多媒體程式設計
影像資料處理

Instructor: 馬豪尚

OpenCV繪製圖形

- 畫直線 line(img, pt1, pt2, color, thickness)
  - img: 影像物件
  - pt1: 起始點座標
  - pt2: 結束點座標
  - color: 線條顏色，使用 BGR
  - thickness: 線條粗細，預設 1
- Example
  - cv2.line(img,(50,50),(250,250),(0,0,255),5)

OpenCV繪製圖形

- 畫箭頭線條 arrowedLine(img, pt1, pt2, color, thickness, tipLength)
  - img: 影像物件
  - pt1: 起始點座標
  - pt2: 結束點座標
  - color: 線條顏色，使用 BGR
  - thickness: 線條粗細，預設 1
  - tipLength 箭頭長度，預設 0.1 (箭頭線條長度 x 0.1)
- Example
  - cv2.arrowedLine(img,(50,50),(250,250),(0,0,255),5)

OpenCV繪製圖形

- 畫四邊形rectangle(img, pt1, pt2, color, thickness)
  - img:影像物件
  - pt1: 左上座標
  - pt2: 右下座標
  - color: 線條顏色，使用 BGR
  - thickness: 線條粗細，預設 1，設定 -1 表示填滿
- Example
  - cv2.rectangle(img,(50,50),(250,250),(0,0,255),5)

OpenCV繪製圖形

- 畫圓形circle(img, center, radius, color, thickness)
  - img:影像物件
  - center: 中心點座標
  - radius: 半徑
  - color: 線條顏色，使用 BGR
  - thickness: 線條粗細，預設 1，設定 -1 表示填滿
- Example
  - cv2.circle(img,(150,150),100,(0,0,255),5)

OpenCV繪製圖形

- 畫橢圓形ellipse(img, center, axes, angle, startAngle, endAngle, color, thickness)
  - img:影像物件
  - center: 中心點座標
  - axes: 長軸與短軸
  - angle: 轉向角度，正值逆時針，負值順時針
  - startAngle: 起始角度，範圍 0 ~ 360
  - endAngle: 結束角度，範圍 0 ~ 360
  - color: 線條顏色，使用 BGR
  - thickness: 線條粗細，預設 1，設定 -1 表示填滿
- Example
  - cv2.ellipse(img,(150,150),(100,50),45,0,360,(0,0,255),5)

OpenCV繪製圖形

- 畫多邊形polylines(img, pts, isClosed, color, thickness)
  - img:影像物件
  - pts: 座標陣列 (使用 numpy 陣列)
  - isClosed: 多邊形是否閉合，True 閉合，False 不閉合
  - color: 線條顏色，使用 BGR
  - thickness: 線條粗細，預設 1
- Example
  - pts = np.array([[150,50],[250,100],[150,250],[50,100]])
  - cv2.polylines(img,[pts],True,(0,0,255),5)

OpenCV影像加入文字

- putText(img, text, org, fontFace, fontScale, color, thickness, lineType)
  - img影像物件
  - text 文字內容
  - org 文字座標 (垂直方向是文字底部到影像頂端的距離)
  - fontFace 文字字型
  - fontScale 文字尺寸
  - color 線條顏色，使用 BGR
  - thickness 文字外框線條粗細，預設 1
  - lineType 外框線條樣式，預設 cv2.LINE_8，設定 cv2.LINE_AA 可以反鋸齒

OpenCV影像加入文字

- Example
  - text = 'Hello'
  - org = (20,90)
  - fontFace = cv2.FONT_HERSHEY_SIMPLEX
  - fontScale = 2.5
  - color = (0,0,255)
  - thickness = 5
  - lineType = cv2.LINE_AA
  - cv2.putText(img, text, org, fontFace, fontScale, color, thickness, lineType)

OpenCV色彩轉換

- cv2.cvtColor(imgobj, code)
  - 第一個參數為影像物件
  - 第二個參數為轉換的色彩模式

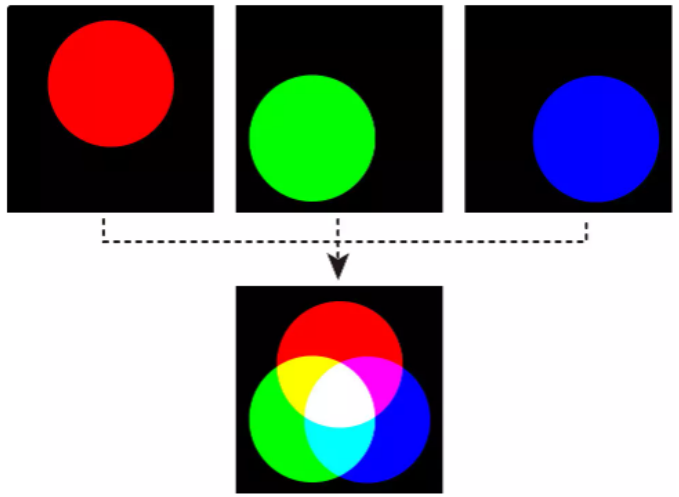
模式代碼	說明
cv2.COLOR_BGR2BGRA	RGB 轉 RGBA。
cv2.COLOR_BGRA2BGR	RGBA 轉 RGB。
cv2.COLOR_BGR2GRAY	RGB 轉灰階。
cv2.COLOR_BGR2HSV	RGB 轉 HSV。
cv2.COLOR_RGB2HLS	RGB 轉 RSL。

OpenCV剪裁影像

- 在 OpenCV 裡讀取的影像，實質上是 NumPy 的陣列
- 使用陣列的切片方式，取出想要的範圍
- Example
  - img[100:300, 100:300]

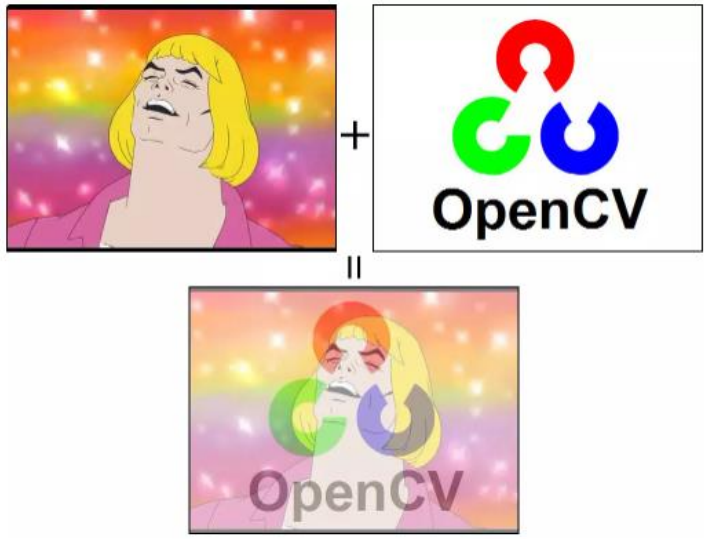
OpenCV影像疊加

- 影像疊加add(img1, img2)
  - 將不同的影像中，同樣位置像素的顏色數值相加
  - 傳入參數為兩個影像物件
- Example
  - img_red = cv2.imread('test-red.png')
  - img_green = cv2.imread('test-green.png')
  - cv2.add(img_red, img_green)



OpenCV影像疊加

- 影像權重疊加addWeighted(img1, alpha, img2, beta, gamma)
  - img1 第一個影像物件
  - img2 第二個影像物件
  - 計算公式：img1*alpha + img2*beta + gamma
- Example
  - img = cv2.imread('meme.jpg')
  - logo = cv2.imread('opencv-logo.jpg')
  - output = cv2.addWeighted(img, 0.5, logo, 0.3, 50)



OpenCV影像相減

- 影像相減subtract(img1, img2)
  - 將不同的影像中，同樣位置像素的顏色數值相減
  - 傳入參數為兩個影像物件
- Example
  - img1 = cv2.imread('test.png')
  - img2 = cv2.imread('test2.png')
  - output = cv2.subtract(img1, img2)

白色 – 黃色
(255,255,255) – (0,255,255)

綠色 – 黃色
(0,255,0) – (0,255,255)

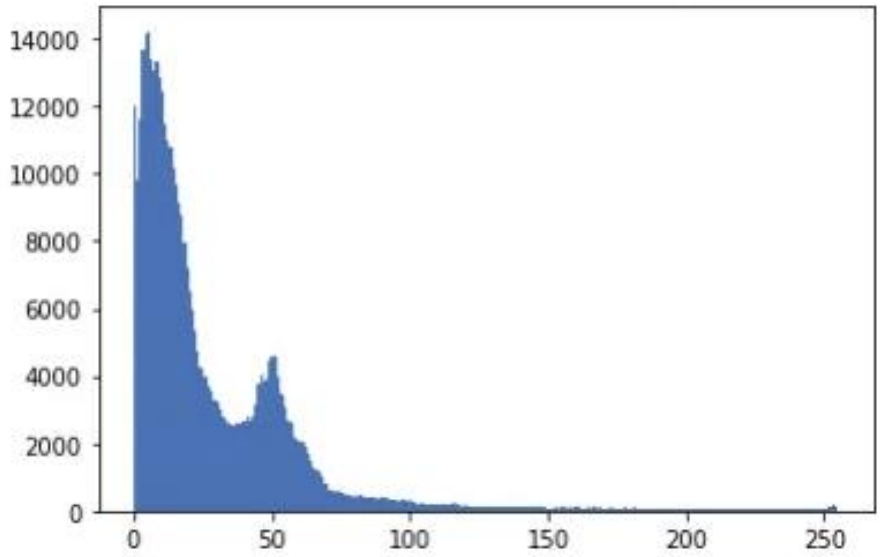
練習1

- 讀取image資料夾內的20張image
- 將影像做以下操作
  - 第1-5張在影像上繪製任意線條
  - 第6-10張加入NUTC的文字(任意大小)
  - 第11-15張做影像權重疊加
    - 將原圖和NUTC logo的size調整成一樣(512*512)
    - 將原圖和NUTC logo疊加(alpha, beta, gamma自訂)
  - 第16-20張做影像相減
    - 將原圖和NUTC logo的size調整成一樣(512*512)
    - 將原圖和NUTC logo相減
- 儲存處理過後的影像

影像增強

影像直方圖 Histogram

- 影像直方圖（Image histogram）是一種用來表現數位影像中像素分布的直方圖
- 例如根據統計影像中不同亮度的像素總數，我們可以畫出一張代表這張影像的影像直方圖



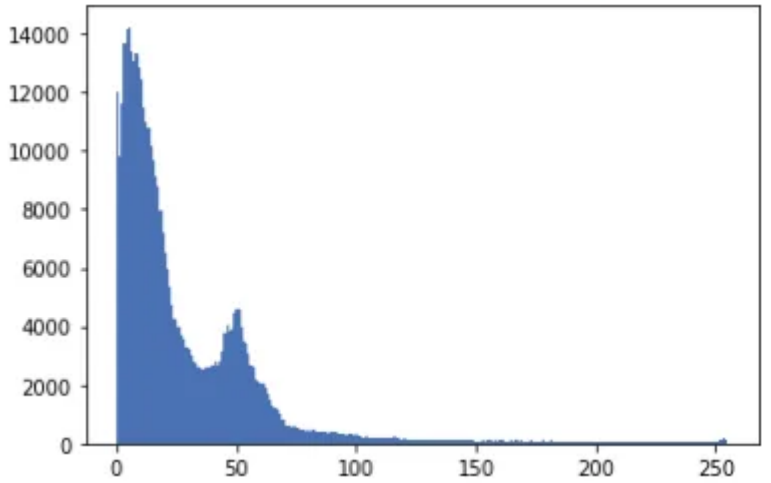
影像亮度

- 影像中的每個像素值的大小即代表亮度
- 統計整張影像中的每個像素的亮度值並表示在直方圖上
- 若直方圖中整體影像亮度值偏小，則亮度較低，反之則亮度較高

影像亮度

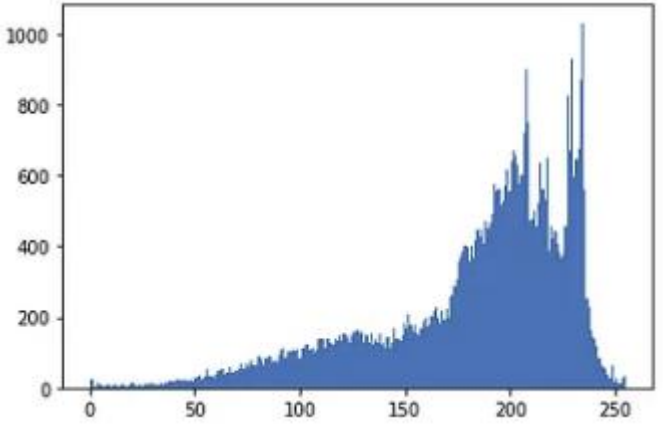
亮度較低

直方圖中的影像亮度值整體偏低



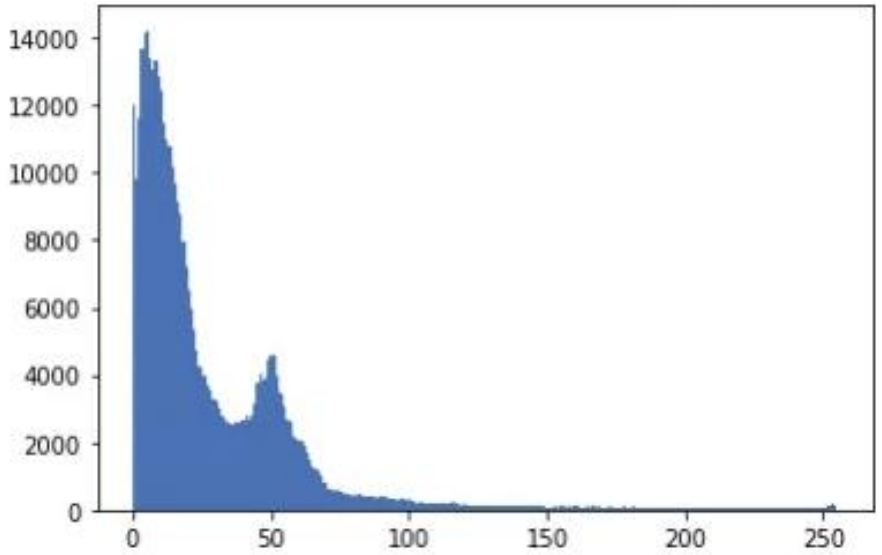
亮度較高

直方圖中的影像亮度值整體偏高



影像的對比度

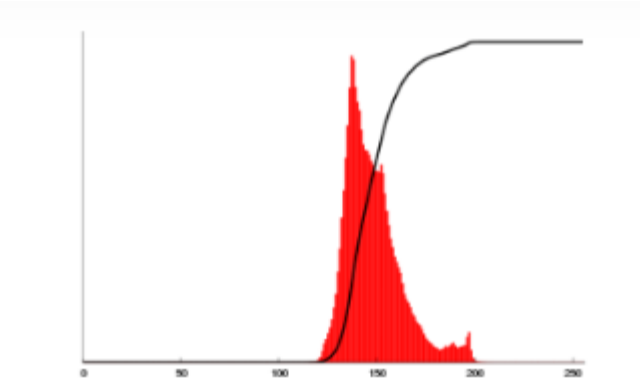
- 對比度為在給定的解析度下物體上黑色與白色的區分程度
- 對比度越高時，畫面中亮暗差距就越明顯
- 在灰階影像直方圖中，如果像素亮度分佈的峰值非常集中在一起，則明暗的對比就不太明顯



影像對比度

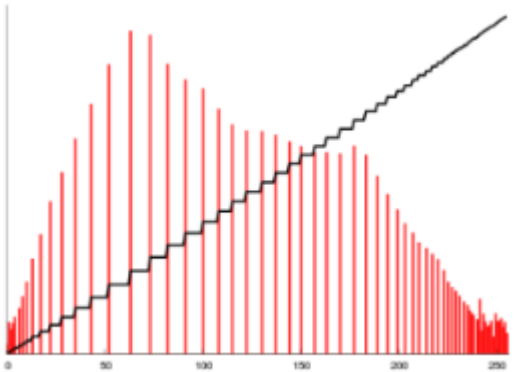
對比度較低

直方圖中的影像亮度值非常集中



對比度較高

直方圖中的影像亮度值較為分散



OopenCV調整影像的對比和亮度

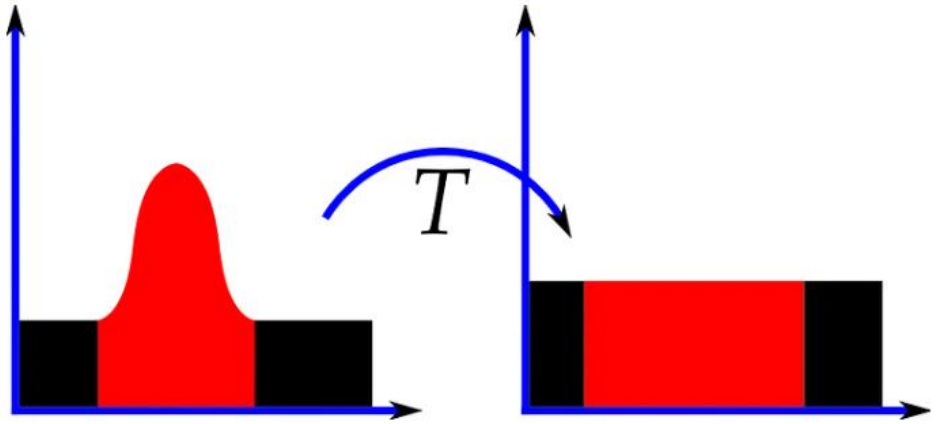
- 透過 NumPy「陣列廣播」的功能，就能迅速更改圖片中每個像素的顏色
  - array+2
- 使用轉換公式去調整 contrast (對比) 和 brightness (亮度) 的數值，就能改變影像的對比度和亮度
  - img * (contrast/127 + 1) - contrast + brightness # 轉換公式
- Example
  - contrast = 200
  - brightness = 0
  - output = img * (contrast/127 + 1) - contrast + brightness

調整影像的對比和亮度

- 調整後的數值大多為浮點數，且可能會小於 0 或大於 255
- 為了保持像素色彩區間為 0 ~ 255 的整數，使用 np.clip() 和 np.uint8() 進行轉換
  - np.clip(array, min, max) : 將array裡的數值限制在min~max之間
  - np.uint8(array) : 將array數值轉成8位元的整數值
- Example:
  - Img_out= np.clip(img, 0, 255)
  - Img_out = np.uint8(Img_out)

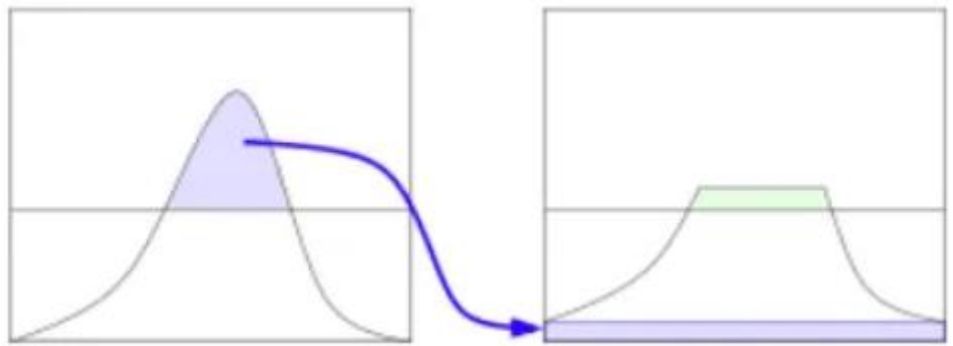
調整影像的對比和亮度

- 直方圖均衡化 Histogram Equalization
  - 通過拉伸影像的像素強度分佈範圍來增強圖像對比度，適用於過曝或背光的圖片
  - 運用累積分布函數(CDF)對灰度值進行調整以實現對比度增強,把原始圖像的灰度直方圖從比較集中的某個灰度區間變成在全部灰度範圍內的均勻分佈
  - cv2.equalizeHist(gray_img)



調整影像的對比和亮度

- 限制對比度自適應直方圖均衡 Contrast Limited Adaptive Histogram Equalization (CLAHE)
  - 對直方圖進行裁剪，使其幅值低於某個上限
  - 被修剪掉的部分不能扔掉，需要將其重新均勻的分佈到直方圖中，生成新的直方圖，以確保直方圖總面積不變
  - clahe = cv2.createCLAHE()
  - clahe_img = clahe.apply(gray_img)



OpenCV加強影像

- convertScaleAbs(img, output, alpha, beta)
  - 根據特定的公式，轉換影像中每個像素，達到影像增強的效果
  - Img: 影像物件
  - output: 輸出影像物件
  - 轉換公式：output = img*alpha + beta, alpha, beta 為公式中的參數

練習2

- 讀取image資料夾內的20張image
- 並將影像做以下操作
  - 第1-5張將亮度調高50
  - 第6-10張將對比度調高100
  - 第11-15張先將影像都轉成灰階然後做灰階均衡化(CLAHE)
  - 第16-20張做加強影像(alpha, beta自訂)
- 儲存處理過後的影像