import os

import cv2 as cv2

import numpy as np

from matplotlib import pyplot as plt

def histogram(img):

histogram\_ = np.zeros(256, dtype=np.uint32) # 存放讀到的pixel值

for k in range(len(histogram\_)):

histogram\_[k] = np.sum(img.flat == k) # 將符合 SUM 起來

return histogram\_

def equalize(img,histogram\_):

def sumcdf(histogram\_):

cdf = np.zeros\_like(histogram\_, dtype=np.uint32) # 累積分布函數（cdf）

cdf[0] = histogram\_[0] # 因為下方 cdf[k-1], 在 cdf[0] 時前面沒有值,有可能會讀到亂七八糟的東西,因此要先將[0]先寫進去

for k in range(len(cdf)):

cdf[k] = cdf[k-1] + np.sum(img.flat == k) # 取的cdf累計"值"

return cdf

cdf\_pro = sumcdf(histogram\_) / (img.shape[0] \* img.shape[1]) # 算出cdf的機率(probability)

cdf\_value = np.zeros\_like(img, dtype=np.uint8) # 存放乘以CDF的pixel值

for row in range(img.shape[0]):

for col in range(img.shape[1]):

cdf\_value[row,col] = np.uint8(np.round((cdf\_pro[img[row,col]] - np.min(cdf\_pro)) / (np.max(cdf\_pro) - np.min(cdf\_pro)) \* 255))

# 將row col的值,乘上對應cdf出現的機率值 = 均化

return cdf\_value # 將均化後的值回傳

def gray():

img\_path='E:/Program\_File/PYTHON/數位影像處理作業/HW\_1/snow.jpg'

img\_filepath = os.path.splitext(img\_path)[0] # 拆分路徑 & 副檔名，0 為路徑

img\_fileextension = os.path.splitext(img\_path)[1] # 1 為副檔名

img\_filename = os.path.basename(img\_filepath) # 取出檔名不含副檔名

img = cv2.imread(img\_filename+img\_fileextension,cv2.IMREAD\_GRAYSCALE)

histogram\_ = histogram(img)

cdf\_value = equalize(img,histogram\_)

histogram\_equ = histogram(cdf\_value)

plt.bar(range(len(histogram\_)), histogram\_, alpha=0.9, width = 1, lw=1)

plt.figure()

plt.bar(range(len(histogram\_)), histogram\_equ, alpha=0.9, width = 1, lw=1)

plt.show()

plt.close()

cv2.imshow('img',img)

cv2.imshow('equalize',cdf\_value)

cv2.waitKey()

cv2.destroyAllWindows()

cv2.imwrite(img\_filename+'\_gray'+img\_fileextension,img)

cv2.imwrite(img\_filename+'\_gray\_equalize'+img\_fileextension,cdf\_value)

def multicolor():

img\_path='E:/Program\_File/PYTHON/數位影像處理作業/HW\_1/star.jpg'

img\_filepath = os.path.splitext(img\_path)[0] # 拆分路徑 & 副檔名，0 為路徑

img\_fileextension = os.path.splitext(img\_path)[1] # 1 為副檔名

img\_filename = os.path.basename(img\_filepath) # 取出檔名不含副檔名

img = cv2.imread(img\_filename+img\_fileextension,-1) # 讀檔

img\_b, img\_g, img\_r = cv2.split(img) # 將對檔案分割，順序為BGR，而非RGB

img\_b\_hist = histogram(img\_b) # 分別將 BGR 做 histogram

img\_g\_hist = histogram(img\_g)

img\_r\_hist = histogram(img\_r)

img\_b\_eq = equalize(img\_b, img\_b\_hist) # 各別接著做 equalize

img\_g\_eq = equalize(img\_g, img\_g\_hist)

img\_r\_eq = equalize(img\_r, img\_r\_hist)

img\_b\_eq\_hist = histogram(img\_b\_eq) # 最後將 equalize 後的 BGR 做 histogram

img\_g\_eq\_hist = histogram(img\_g\_eq)

img\_r\_eq\_hist = histogram(img\_r\_eq)

plt.subplot(3,2,1) # 子圖，( 3 \* 2 (列column \* 行row), 第一個圖，左上)

plt.bar(np.arange(img\_b\_hist.shape[0]), img\_b\_hist, alpha=0.9, width = 1, lw=1, align = 'center',color = 'b')

plt.subplot(3,2,2) # 子圖，( 3 \* 2 , 第二個圖，右上)

plt.bar(np.arange(img\_b\_eq\_hist.shape[0]), img\_b\_eq\_hist, alpha=0.9, width = 1, lw=1, align = 'center',color = 'b')

plt.subplot(3,2,3) # 子圖，( 3 \* 2 , 第三個圖，左中)

plt.bar(range(len(img\_g\_hist)), img\_g\_hist, alpha=0.9, width = 1, lw=1, align = 'center',color = 'g')

plt.subplot(3,2,4)

plt.bar(range(len(img\_g\_eq\_hist)), img\_g\_eq\_hist, alpha=0.9, width = 1, lw=1, align = 'center',color = 'g')

plt.subplot(3,2,5)

plt.bar(range(len(img\_r\_hist)), img\_r\_hist, alpha=0.9, width = 1, lw=1, align = 'center',color = 'r')

plt.subplot(3,2,6)

plt.bar(range(len(img\_r\_eq\_hist)), img\_r\_eq\_hist, alpha=0.9, width = 1, lw=1, align = 'center',color = 'r')

plt.show()

plt.close() # 開啟圖片後要記得關閉，當有大量的圖片或資料開啟後，記憶體可能會爆，因此要手動關閉他

img\_merge = cv2.merge([img\_b\_eq,img\_g\_eq,img\_r\_eq]) # 將均化後的 BGR 合併

cv2.imwrite(img\_filename+'\_merge\_equalize'+img\_fileextension,img\_merge)

cv2.imshow('img',img)

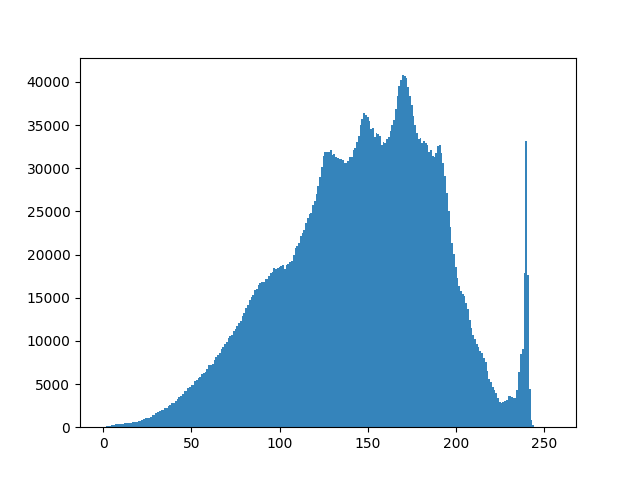
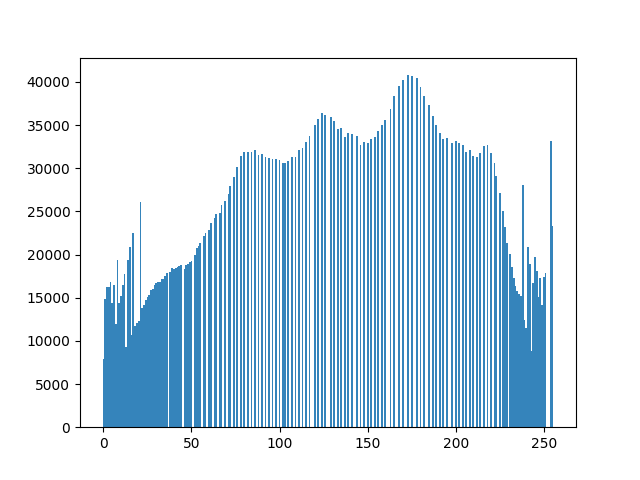
cv2.imshow('equalize',img\_merge)

cv2.waitKey(0)

cv2.destroyAllWindows()

# gray()

# multicolor()

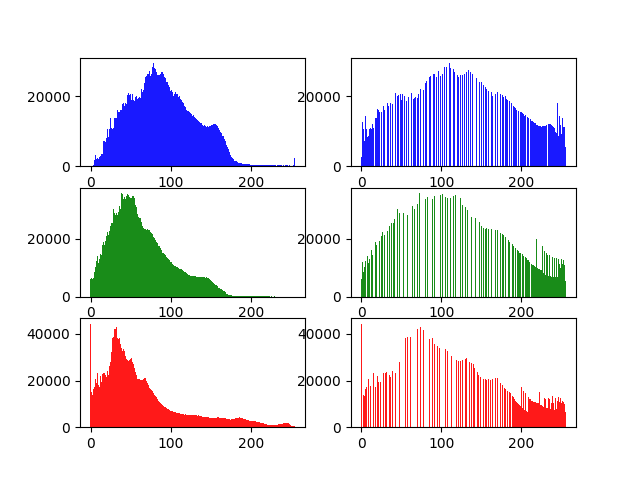
GRAY :







multicolor :



心得 :

在做直方圖以及均化之前對numpy、cv2、matplotlib，完全沒有概念，花了許多時間對其各種用法進行嘗試。

在寫直方圖時整體還算順利，但是當要做均化時我參考了維基百科上的數學公式，有做出CDF，在我以為我已經成功時，跑出來的圖卻並不正確，因為實際上我只有算出機率，並沒有乘上原本的值，在與同學請教過後，才成功的求出均化後的值並成功輸出