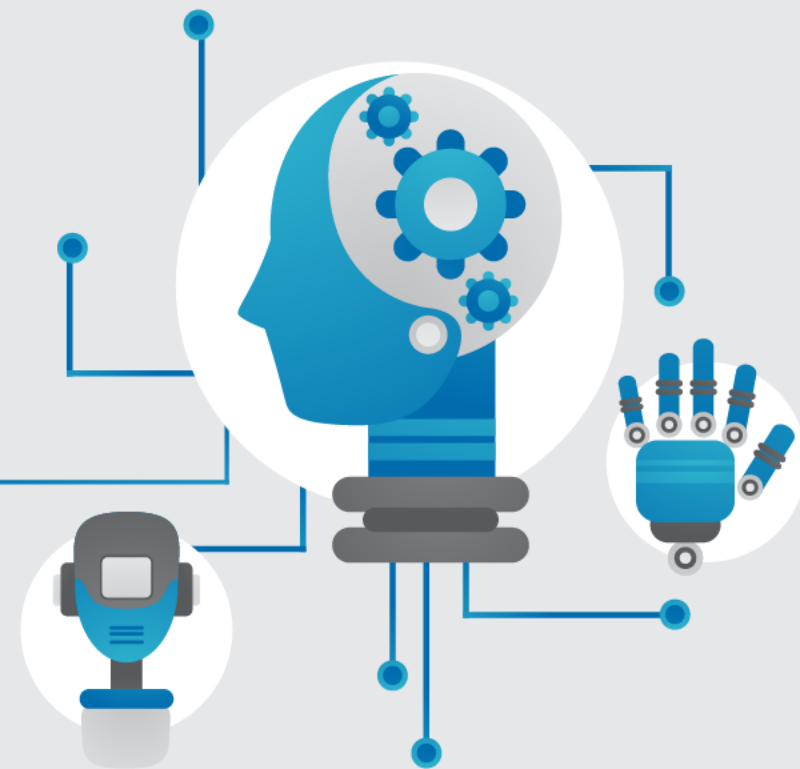


常用的OpenCV 影像處理功能介紹

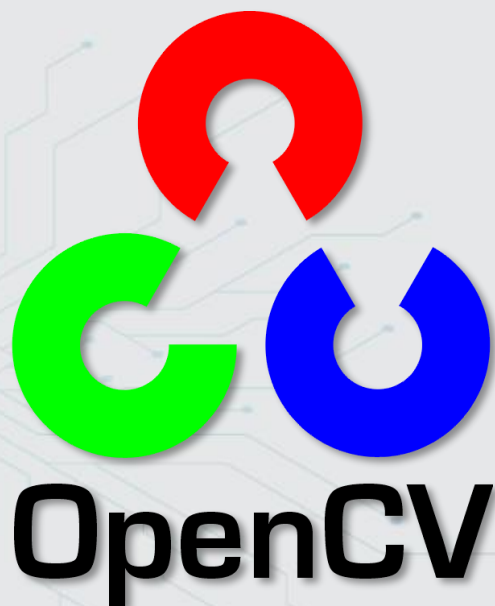




OpenCV



- › OpenCV : **Open** Source **C**omputer **V**ision Library
- › 一個跨平台的電腦視覺庫
- › 由Intel (英特爾) 公司發起並參與開發，以BSD授權條款授權發行，可在商業和研究領域中免費使用。





OpenCV下載與安裝

機器學習實務



官方網站：<https://opencv.org/>

› Python 安裝套件：

```
C:\> pip install opencv-python
```

› Python 程式匯入套件：`import cv2`





讀取圖片



› 指令：cv2.imread([filename], [flag])

› 參數：

filename：檔案路徑 (string)

flag：讀取模式 (int)

› 回傳：影像資料 (numpy.array)

› 說明：常用的flag選項

cv2.IMREAD_GRAYSCALE

cv2.IMREAD_COLOR [Default]

cv2.IMREAD_UNCHANGED



讀取圖片範例

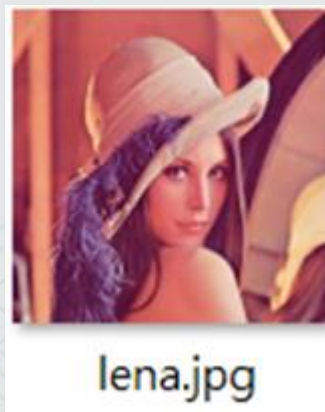


› 範例程式

```
import cv2
```

```
img=cv2.imread('lena.jpg')
```

```
print(img) →
```



USC SIPI Images Database

```
array([[128, 138, 225],
       [127, 137, 224],
       [126, 136, 224],
       ...,
       [126, 145, 236],
       [110, 129, 220],
       [ 86, 104, 197]],

      [[127, 137, 224],
       [127, 137, 224],
       [127, 137, 224],
       ...,
       [130, 150, 235],
       [112, 132, 217],
       [ 83, 102, 189]],

      [[127, 137, 224],
       [127, 137, 224],
       [128, 138, 225],
       ...,
       [127, 148, 230],
       [112, 133, 215]]]
```



儲存圖片



› 指令：imwrite([filename], [image])

› 參數：

filename：儲存檔名路徑 (string)

image：圖片資料 (numpy.array)

› 回傳：是否儲存成功 (boolean)



儲存圖片範例



› 範例程式

```
import cv2
```

```
img=cv2.imread('lena.jpg')
```

```
cv2.imwrite('lena_imwrite.jpg', img)
```



lena_imwrite.jpg



lena.jpg



裁切圖片



- › 運用python slice擷取np.array
- › slice operate : 使用冒號，代表開頭、結尾和步距

```
import numpy as np
```

```
arr=np.array([[1,2,3], [4,5,6]])
```

```
print(arr)
```

```
[[1 2 3]  
 [4 5 6]]
```

```
print(arr[:, 1:3])
```

```
[[ 2 3]  
 [ 5 6]]
```

#0:3:2，代表index 0到3 每次2步

```
print(arr[:, 0:3:2])
```

```
[[ 1 3]  
 [ 4 6]]
```




裁切圖片



› 範例程式

```
import cv2
```

```
img=cv2.imread('lena.jpg')
```

```
#裁切出人物的部分
```

```
crop=img[50:500, 50:420, :]
```

```
cv2.imwrite('lena_imwrite.jpg', crop)
```





轉換顏色



- › 指令：`cvtColor([image], [code])`
- › 參數：
 - `image`：影像資料 (`numpy.array`)
 - `code`：轉換格式 (`int`)
- › 回傳：轉換完的影像資料 (`numpy.array`)
- › 說明：常用的`code`選項
 - `cv2.COLOR_BGR2GRAY`
 - `cv2.COLOR_RGB2BGR`



轉換顏色



› 範例程式

```
import cv2
```

```
img=cv2.imread('lena.jpg')
```

```
img=cv2.cvtColor(img ,cv2.COLOR_BGR2GRAY)
```

```
cv2.imwrite('lena_imwrite.jpg', img)
```





切割通道



- › OpenCV中顏色通道排列為BGR
(png為BGRA, A:alpha透明度)
- › 利用裁切 (slice) 可以切出各通道灰階影像
- › 切出的紅通道，愈紅的地方會愈接近白色，相反愈黑
- › 利用此特性可將雜訊變淡



切割通道



› 範例程式

```
import cv2
```

```
img=cv2.imread('lena.jpg')
```

```
#取出藍通道
```

```
img=img[:, :, 0]
```

```
cv2.imwrite('lena_b.jpg', img)
```

```
#取出紅通道
```

```
img=img[:, :, 2]
```

```
cv2.imwrite('lena_r.jpg', img)
```





縮放



› 指令：`cv2.resize([src], [size], [interpolation])`

› 參數：

`src`：來源影像 (`numpy.array`)

`size`：要縮放的新shape (`tuple`)

`interpolation`：縮放的方式 (`int`)

› 回傳：縮放後的影像資料 (`numpy.array`)

› 說明：常用的interpolation code選項

`cv2.INTER_NEAREST` [最近鄰插值]

`cv2.INTER_LINEAR` [雙線性插值；Default，在品質和時間上取得不錯的平衡]

`cv2.INTER_AREA` [使用像素區域關係進行重採樣；常用於縮小影像]

`cv2.INTER_CUBIC` [4x4像素領域的三次差值；常用於放大影像]

`cv2.INTER_LANCZOS4` [8x8像素領域的Lanczos插值]



縮放



› 範例程式

```
import cv2
```

```
img=cv2.imread('lena.jpg')
```

```
new_image=cv2.resize(img, (100, 50),  
interpolation=cv2.INTER_AREA)
```

```
cv2.imwrite('lena_resize.jpg', new_image)
```

