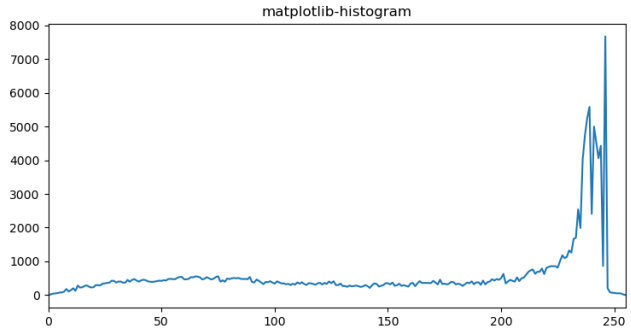
```
# plot histogram of both images
plt.subplot(221),plt.imshow(gray_cv,cmap="gray")
plt.title("image from openCV")
plt.xticks([])
plt.yticks([])

plt.subplot(222)
hist_cv = cv.calcHist([gray_cv],[0],None,[256],[0,256])
plt.xlim([0,255])
plt.plot(hist_cv)
plt.title("openCV-histogram")

plt.subplot(223),plt.imshow(gray_mat,cmap="gray")
plt.title("image from matplotlib")
plt.xticks([])
plt.yticks([])

plt.subplot(224)
hist_mat,bin = np.histogram(gray_mat.ravel(),256,[0,255])
plt.xlim([0,255])
plt.plot(hist_mat)
plt.title("matplotlib-histogram")

plt.show()
```

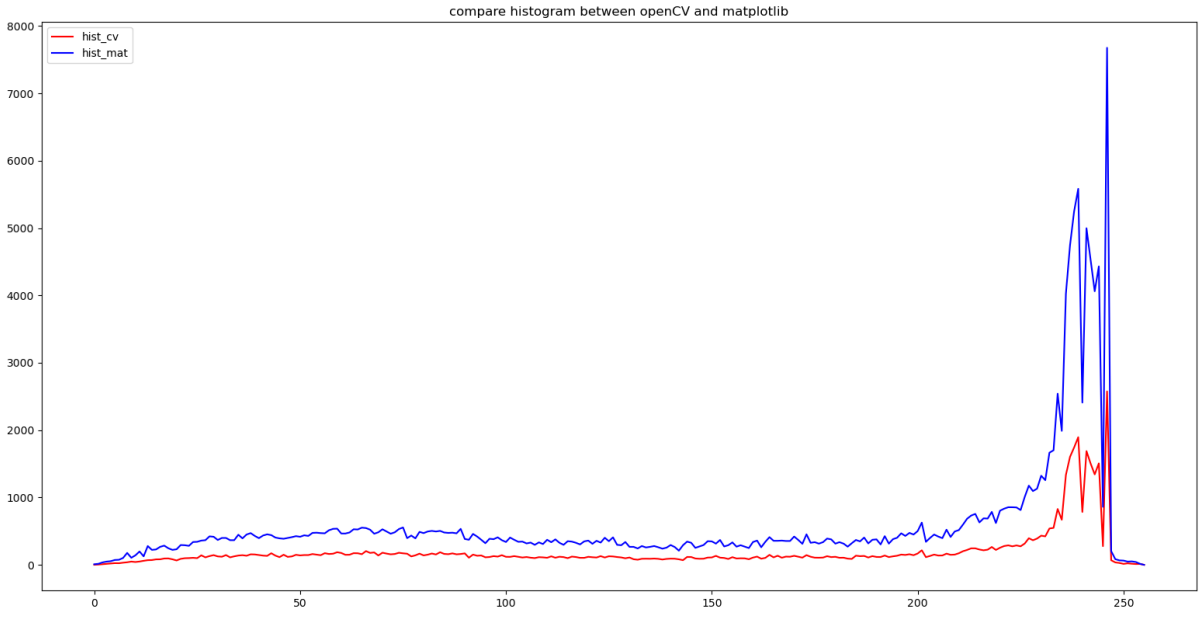
To clearly compare the results, plot in the same figure. However, they are on different scales.

```
hist_cv = cv.calcHist([gray_cv],[0],None,[256],[0,256])
hist_mat,bin = np.histogram(gray_mat.ravel(),256,[0,255])
plt.plot(hist_cv,'r')
plt.plot(hist_mat,'b')
plt.ylim(0,2500)
plt.legend(['hist_cv','hist_mat'])
plt.title("compare histogram between openCV and matplotlib")
plt.show()
```

The results are shown below. From the histogram, the pattern of distribution is quite similar.

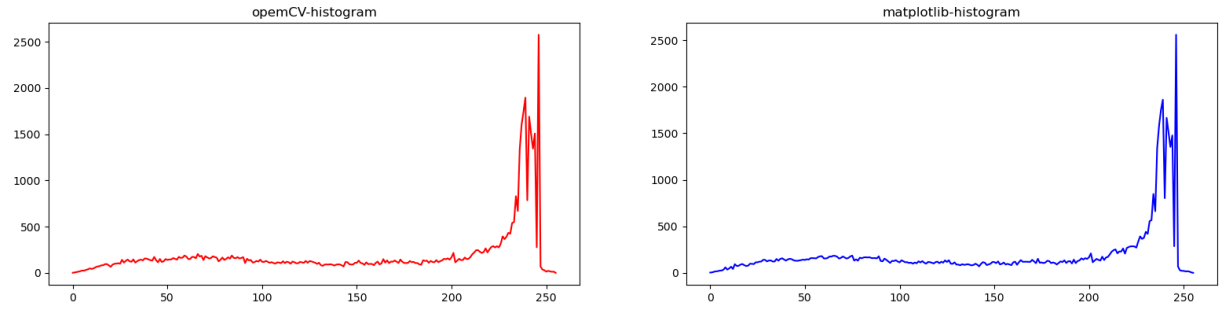


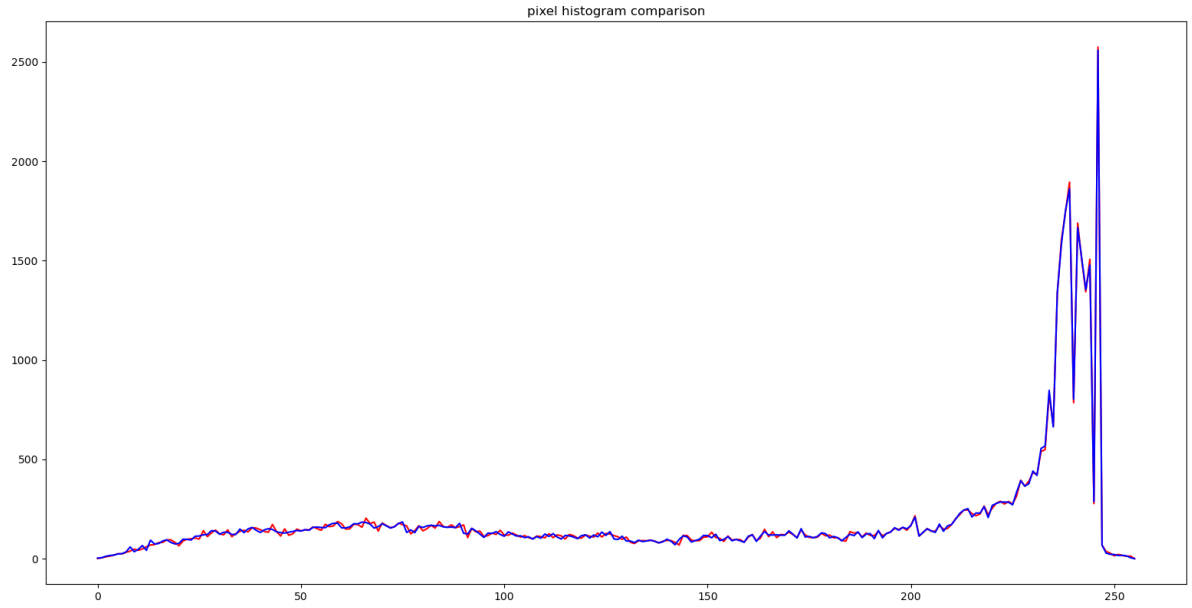




Plot histogram by using pixels in each image. The results are shown below. By using the pixels, both histograms are in the same scale which can be obviously seen that they are congruent.

```
# histogram from pixels
x = np.arange(0,256)

row, col = gray_cv.shape[0], gray_cv.shape[1]
a = np.zeros((256))
for i in range (0,row):
    for j in range (0,col):
        a[gray_cv[i,j]] += 1
plt.subplot(221)
plt.title("opemCV-histogram")
plt.plot(x,a,color='r')

row1, col1 = gray_mat.shape[0], gray_mat.shape[1]
b = np.zeros((256))
for i in range (0,row1):
    for j in range (0,col1):
        b[gray_mat[i,j]] += 1
plt.subplot(222)
plt.title("matplotlib-histogram")
plt.plot(x,b,color='b')
plt.show()
```





Note: I participated in the TESA competition which causes turning this homework out of your deadline.