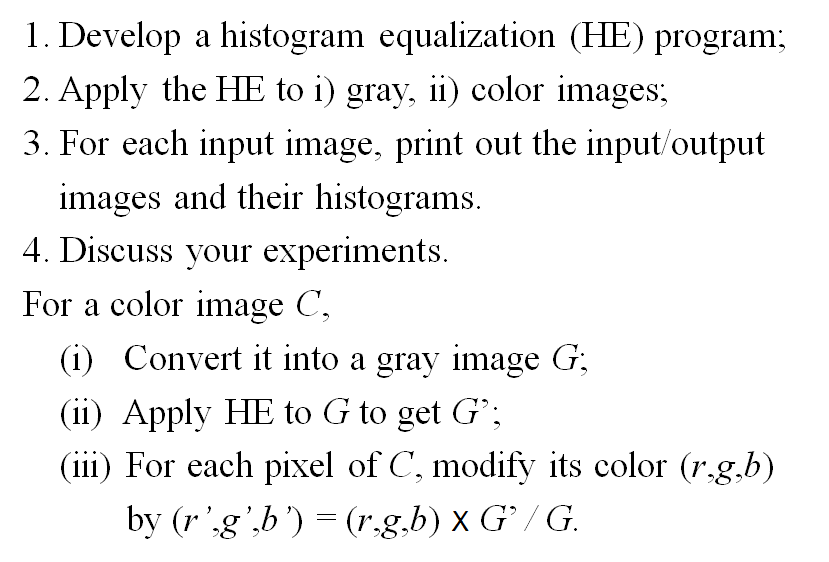
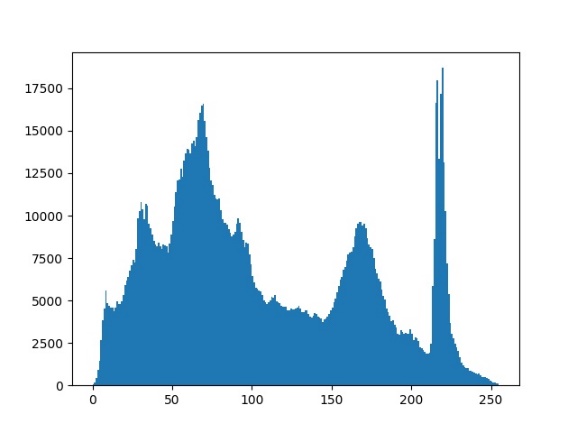
影像處理 Homework 3

資工112 40847015S 紀軒宇

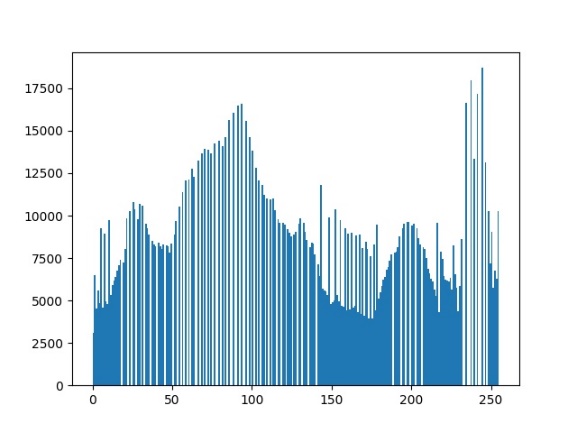
* Statement



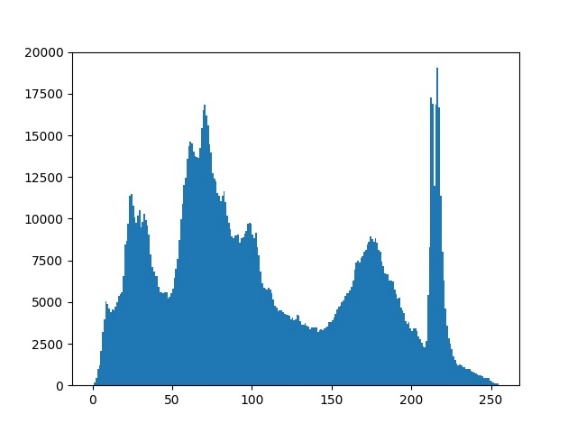
* Code
* #import the library
* import cv2
* import numpy as np
* import math
* import matplotlib.pyplot as plt
* def histo\_equalization(img\_name):
* img = cv2.imread(img\_name)
* gray\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)
* h\_gray\_img = gray\_img.copy()
* y, x = gray\_img.shape
* level = list(range(256))
* plt.hist(gray\_img.flatten(), bins=255)
* plt.savefig(img\_name[0]+"\_hist.jpg")
* plt.close()
* img\_1dim = gray\_img.flatten()
* his\_data = np.bincount(img\_1dim, minlength=256)
* acc\_data = np.cumsum(his\_data)
* acc\_prob = acc\_data/(x\*y)
* roundvalue = np.rint(acc\_prob\*255).astype('uint8')
* for i in range(y):
* for j in range(x):
* h\_gray\_img[i][j] = roundvalue[h\_gray\_img[i][j]]
* output = img\*(np.tile(np.reshape(np.divide(h\_gray\_img, gray\_img, where=gray\_img!=0), (y, x, 1)), (1, 1, 3)))
* output = np.clip(output, 0, 255).astype('uint8')
* plt.hist(h\_gray\_img.flatten(), bins=255)
* plt.savefig(img\_name[0]+"\_hist\_H.jpg")
* plt.close()
* cv2.imwrite(img\_name[0]+"\_H.jpg", output)
* return
* histo\_equalization("G.jpg")
* histo\_equalization("C.jpg")
* img = cv2.imread("C.jpg")
* b, g, r = cv2.split(img)
* plt.hist(b.flatten(), bins=255, color="#0000FF")
* plt.savefig("C\_b\_hist.jpg")
* plt.close()
* plt.hist(g.flatten(), bins=255, color="#00FF00")
* plt.savefig("C\_g\_hist.jpg")
* plt.close()
* plt.hist(r.flatten(), bins=255, color="#FF0000")
* plt.savefig("C\_r\_hist.jpg")
* plt.close()
* img = cv2.imread("C\_H.jpg")
* b, g, r = cv2.split(img)
* plt.hist(b.flatten(), bins=255, color="#0000FF")
* plt.savefig("C\_H\_b\_hist.jpg")
* plt.close()
* plt.hist(g.flatten(), bins=255, color="#00FF00")
* plt.savefig("C\_H\_g\_hist.jpg")
* plt.close()
* plt.hist(r.flatten(), bins=255, color="#FF0000")
* plt.savefig("C\_H\_r\_hist.jpg")
* plt.close()
* 輸入圖片(灰階)

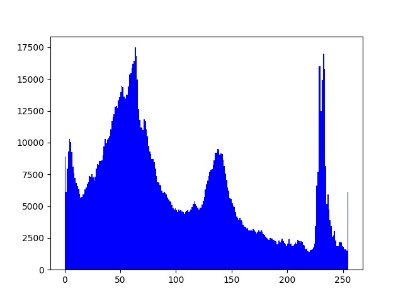
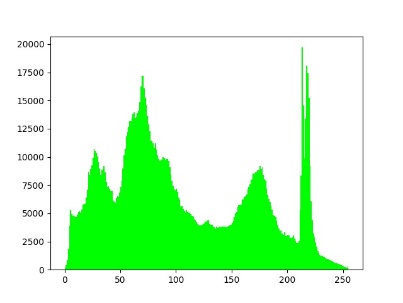
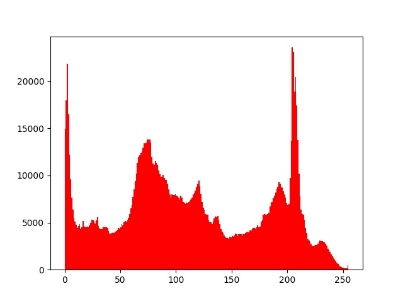
* 輸出圖片(灰階)

* 輸入圖片(彩色)

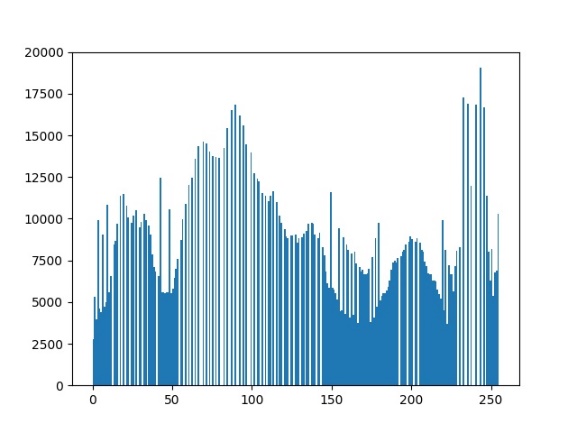
 

Grayscale

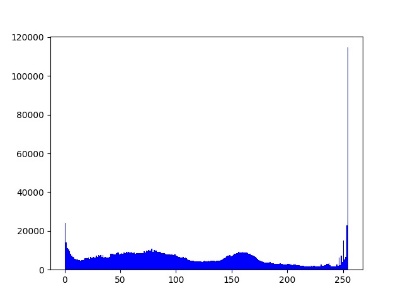
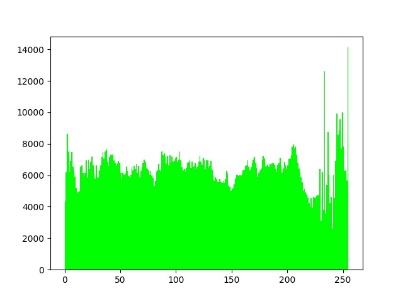
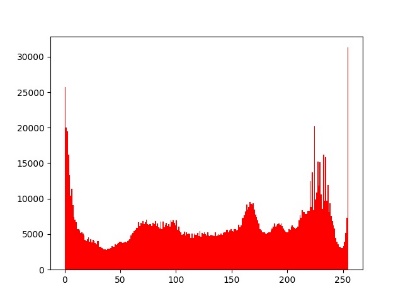


Red Green Blue

* 輸出圖片(彩色)



Grayscale



Red Green Blue

* 心得

藉由基於統計的方法，可以將影像的對比度做調整，從輸入可以看出有一些灰階特別的多，藉由這樣的方法，我們將那些出現頻率較高的灰階平均的移動到了其附近的幾個灰階度，最後得出輸出的結果看出影像的對比度提升了。