多媒體程式設計
影像資料處理

Instructor: 馬豪尚

OpenCV

- OpenCV 全名是 Open Source Computer Vision Library (開源計算機視覺函式庫)，可以在商業和研究領域中免費使用，是目前發展最完整的電腦視覺開源資源
- OpenCV 是一個跨平台的電腦視覺函式庫，可應用於臉部辨識、手勢辨識、圖像分割...等影像辨識相關的領域
- OpenCV安裝
  - pip install opencv-python
- 載入函式庫
  - Import cv2

OpenCV開啟圖片

- imread('image', mode)
  - 第一個參數為檔案的路徑和名稱
  - 第二個參數可不填表示以何種模式 (mode) 開啟圖片
  - 開啟的圖片支援常見的 jpg、png...等格式

- Example
  - imgobj = cv2.imread('lena.jpg')
  - imgobj = cv2.imread('lena.jpg', cv2.IMREAD_GRAYSCALE)

imread模式

模式	說明
cv2.IMREAD_UNCHANGED	原本的圖像 (如果圖像有 alpha 通道則會包含)。
cv2.IMREAD_GRAYSCALE	灰階圖像。
cv2.IMREAD_COLOR	BGR 彩色圖像。
cv2.IMREAD_ANYDEPTH	具有對應的深度時返回 16/32 位元圖像，否則將其轉換為 8 位元圖像。
cv2.IMREAD_ANYCOLOR	以任何可能的顏色格式讀取圖像。
cv2.IMREAD_LOAD_GDAL	使用 gdal 驅動程式加載圖像。
cv2.IMREAD_REDUCED_GRAYSCALE_2	灰階圖像，圖像尺寸減小 1/2。
cv2.IMREAD_REDUCED_COLOR_2	BGR 彩色圖像，圖像尺寸減小 1/2。
cv2.IMREAD_REDUCED_GRAYSCALE_4	灰階圖像，圖像尺寸縮小 1/4。
cv2.IMREAD_REDUCED_COLOR_4	BGR 彩色圖像，圖像尺寸減小 1/4。
cv2.IMREAD_REDUCED_GRAYSCALE_8	灰階圖像，圖像尺寸縮小 1/8。
cv2.IMREAD_REDUCED_COLOR_8	BGR 彩色圖像，圖像尺寸減小 1/8。
cv2.IMREAD_IGNORE_ORIENTATION	不要根據 EXIF 資訊的方向標誌旋轉圖像。

OpenCV 顯示圖片

- imshow('window', imgobj)
  - 會開一個新視窗來顯示圖片
  - 第一個參數為字串，表示要開啟圖片的視窗名稱
  - 第二個參數為使用 imread() 讀取的圖片物件
- Note: Google Colab不能使用這個函式，不允許開新視窗，以下為在Colab裡面使用imshow的方法
  - from google.colab.patches import cv2_imshow
  - cv2_imshow(imgobj)

OpenCV 關閉視窗

- waitKey() 等待多久關閉顯示視窗
  - 表示等待與讀取使用者按下的按鍵，包含一個單位為「毫秒」的參數
  - 例如: cv2.wait(2000)
- destroyWindow(name)
  - 關閉指定名稱的視窗
- destroyAllWindows()
  - 關閉所有視窗

OpenCV 寫入圖片

- imwrite('img_Name', imgobj, ImwriteFlags)
  - 第一個參數為檔案的路徑和名稱
  - 第二個參數為要寫入的資料內容
  - 第三個參數ImwriteFlags為圖片壓縮品質的設定
- Example
  - cv2.imwrite('lena.jpg', imgobj, [cv2.IMWRITE_JPEG_QUALITY, 80]) # 存成 jpg
  - cv2.imwrite('lena.png', img)

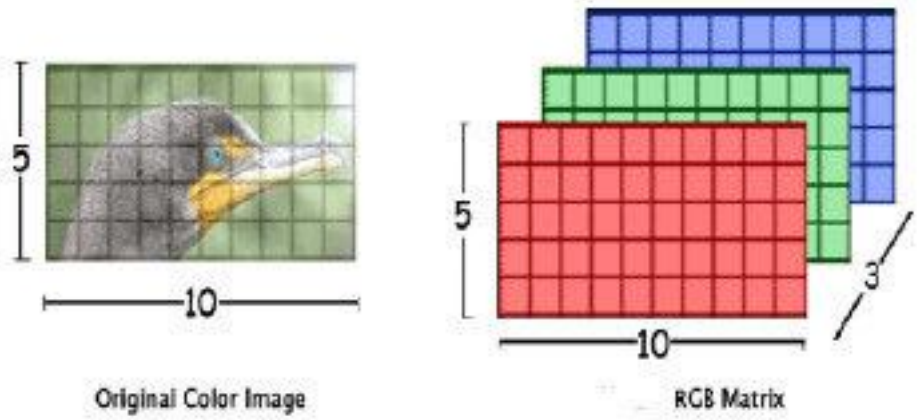
OpenCV 寫入圖片- 常見壓縮和品質

imwirte()第三個參數
Example: [cv2.IMWRITE_JPEG_QUALITY, 80]

Enumerator	
cv2.IMWRITE_JPEG_QUALITY	JPEG壓縮，它可以是從 0 到 100 的質量（越高越好）。默認值為 95。
cv2.IMWRITE_PNG_COMPRESSION	PNG壓縮，它可以是從0到9的壓縮級別。更高的值代表著更小的尺寸和更長的壓縮時間
cv2.IMWRITE_PNG_BILEVEL	二進制級別 PNG，0 或 1，默認為 0
cv2.IMWRITE_HDR_COMPRESSION	HDR 壓縮
cv2.IMWRITE_TIFF_COMPRESSION	TIFF壓縮，用於指定圖像壓縮方案。有關與壓縮格式對應的整數常量，請參閱 libtiff 文檔
cv2.IMWRITE_JPEG2000_COMPRESSION_X1000	JPEG2000壓縮，用於指定目標壓縮率（乘以 1000）。該值可以從 0 到 1000。默認值為 1000。

OpenCV儲存陣列產生的圖片

- 在 Python 裡，圖片可以使用「三維陣列」的方式表現
  - 長寬各多少個像素、每個像素裡包含的顏色資訊是什麼
- 讀進來的物件其實是用array來表示這個三維陣列
  - Imgobj = cv2.imread('lena.jpg')
- 自行產生三維陣列
  - Img_array = np.zeros((500,300,3), dtype='uint8')
- 搭配imwrite()寫入
  - cv2.imwrite('img.jpg', Img_array)

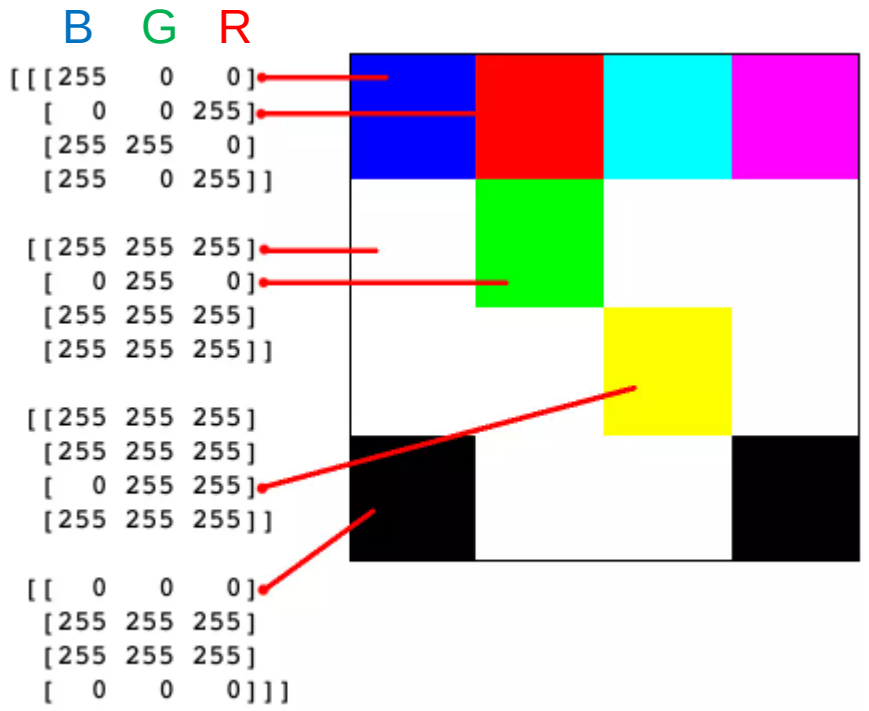


OpenCV取得影像資訊

- Imgobj.shape
  - 取得長寬與色版數量
  - 如果影像不具有三個色版(RGB)，則只會取得寬與長
- Imgobj.size
  - 取得像素總數
  - 像素總數為「寬 x 長 x 色版數量」
- Imgobj.dtype
  - 取得影像的數據類型

取得影像像素的色彩資訊

- 可以印出圖片的「三維陣列」資訊
  - print(imgobj)
    - 看到每一個像素都有 B、G、R 三個顏色資訊，顏色範圍均是 0～255



OpenCV改變影像大小

- rsize(imgobj, size, interpolation)
  - 第一個參數為影像物件
  - 第二個參數為要改變的目標大小
    - 用(寬, 高)來設定，例如:(100, 200)
  - 第三個參數為改變插值方式
    - 可以不輸入這個參數
    - 預設使用 INTER_LINEAR

OpenCV影像翻轉

- cv2.flip(imgobj, axis)
  - 第一個參數為影像物件
  - 第二個參數為翻轉的參考軸
    - 0 → 以 x 軸為中心上下翻轉。
    - 1 → 以 y 軸為中心左右翻轉。
    - -1 → 同時進行上下左右翻轉。

OpenCV影像旋轉

- 旋轉影像 cv2.transpose(imgobj)
  - 可以將影像「逆時針」旋轉 90 度

- 旋轉影像 cv2.rotate(imgobj, rotate_state)
  - 這個方法可以在第二個參數rotate_state使用以下的值設定逆時針旋轉 90 度、順時針旋轉 90 度，以及旋轉 180 度。
    - cv2.ROTATE_90_CLOCKWISE
    - cv2.ROTATE_90_COUNTERCLOCKWISE
    - cv2.ROTATE_180

OpenCV Resize插值方式

Enumerator	
cv2.INTER_NEAREST	最近鄰插值
cv2.INTER_LINEAR	雙線性插值
cv2.INTER_CUBIC	雙三次插值
cv2.INTER_AREA	使用像素面積關係重採樣。它可能是圖像抽取的首選方法，因為它可以提供無摩爾紋的結果。但是當圖像被縮放時，它類似於 INTER_NEAREST 方法。
cv2.INTER_LANCZOS4	8x8 鄰域上的 Lanczos 插值
cv2.INTER_LINEAR_EXACT	位精確雙線性插值
cv2.INTER_NEAREST_EXACT	位精確最近鄰插值。這將產生與 PIL、scikit-image 或 Matlab 中的最近鄰方法相同的結果。

OpenCV影像的幾何變形

- 影像仿射轉換 cv2.warpAffine(img, M, (w, h))
  - 可以將來源的圖像，根據指定的「仿射矩陣」，輸出成仿射轉換後的新影像
  - 仿射矩陣必須採用 numpy 的矩陣格式且採用2x3的矩陣大小
  - (w, h)為指定影像大小寬高
- 仿射矩陣範例(水平位移)
  - M = np.float32([[1, 0, 100], [0, 1, 50]])
  - 2x3 矩陣，x 軸平移 100，y 軸平移 50

OpenCV影像的幾何變形

- getRotationMatrix2D((x, y), angle, scale)
  - 可以產生旋轉指定角度影像的仿射矩陣
  - 第一個參數為旋轉的中心點座標
  - 第二個參數為旋轉角度(- 為順時針，+為逆時針)
  - 第三個參數為旋轉後的影像尺寸
- Example
  - cv2.getRotationMatrix2D((240, 180), 45, 1)
    - 中心點 (240, 180)，逆時針旋轉 45 度，尺寸 1

OpenCV圖像仿射變換

- getAffineTransform(輸入影像三個點的座標，輸出影像三個點的座標)
  - 根據輸入影像的三個點，對應輸出影像的三個點，產生仿射矩陣
  - 使用 2x3 的 numpy 矩陣作為三個點的座標格式
- Example
  - p1 = np.float32([[100,100],[480,0],[0,360]])
  - p2 = np.float32([[0,0],[480,0],[0,360]])
  - cv2.getAffineTransform(p1, p2)

練習

- 用opencv套件完成
- 讀取image資料夾內的20張image
- 將所有image的size調整為512*512
- 將影像做以下操作
  - 第1-5張做上下翻轉
  - 第6-10張做順時針旋轉90度
  - 第11-15張做順時針旋轉45度
  - 第16-20張做仿射變換
    - 將圖片座標中(0, 0)轉換到(50, 50)
    - 將圖片座標中(512, 0)轉換到(512, 10)
    - 將圖片座標中(512, 512)轉換到(512, 512)
- 儲存處理過後的影像