

Homework 3 (Due:3/23)

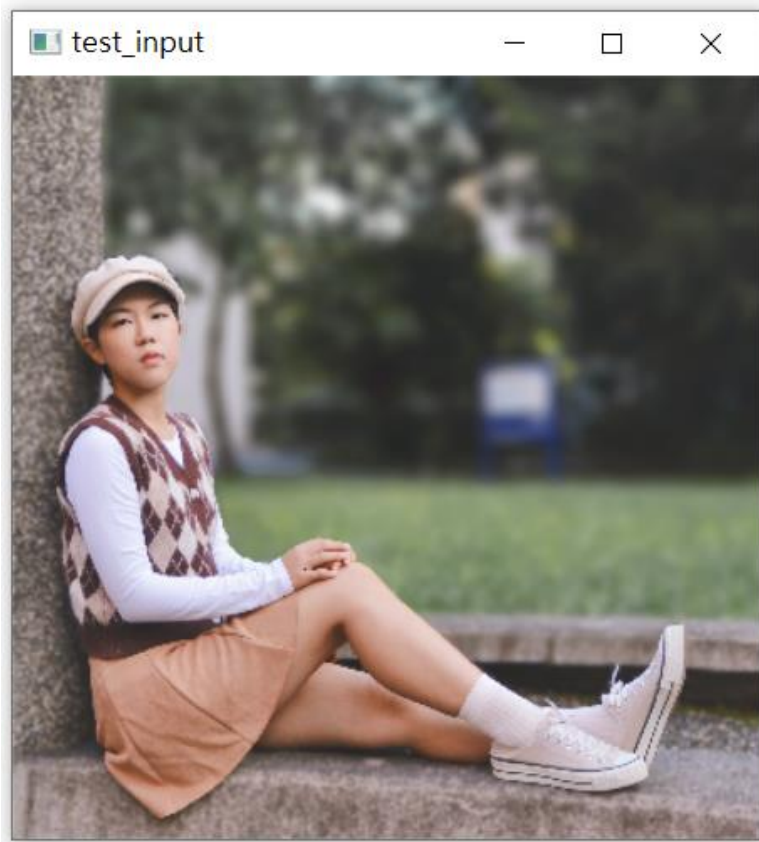
1. Develop a histogram equalization (HE) program;
2. Apply the HE to i) gray, ii) color images;
3. For each input image, print out the input/output images and their histograms.
4. Discuss your experiments.

For a color image C ,

- (i) Convert it into a gray image G ;
- (ii) Apply HE to G to get G' ;
- (iii) For each pixel of C , modify its color (r,g,b) by $(r',g',b') = (r,g,b) \times G' / G$.

C

input



Input_HE



G

Input_gray



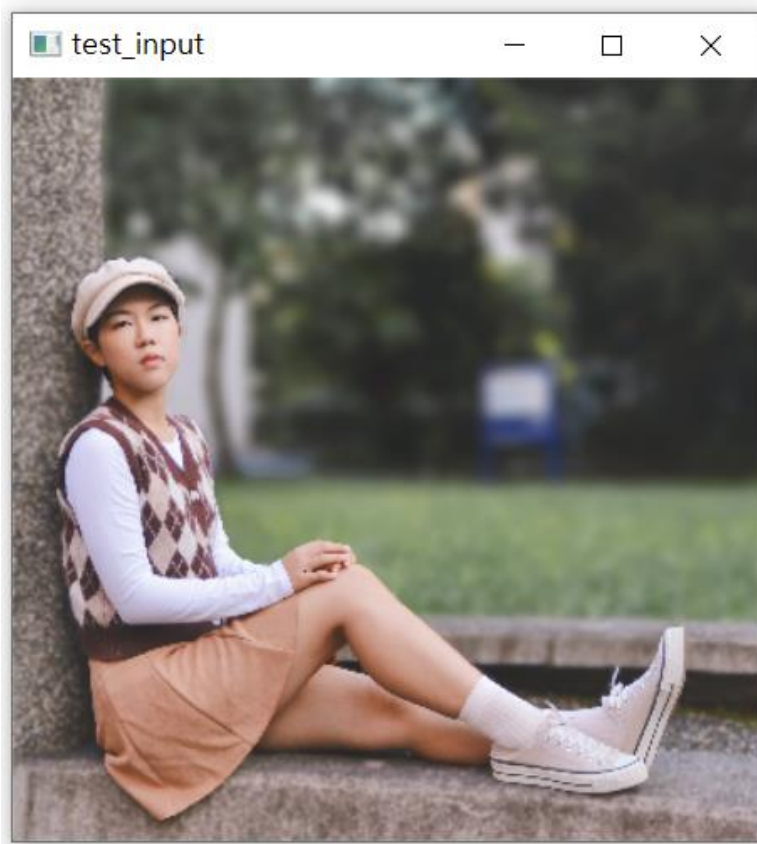
G'

input_gray_HE

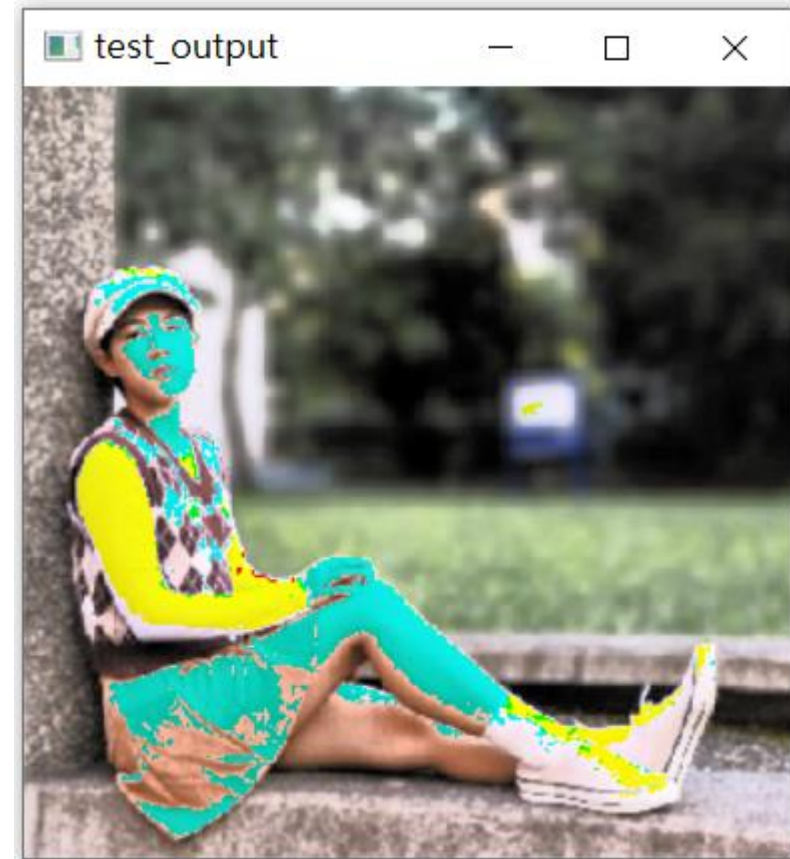


C

input



output



Source code

```
import cv2
import numpy as np

#讀入影像 imgC -> imgG
imgC = cv2.imread('test_input.jpg')
imgG = cv2.cvtColor(imgC,cv2.COLOR_BGR2GRAY)

zero_channel = np.zeros(imgC.shape[0:2], dtype = "uint8")
B, G, R = cv2.split(imgC)

#製作G'
hist, bins = np.histogram(imgG.flatten(), 256, [0, 256]) # 計算灰度直方圖
cdf = hist.cumsum() # 計算累積分布函數
# 將累積分布函數縮放到 0 至 255 間
cdf_scaled = ((cdf - cdf.min()) * 255 / (cdf.max() - cdf.min())).astype('uint8' )
imgGG = cv2.LUT(imgG, cdf_scaled) # 將像素值重新映射
```

Source code

```
#處理影像 (r' ,g' ,b' ) = (r,g,b) X G' / G
height, width = imgG.shape
for i in range(height):
    for j in range(width):
        t=imgGG[i][j]/imgG[i][j]
        B[i][j]=(B[i][j])*t
        G[i][j]=(G[i][j])*t
        R[i][j]=(R[i][j])*t

imgC = cv2.merge([B,G,R])

#新圖片存檔
cv2.imwrite('test_output.jpg', imgC)
print("test_output.jpg is SAVED~")
```


Comments

In the gray image case everything looks okay (p3). However when I inputted the color image, the output looks weird. Google said the reason is that the rate of R,G and B is broken. Then I clipped the image and inputted it again, founding that it looks better. So I guess that it is because **A** and **B** has different value of G' / G .

