生機碩一 許喬淇 r12631001

Part 1: (50%) Histogram of an Image

- **程式碼**

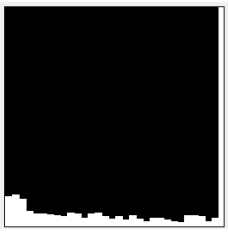
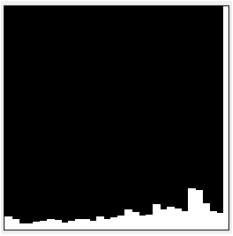
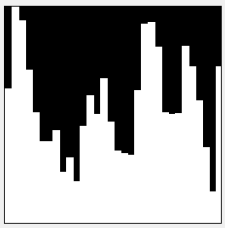
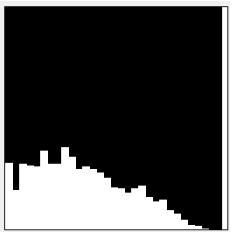
1. 建立 base32 轉 0~31 灰階值之字典 base32_dic
2. 初始化陣列 img_JET、img_LISA、img_LINCOLN、img_LIBERTY，用以儲存轉換後的灰階影像
3. 轉換 base32 為灰階值之方法：load_img()
   (1) 用 with open(filename, "r") as f 讀入圖片，讀取的方式為 read().splitlines()，依照特定字元切割，以 List 資料型態儲存
   (2) 用兩個 for 迴圈，逐一拜訪每個元素，並根據字典 base32_dic 將符號轉換為灰階值
4. 畫直方圖之方法：show_histogram()
   (1) 初始化一陣列 his_arr，其長度為 32，用以儲存每個灰階值之數量
   (2) 用兩個 for 迴圈計算每個灰階值之數量，數量存入陣列 his_arr
   (3) 初始化直方圖高、寬(hist_height, hist_width)以及直方圖陣列
   (4) 正規化陣列 his_arr
   (5) 用 cv2.rectangle()畫出直方圖
   (6) 把直方圖輸出至介面

5. 將直方圖 return 給其他函式使用：histogram()

   作法同 4.

- **結果**

直方圖之縱軸數值為數量，橫軸為灰階值，並每張圖分別經正規化處理，故每張圖中灰階數量最多的值均為圖高

	JET.64	LIBERTY.64	LINCOLN.64	LISA.64
灰階圖				
直方圖				

- **討論**

1. 觀察 JET.64 和 LIBERTY.64 之直方圖發現白色的數量遠多於其他灰階值

2. 觀察 LINCOLN.64 之直方圖發現淺色和深色灰階分布數量相近

3. 觀察 LISA.64 之直方圖發現有隨著灰階值上升數量下降的情形

Part 2: (50%) Arithmetic Operations of an Image Array

1. Add or subtract a constant value to each pixel in the image：

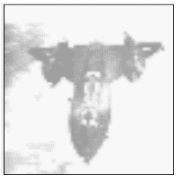
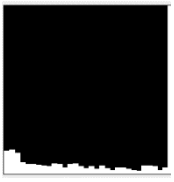
   add_constant()

   (1) 使用 np.add()將原圖的所有像素 img_arr.astype("uint16")都加上一常數值 constant，得到 img_add

   (2) 使用 np.clip()將 img_add 中元素限制在 0 和 31 之間

   (3) 繪製直方圖和灰階圖

   ■ 結果

constant	原灰階圖	原直方圖	新灰階圖	新直方圖
10				
20				
30				

   ■ 討論

   影像隨著常數的增加變淡，直方圖則是往右偏移，並且白色像素點數量增加

2. Multiply a constant to each pixel in the image：multiply_constant()

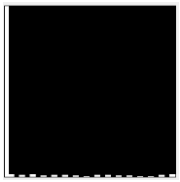
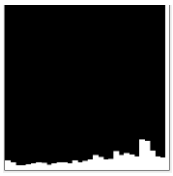
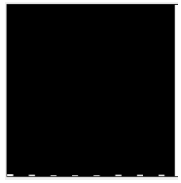
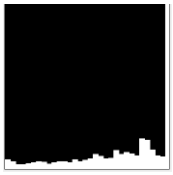
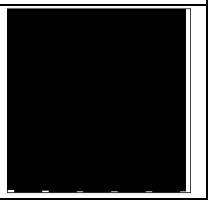
   (1) 使用 np.multiply()將原圖的所有像素 img_arr.astype("uint16")

都乘以一常數值 constant，得到 img_multiply

(2) 使用 np.clip()將 img_multiply 中元素限制在 0 和 31 之間

(3) 繪製直方圖和灰階圖

■ 結果

constant	原灰階圖	原直方圖	新灰階圖	新直方圖
2				
4				
6				

■ 討論

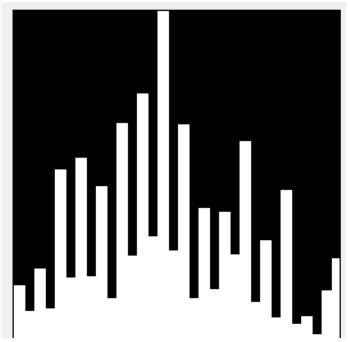
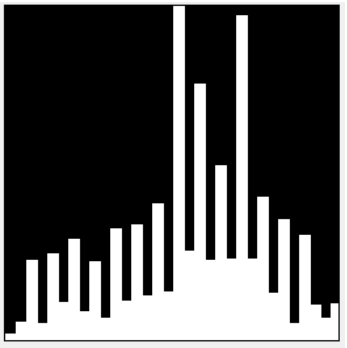
影像隨著常數的增加變淡，觀察直方圖發現白色像素點數量增加，但沒有偏移之情形。

3. Create a new image which is the average image of two input images：two_average()

(1) 選擇兩張圖片，將其矩陣相加後除以 2，得到 img_avg

(2) 繪製直方圖和灰階圖

■ 結果

LISA & LINCOLN 灰階圖	LISA & LINCOLN 直方圖
	
JET & LINCOLN 灰階圖	JET & LINCOLN 直方圖
	

■ 討論

可以在新的影像上看到相加的兩張圖片，並且較明顯的皆為深色像素點。

4. Create a new image g(x,y) in which the value of each pixel is determined by calculating the pixel values of the input image f(x,y)：

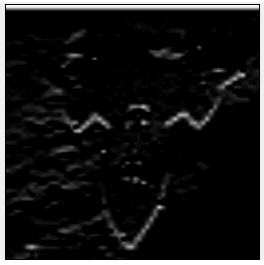
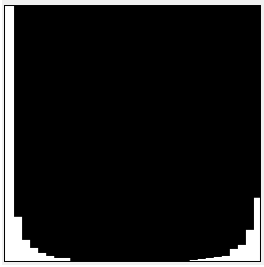
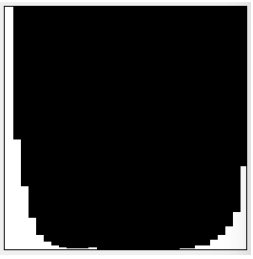
   $g(x,y) = f(x,y) - f(x-1,y)$ ：g_function()

   (1) 初始化陣列 img_g

   (2) 計算 g(x,y)，當 x=0，g(x,y) = f(x,y)

(3) 繪製直方圖和灰階圖

■ 結果

	新的影像	新的直方圖
JET.64		
LISA.64		

■ 討論

將 f(x,y)轉換為 g(x,y)可以觀察到影像中的輪廓，是因為當 f(x,y)、f(x-1,y)相差較大時，相減後之數值也會較大，有助於檢測到圖像中的邊緣，且直方圖中可以發現黑跟白的像素點最多，呈現開口向上之曲線。