



Python OPEN CV

1.1 OpenCV 讀取及顯示圖形

- 安裝OpenCV
 - `Pip install -v opencv-python==4.2.0.34`
- 匯入OpenCV
 - `Import cv2`

1.1 OpenCV 讀取及顯示圖形

- 建立視窗顯示影像
 - `Cv2.namedWindow(視窗名稱, 視窗旗標)`
- 視窗旗標
 - `Cv2.WINDOW_NORMAL` – Allows to manually change window size
 - `Cv2.WINDOW_AUTOSIZE(Default)` – Automatically sets the window size
 - `Cv2.WINDOW_FULLSCREEN` – Changes the window size to fullscreen
 - `Cv2.WINDOW_KEEPRATIO` – Keep the ratio of image

1.1 OpenCV 讀取及顯示圖形

- 讀取影像
 - 影像變數 = `cv2.imread(影像路經, 讀取旗標)`
- 讀取旗標(數值)
 - `Cv2.IMREAD_COLOR(Default)` : 1
 - `Cv2.IMREAD_GRAYSCALE` : 0(灰階)
 - `Cv2.IMREAD_UNCHANGE` : -1
 - 例 : `Img=imread(path,0)`

1.1 OpenCV 讀取及顯示圖形

- 視窗中顯示影像
 - `cv2.imshow(視窗名稱, 影像變數)`
- 視窗中顯示影像時間
 - `cv.waitKey(n)`
 - 單位：毫秒。
 - 0：無限長，案任何鍵結束

1.1 OpenCV 讀取及顯示圖形

- 關閉單一視窗影像
 - `Cv2.destroyWindow(視窗名稱)`
- 關閉所有視窗影像
 - `Cv2.destroyAllWindows()`

1.1 OpenCV 讀取及顯示圖形

- 程式說明及練習

1.2 OpenCV 讀取儲存圖形

- 儲存影像

- `Cv2.imwrite (存檔路徑, 影像變數, [int(存檔旗標)], 值)`

- 存檔旗標

- `cv2.IMWRITE_JPEG_QUALITY` : 存jpeg, jpg格式, 其值0~100(高品質)
 - `cv2.IMWRITE_WEBP_QUALITY` : 存webp格式, 其值0~100(高品質)
 - `cv2.IMWRITE_PNG_COMPRESSION` : 設定png格式壓縮比0~9, 預設3。

1.2 OpenCV 讀取儲存圖形

- 程式說明及練習

1.3 OpenCV 基本繪圖

- 畫直線
 - `Cv2.line(畫布, 起始點, 結束點, 顏色, 寬度)`
 - 顏色 0~255
- 例
 - `cv2.line(image, (20,60), (300,400), (0,0,255),2)`

1.3 OpenCV 基本繪圖

- 畫矩形
 - `Cv2.rectangle(畫布, 起始點, 結束點, 顏色, 寬度)`
 - 寬度：`>0` 繪圖寬度, `<0` 實心矩形
- 例
 - `cv2.rectangle(image, (20,60), (300,400), (255,0,0),-1)`

1.3 OpenCV 基本繪圖

- 畫圓形
 - `Cv2.circle(畫布, 圓心, 半徑, 顏色, 寬度)`
 - 寬度： >0 繪圖寬度, <0 實心圓形
- 例
 - `cv2.circle(image, (300,300),40, (0,255,0),2)`

1.3 OpenCV 基本繪圖

- 畫多邊形
 - `Cv2.polylines`(畫布, 點座標串列, 封閉, 顏色, 寬度)
 - 寬度 : >0 繪圖寬度, <0 實心圓形
 - 封閉 : `True` / 開口 : `False`
- 點座標串列
 - `Import numpy`
 - `Numpy.array([[第一個座標], [第一個座標],.....])`

1.3 OpenCV 基本繪圖

- 畫多邊形範例
 - `Pts=numpy.array([[20,60],[300,600],[150,200]], numpy, int32)`
 - `Cv2.polylines(畫布, [pts], True, (0,0,255) 2)`

1.3 OpenCV 基本繪圖

- 文字
 - `Cv2.putText(畫布, 文字, 位置, 字體, 字體尺寸, 顏色, 文字粗細)`
- 字體尺寸
 - `cv2.FONT_HERSHEY_SIMPLEX`

1.3 OpenCV 基本繪圖

- 程式說明及練習

1.4 OpenCV 臉部偵測

- 偵測CsacadeClassifier物件
 - 偵測物件變數=cv2.CascadeClassifier(偵測特徵檔路徑)
 - OpenCV提供 cv2.data.harcascades方法取得偵測特徵檔路徑。
 - 偵測特徵檔：haarcascade_frontalface_default.xml
- faceCascade=cv2. CascadeClassifier(cv2.data.harcascades + “haarcascade_frontalface_default.xml”)

1.4 OpenCV 臉部偵測

- 偵測物件方法
 - 偵測物件detecMultiScale方式
 - 偵測結果變數=偵測物件變數. detecMultiScale(圖片,參數1,參數2,.....)
- 參數
 - scaleFactor :
 - minNeighbors
 - minSize
 - maxSize
 - flags

1.4 OpenCV 臉部偵測

- `detectMultiScale` 參數
 - `ScaleFactor` : 每次搜尋方塊減少比例
 - `minNeighbors` : 每個目標至少檢測到幾次以上, 才可被認定是真數據。
 - `minSize` : 設定數據搜尋的最小尺寸, 如 `minSize=(40,40)`
 - `maxSize` : 設定數據搜尋的最大尺寸
 - `Flags` :

1.4 OpenCV 臉部偵測

- 程式說明及練習

1.5 擷取臉部圖形及存檔

- 擷取臉部， 指定範圍
 - 圖片變數[左上角y座標：右下角y座標,左上角x座標：右下角x座標]
- 存新變數
 - `Newimg=img[40:100,50:250]`
- 重新定義尺寸(讓不同圖片獲得一樣尺寸)

1.1 建立和輸出字串

Sample

1. Str1="ESP32影像辨識"
2. Name1=str("湯姆克魯斯")
3. print('Str1 = ' + Str1)
4. Print(Name1)
5. Print(Name1 + '參加' + Str1)

1.1 建立和輸出字串

- 常見轉義字元：
- 常見轉義字元 呈現結果
- `\'` 單引號'
- `\"` 雙引號"
- `\n` 換行, `\n`後的内容會換到下一行
- `\t` 定位空格, 一個tab鍵的空格
- `\\` 反斜線\

1-1.2 取出字元和走訪字串

- 走訪字串的每一個字元

1. Str1="ESP32影像辨識"
2. For ch in Str1:
 1. Print(ch, end=' ')

1-1.2 取出字元和走訪字串

- 使用索引運算子取得指定字元

- Str1="ESP32影像辨識"

- Print(Str1[0])

- Print(Str1[4])

- Print(Str1[-3])

- Print(Str1[-7])

0	1	2	3	4	5	6	7	8
E	S	P	3	2	影	像	辨	識
-9	-8	-7	-6	-5	-4	-3	-2	-1

1-1.3 字串函數與方法

- 使用索引運算子取得指定字元

- Str1="ESP32影像辨識"

- Print(Str1[0])

- Print(Str1[4])

- Print(Str1[-3])

- Print(Str1[-7])

0	1	2	3	4	5	6	7	8
E	S	P	3	2	影	像	辨	識
-9	-8	-7	-6	-5	-4	-3	-2	-1