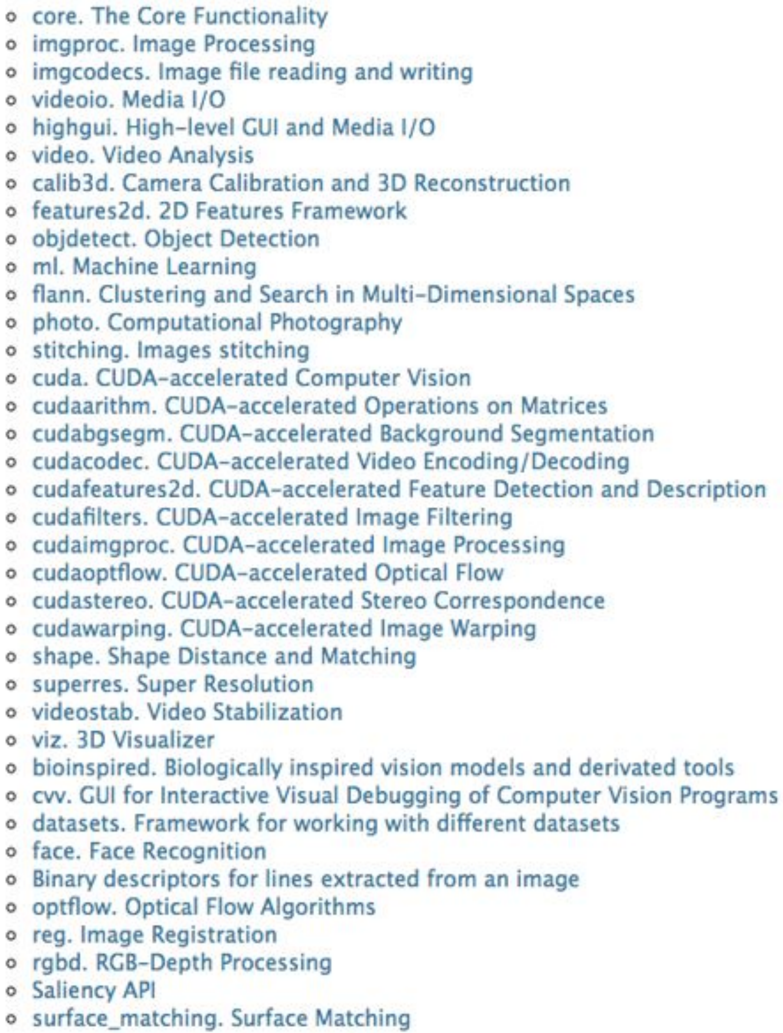
9/10 Lab 01

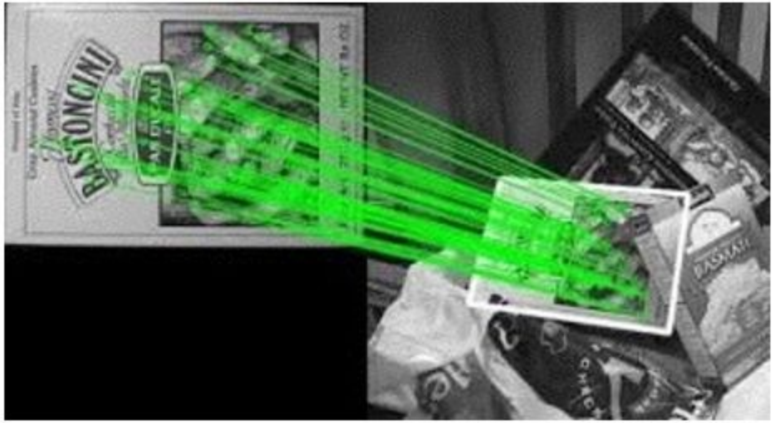
1. OpenCV introduction
2. Python 3 & opencv installation
3. Lab 01

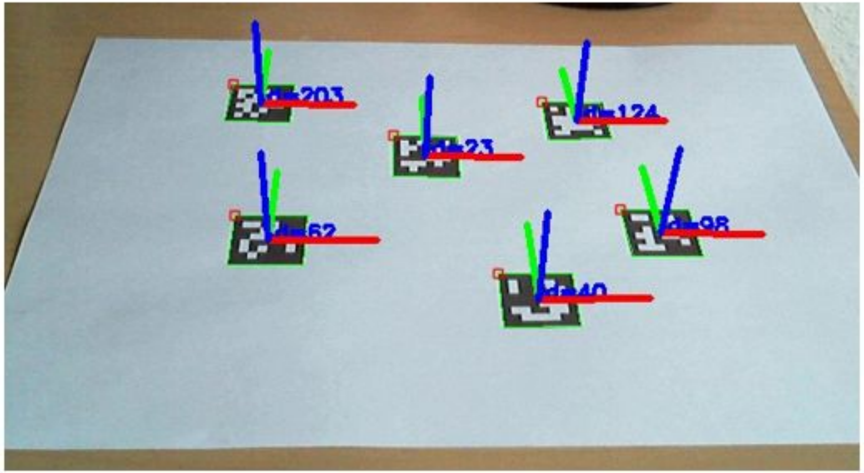
OpenCV



- core. The Core Functionality
- imgproc. Image Processing
- imgcodecs. Image file reading and writing
- videoio. Media I/O
- highgui. High-level GUI and Media I/O
- video. Video Analysis
- calib3d. Camera Calibration and 3D Reconstruction
- features2d. 2D Features Framework
- objdetect. Object Detection
- ml. Machine Learning
- flann. Clustering and Search in Multi-Dimensional Spaces
- photo. Computational Photography
- stitching. Images stitching
- cuda. CUDA-accelerated Computer Vision
- cudaarithm. CUDA-accelerated Operations on Matrices
- cudabgsegm. CUDA-accelerated Background Segmentation
- cudacodec. CUDA-accelerated Video Encoding/Decoding
- cudafeatures2d. CUDA-accelerated Feature Detection and Description
- cudafilters. CUDA-accelerated Image Filtering
- cudaimgproc. CUDA-accelerated Image Processing
- cudaoptflow. CUDA-accelerated Optical Flow
- cudastereo. CUDA-accelerated Stereo Correspondence
- cudawarping. CUDA-accelerated Image Warping
- shape. Shape Distance and Matching
- superres. Super Resolution
- videostab. Video Stabilization
- viz. 3D Visualizer
- bioinspired. Biologically inspired vision models and derivated tools
- cvv. GUI for Interactive Visual Debugging of Computer Vision Programs
- datasets. Framework for working with different datasets
- face. Face Recognition
- Binary descriptors for lines extracted from an image
- optflow. Optical Flow Algorithms
- reg. Image Registration
- rgbd. RGB-Depth Processing
- Saliency API
- surface_matching. Surface Matching

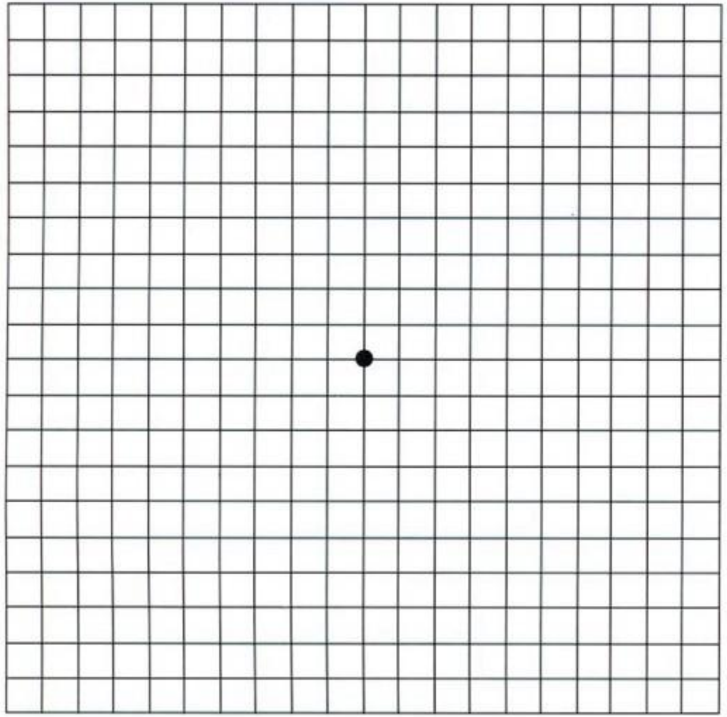
feature detection





pattern recognition

Mat



rows: 長

cols: 寬

type: 像素型態

channels: 通道數

normal:

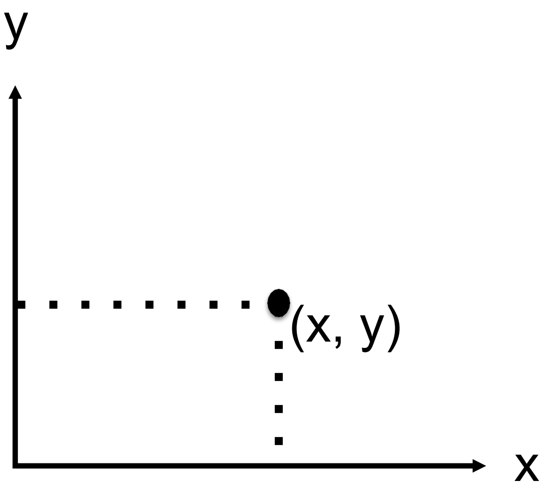
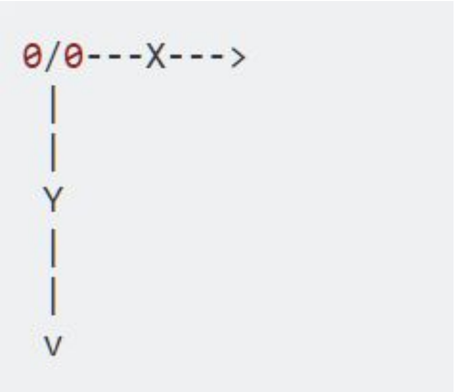
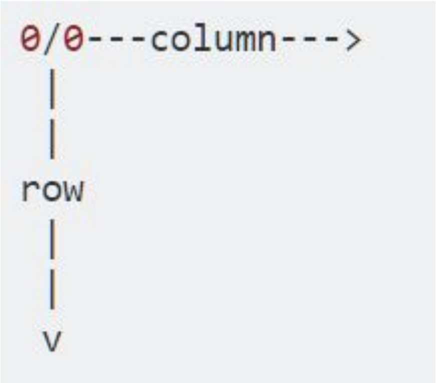


image:

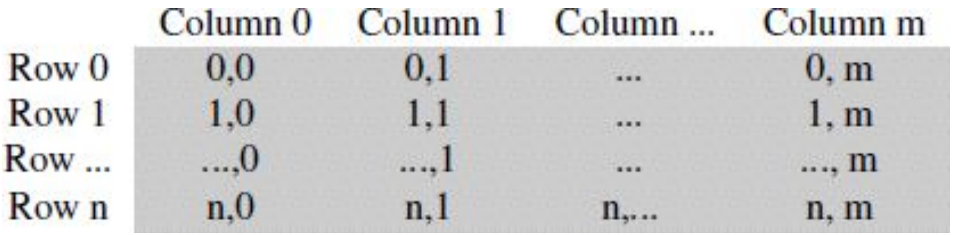


```
0/0---X--->
 |
 |
 Y
 |
 |
 v
```



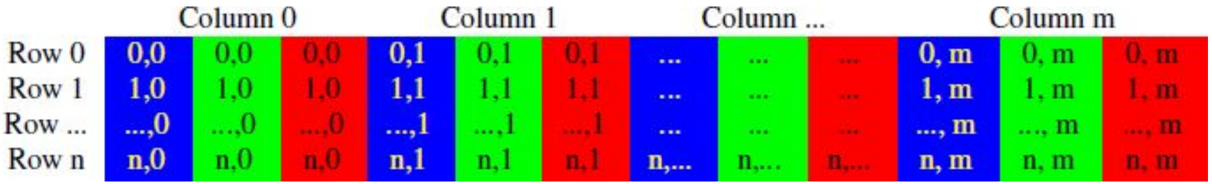
```
0/0---column--->
 |
 |
row
 |
 |
 v
```

Mat value access

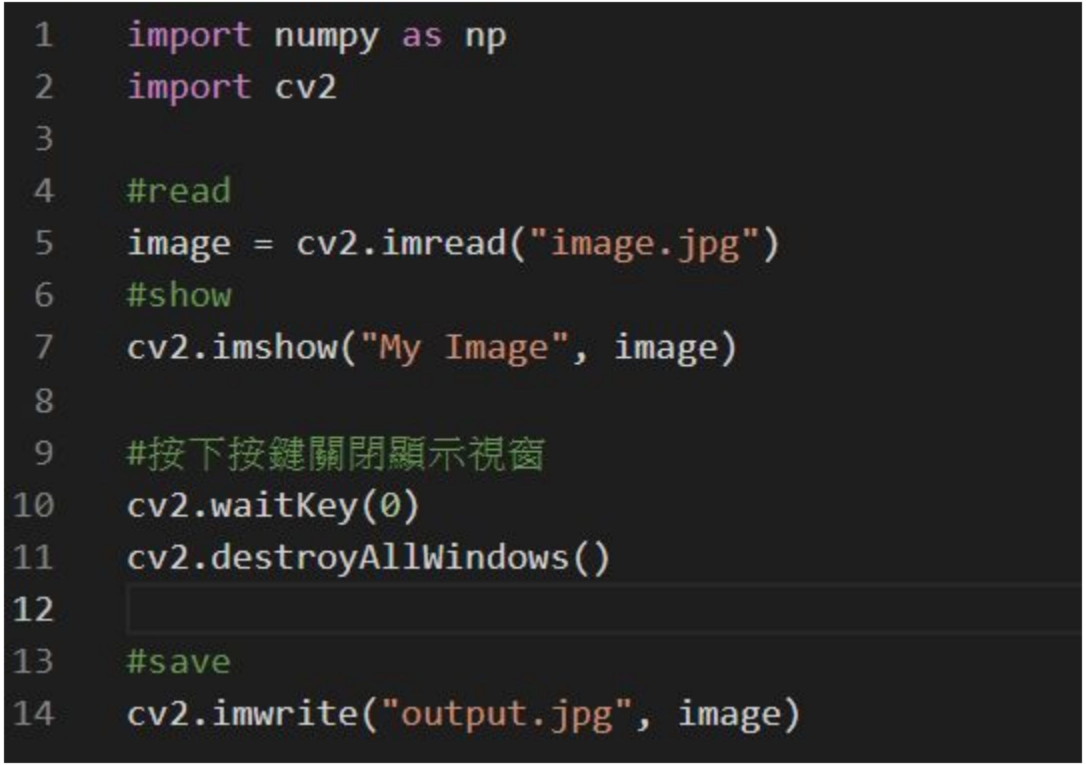


	Column 0	Column 1	Column ...	Column m
Row 0	0,0	0,1	...	0, m
Row 1	1,0	1,1	...	1, m
Row ...	...,0	...,1	...	..., m
Row n	n,0	n,1	n,...	n, m

3-channel : B, G, R



	Column 0			Column 1			Column ...			Column m		
Row 0	0,0	0,0	0,0	0,1	0,1	0,1	...	...	...	0, m	0, m	0, m
Row 1	1,0	1,0	1,0	1,1	1,1	1,1	...	...	...	1, m	1, m	1, m
Row ...	...,0	...,0	...,0	...,1	...,1	...,1	...	...	...	..., m	..., m	..., m
Row n	n,0	n,0	n,0	n,1	n,1	n,1	n,...	n,...	n,...	n, m	n, m	n, m



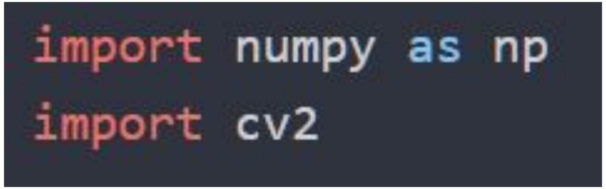
```python
import numpy as np
import cv2

#read
image = cv2.imread("image.jpg")
#show
cv2.imshow("My Image", image)

#按下按鍵關閉顯示視窗
cv2.waitKey(0)
cv2.destroyAllWindows()

#save
cv2.imwrite("output.jpg", image)
```

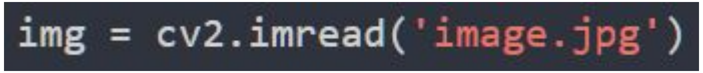
標頭引入



```python
import numpy as np
import cv2
```

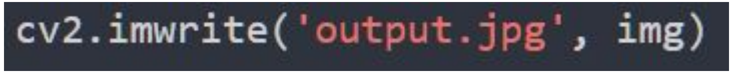
讀寫圖片

讀取:



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

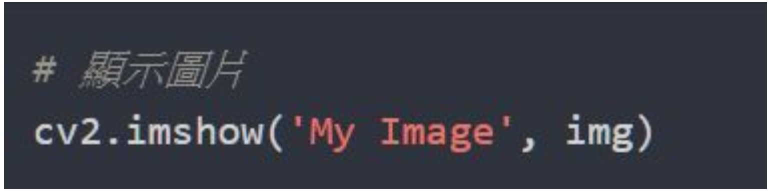
儲存:



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

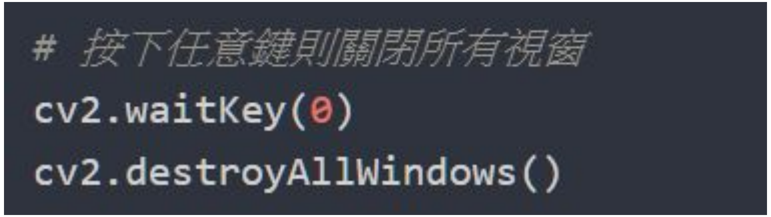
顯示圖片

秀出影像：



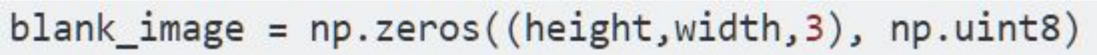
```
# 顯示圖片
cv2.imshow('My Image', img)
```

等待按鍵輸入：



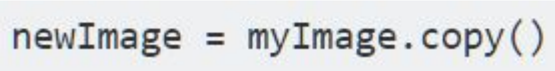
```
# 按下任意鍵則關閉所有視窗
cv2.waitKey(0)
cv2.destroyAllWindows()
```

開一個指定大小的黑圖：



```
blank_image = np.zeros((height,width,3), np.uint8)
```

複製圖片:



```
newImage = myImage.copy()
```

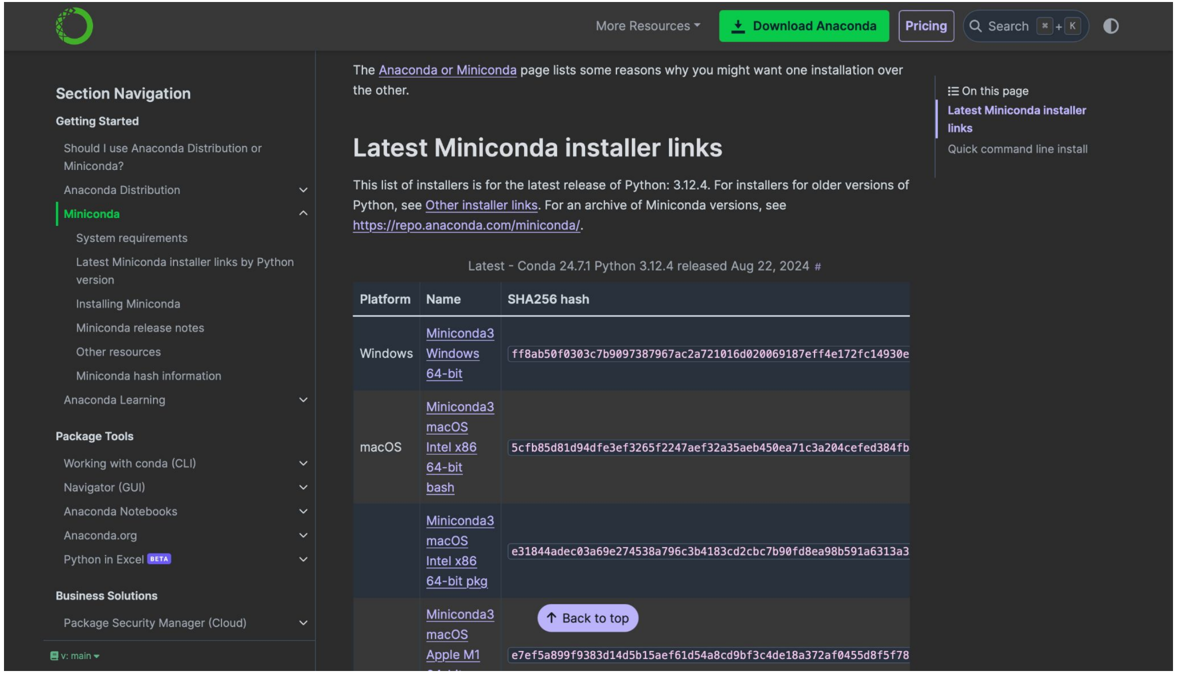
操作像素

image[row, col, channel]

Python 3 & OpenCV Installation

安裝conda

miniconda: https://docs.conda.io/projects/miniconda/en/latest/



The Anaconda or Miniconda page lists some reasons why you might want one installation over the other.

Latest Miniconda installer links

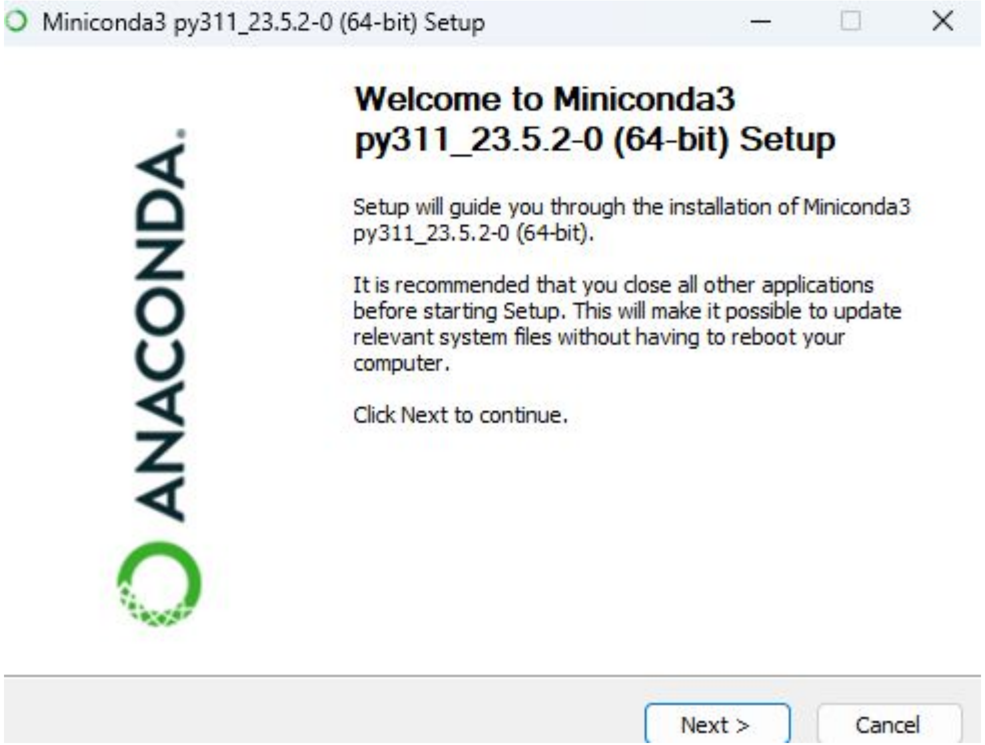
This list of installers is for the latest release of Python: 3.12.4. For installers for older versions of Python, see Other installer links. For an archive of Miniconda versions, see https://repo.anaconda.com/miniconda/.

Latest - Conda 24.7.1 Python 3.12.4 released Aug 22, 2024 #

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	`ff8ab50f0303c7b9097387967ac2a721016d020069187eff4e172fc14930e`
macOS	Miniconda3 macOS Intel x86 64-bit bash	`5cfb85d81d94dfe3ef3265f2247aef32a35aeb450ea71c3a204cefed384fb`
	Miniconda3 macOS Intel x86 64-bit pkg	`e31844adec03a69e274538a796c3b4183cd2cbc7b90fd8ea98b591a6313a3`
	Miniconda3 macOS Apple M1	`e7ef5a899f9383d14d5b15aef61d54a8cd9bf3c4de18a372af0455d8f5f78`

安裝conda

一直按next直到下一頁的畫面



Miniconda3 py311_23.5.2-0 (64-bit) Setup

Welcome to Miniconda3 py311_23.5.2-0 (64-bit) Setup

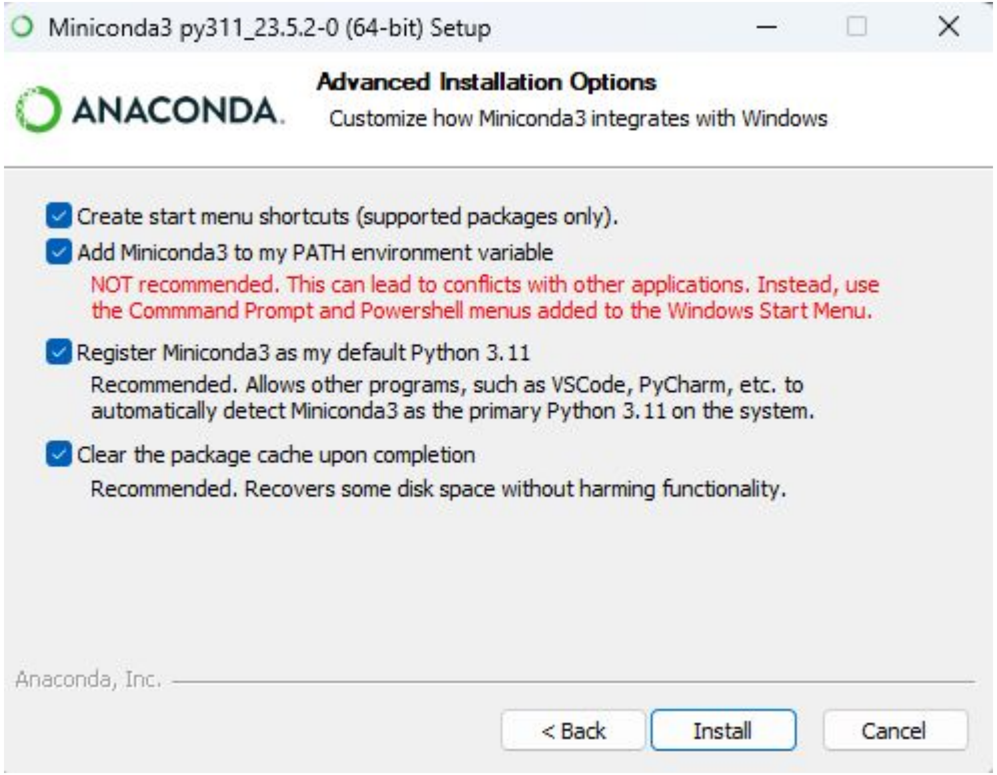
Setup will guide you through the installation of Miniconda3 py311_23.5.2-0 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

安裝conda

這邊建議4個都打勾



Miniconda3 py311_23.5.2-0 (64-bit) Setup

Advanced Installation Options

Customize how Miniconda3 integrates with Windows

☑ Create start menu shortcuts (supported packages only).

☑ Add Miniconda3 to my PATH environment variable

NOT recommended. This can lead to conflicts with other applications. Instead, use the Commmand Prompt and Powershell menus added to the Windows Start Menu.

☑ Register Miniconda3 as my default Python 3.11

Recommended. Allows other programs, such as VSCode, PyCharm, etc. to automatically detect Miniconda3 as the primary Python 3.11 on the system.

☑ Clear the package cache upon completion

Recommended. Recovers some disk space without harming functionality.

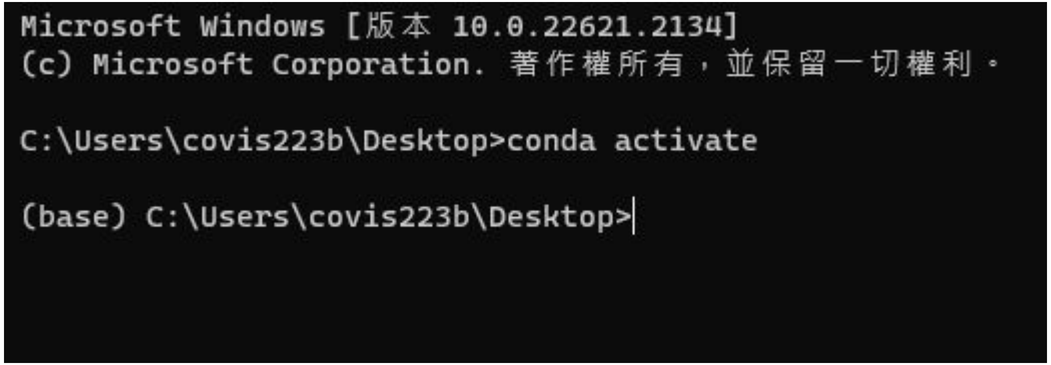
安裝conda

conda activate

-確認是否安裝成功

conda update conda

-更新conda



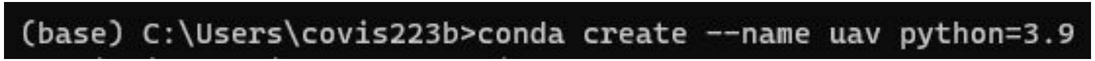
```
Microsoft Windows [版本 10.0.22621.2134]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\covis223b\Desktop>conda activate

(base) C:\Users\covis223b\Desktop>
```

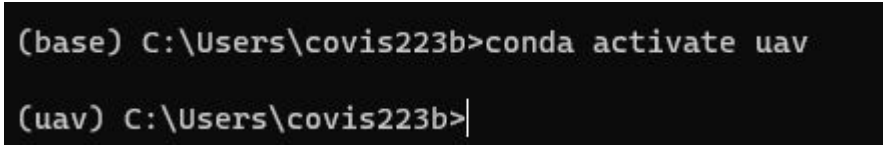
創建conda環境

conda create --name uav python=3.9



```
(base) C:\Users\covis223b>conda create --name uav python=3.9
```

conda activate uav



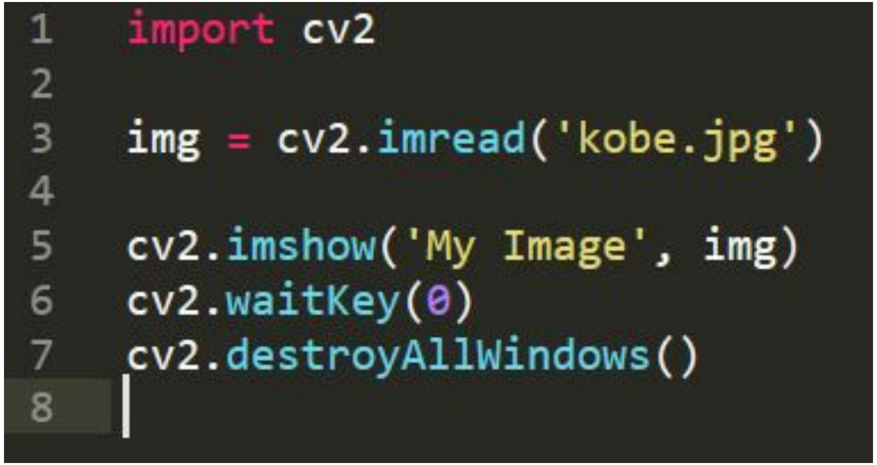
```
(base) C:\Users\covis223b>conda activate uav

(uav) C:\Users\covis223b>
```

！！！務必使用conda環境操作無人機！！！

3. 安裝opencv

- pip install opencv-python**==4.4.0.46**
- Test :



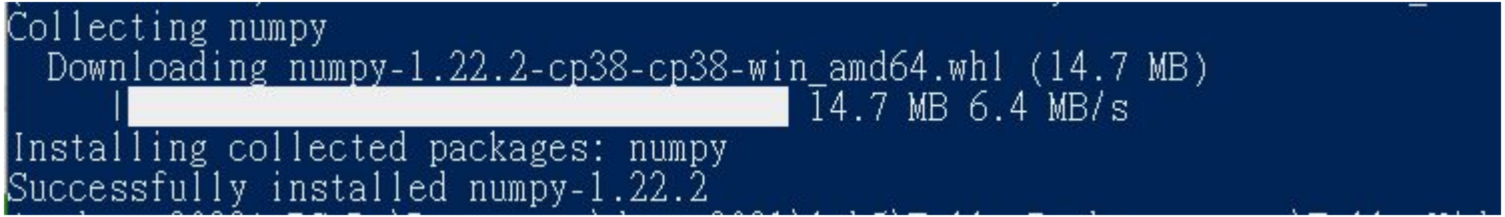
```python
import cv2

img = cv2.imread('kobe.jpg')

cv2.imshow('My Image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

4. 安裝numpy

- pip install numpy



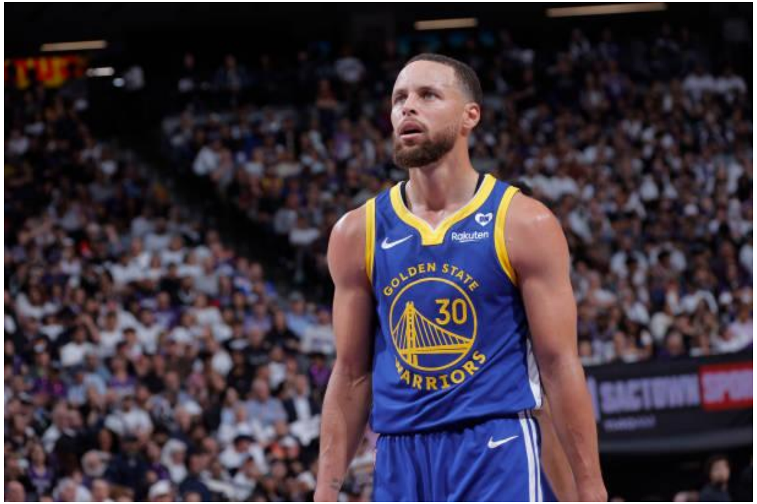
```
Collecting numpy
  Downloading numpy-1.22.2-cp38-cp38-win_amd64.whl (14.7 MB)
     |████████████████████████████████| 14.7 MB 6.4 MB/s
Installing collected packages: numpy
Successfully installed numpy-1.22.2
```

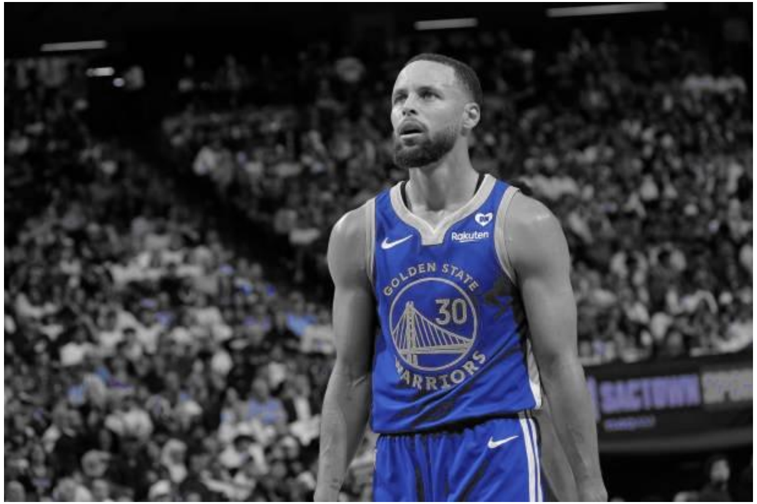
Lab 01

1. 圖片灰階與顏色濾鏡，對比與亮度
2. Bilinear Interpolation
3. 邊緣偵測(filtering & Sobel Operator)

1.1 灰階與顏色濾鏡(20%)

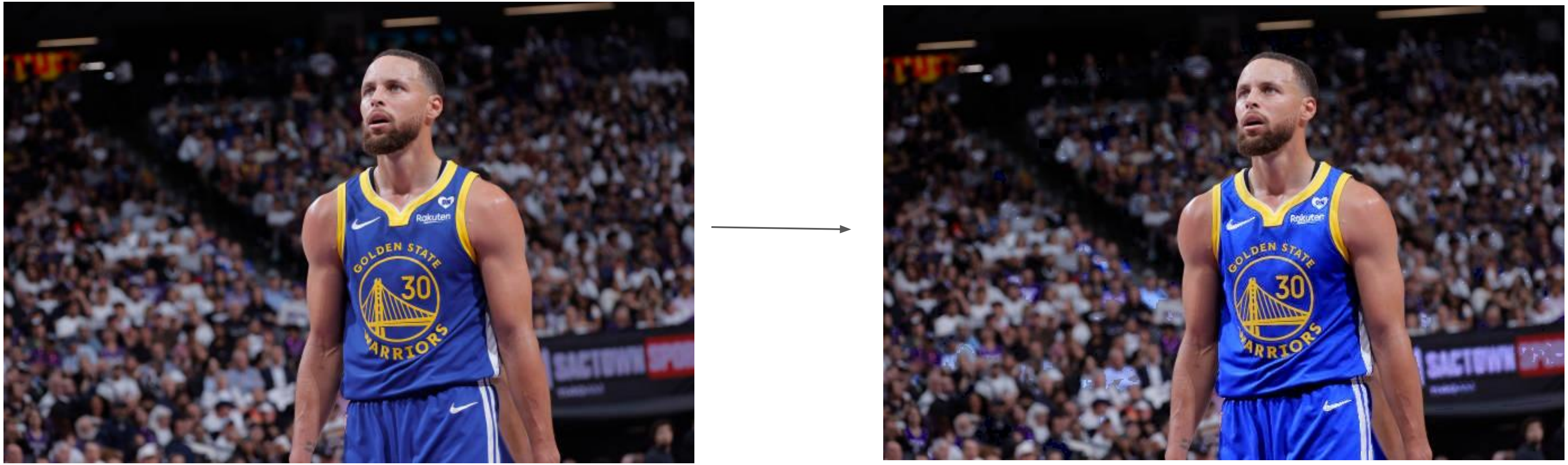
- 將原始圖片中的「藍點」予以保留，並把其餘的點改為灰階。
- Hint: B > 100 and B * 0.6 > G and B * 0.6 > R





1.2 對比與亮度(10%)

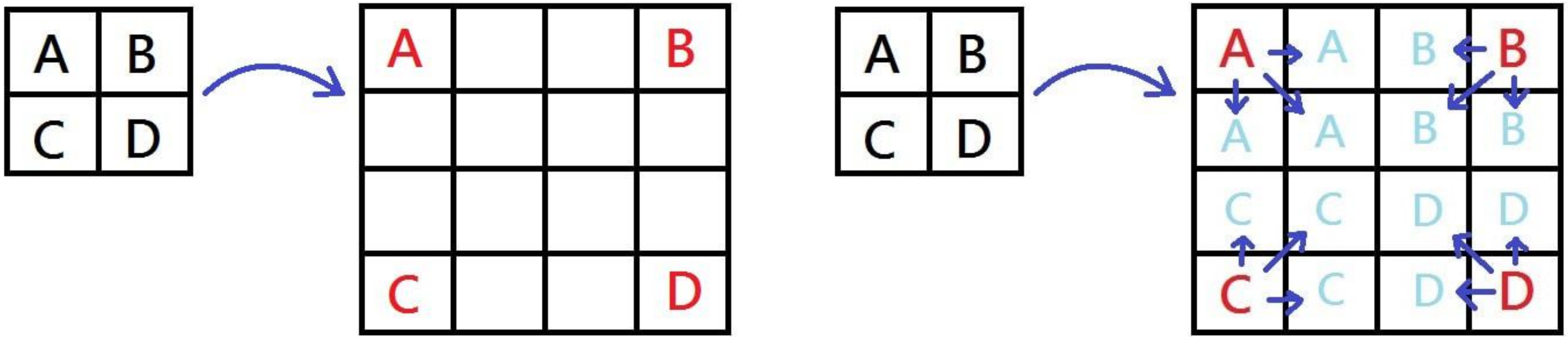
- 更改原始圖片中的「藍點與黃點」像素的對比與亮度，其餘像素保持原樣
- Hint: (B + G) * 0.3 > R
- new_image = (old_image - 127) × (contrast/127 + 1) + 127 + brightness
    - Hint: 記得注意overflow的問題
        - 可能會用到的函式 np.array(img, dtype=np.int32)、np.clip(img, 0, 255)、np.array(img, dtype=np.uint8)



EX:contrast=100, brightness=40

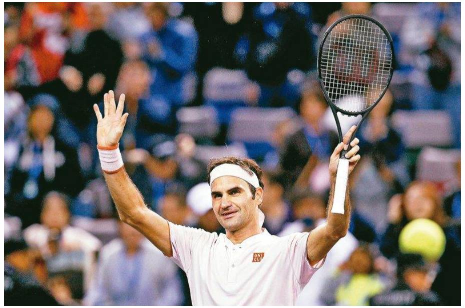
Interpolation - 最近相鄰內插法

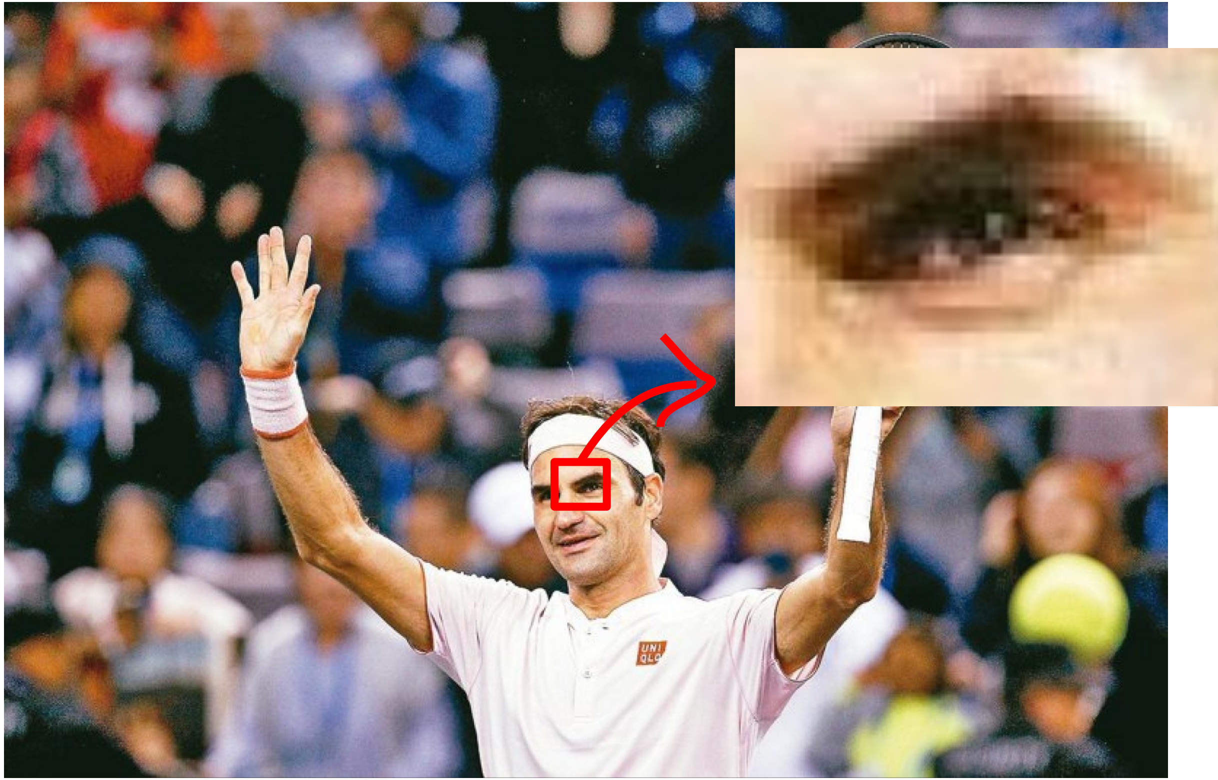
- 根據輸出影像的像素位置 ,找到輸入影像中最鄰近的點 ,即當作輸出影像的像素強度。
- 以下圖為例



Interpolation - 最近相鄰內插法

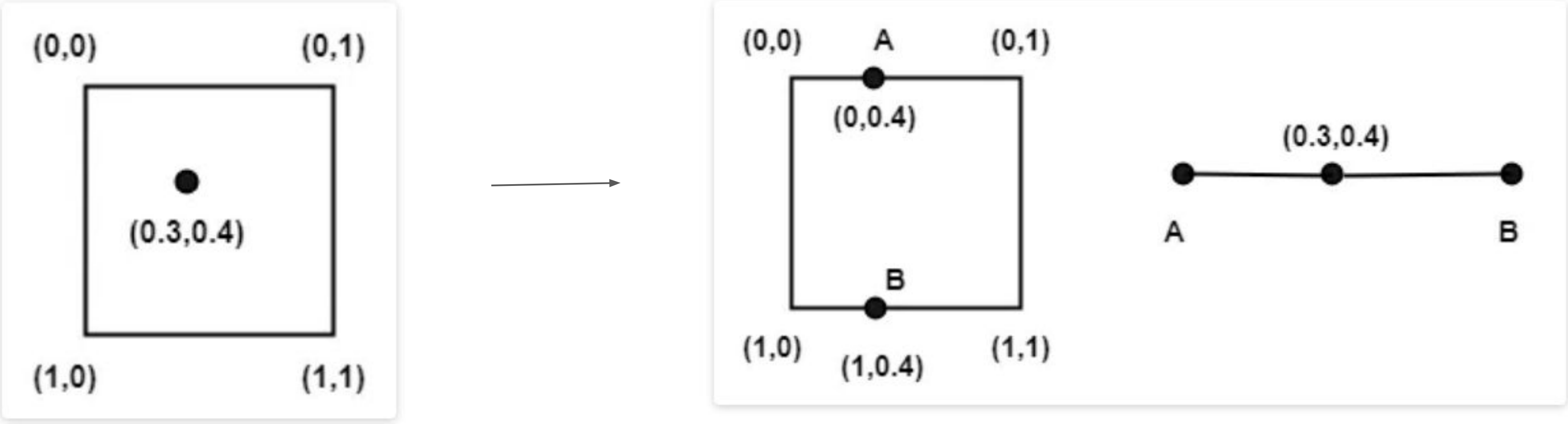
- 將照片放大3倍





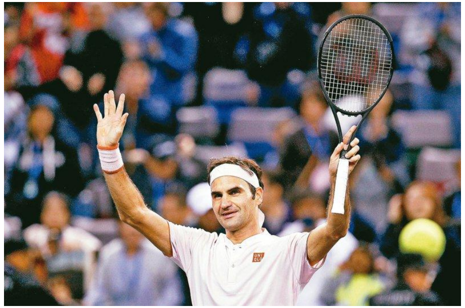
2. Interpolation - 雙線性內插法 (40%)

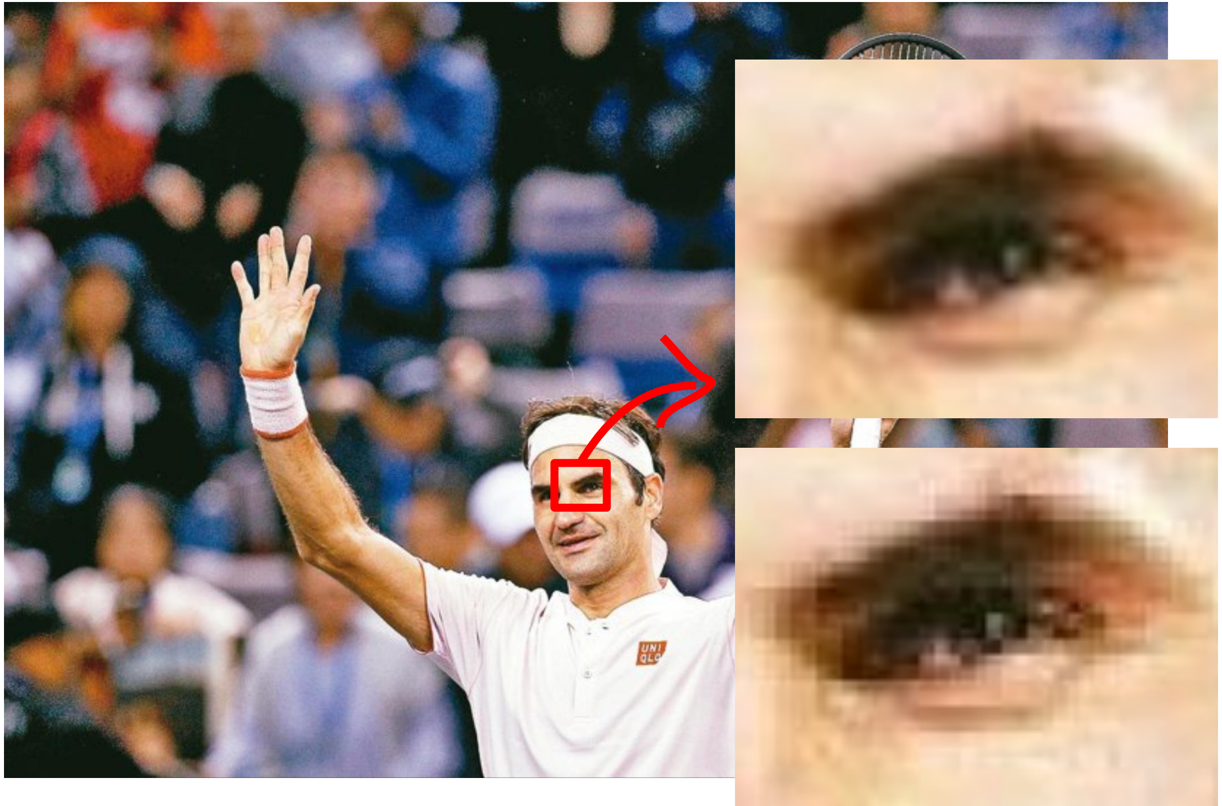
- 根據輸出影像的像素位置, 找到輸入影像中最鄰近的四個點 ,再利用雙線性 內插法求出輸出影像的像素強度。



2. Interpolation - 雙線性內插法 (40%)

- 以參數方式輸入影像以及倍率
- 學會使用 OpenCV API (10%)
  自行實作雙線性內插法 (40%)
- 下圖為輸入影像
  右圖為
  倍率=3之結果





3.邊緣偵測(filtering & Sobel Operator) (30%)

img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) 將圖片轉為灰階

img = cv2.GaussianBlur(img, (5, 5), 0) 對灰階圖做高斯模糊(去雜訊)

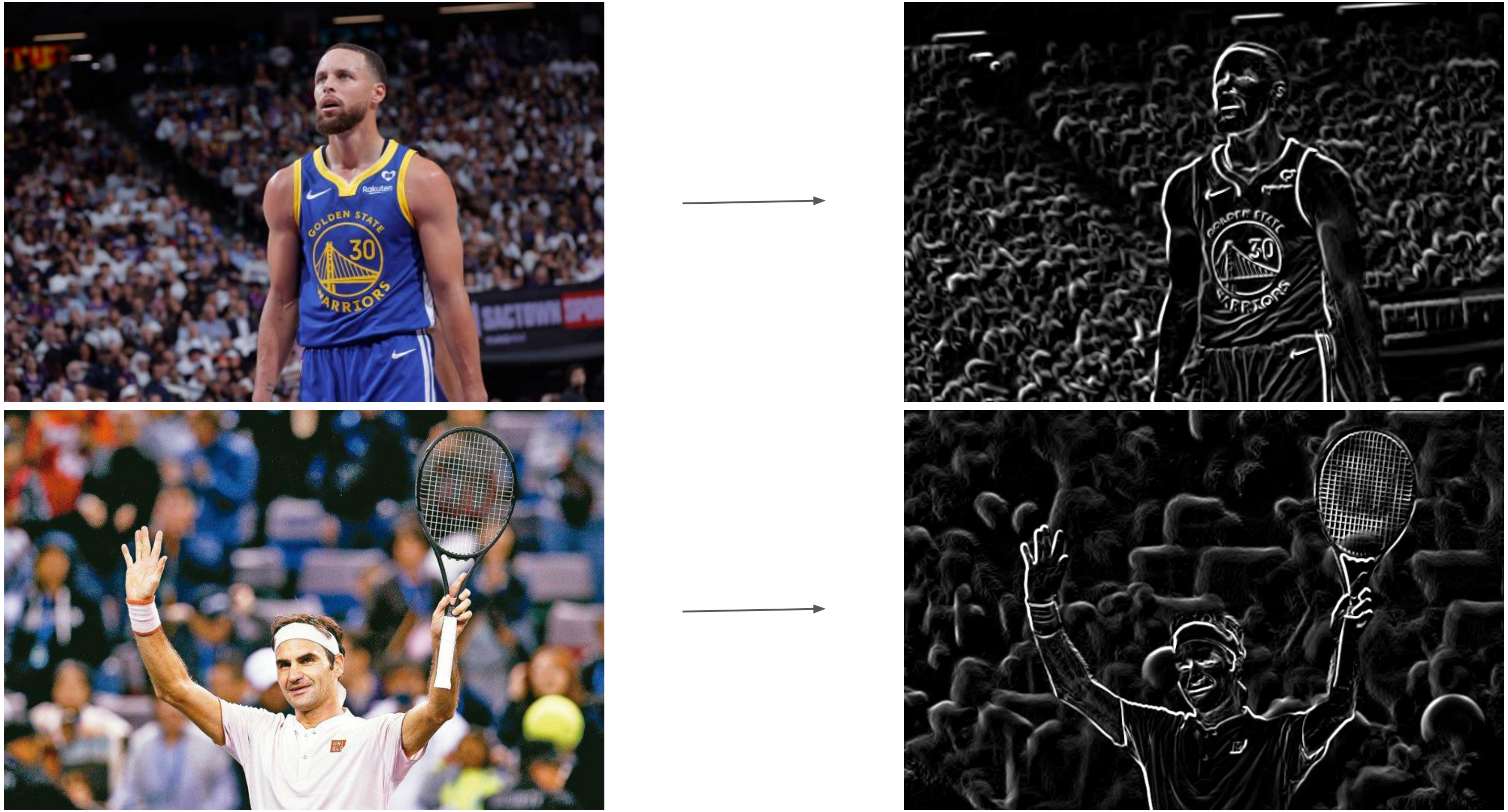
$$\mathbf{G_x} = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G_y} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

cv2.filter2D(img, -1, kernel)

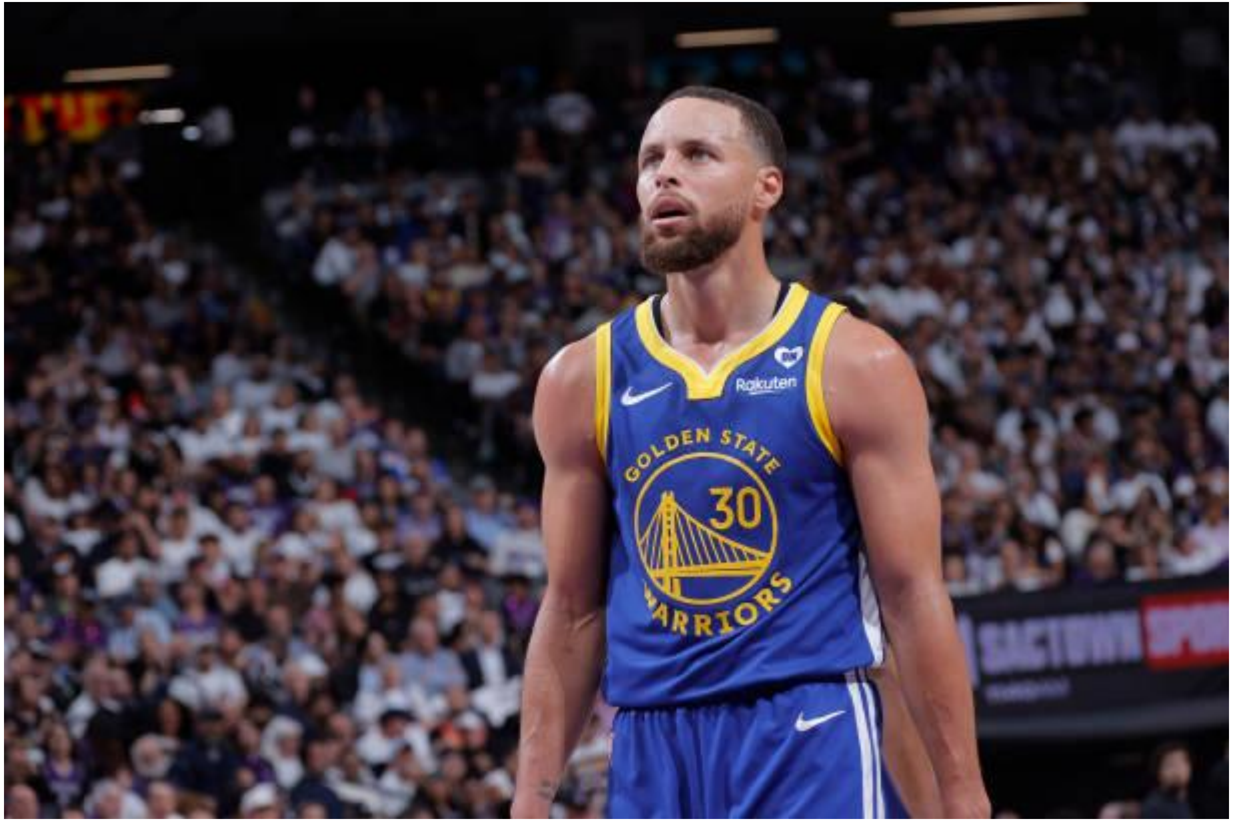
Do NOT use cv2.Sobel() directly

$$G = \sqrt{G_x^2 + G_y^2}$$

3.邊緣偵測(filtering & Sobel Operator) (30%)



Demo image for Q1 (test.jpg)



Demo image for Q2 and Q3 (ive.jpg)