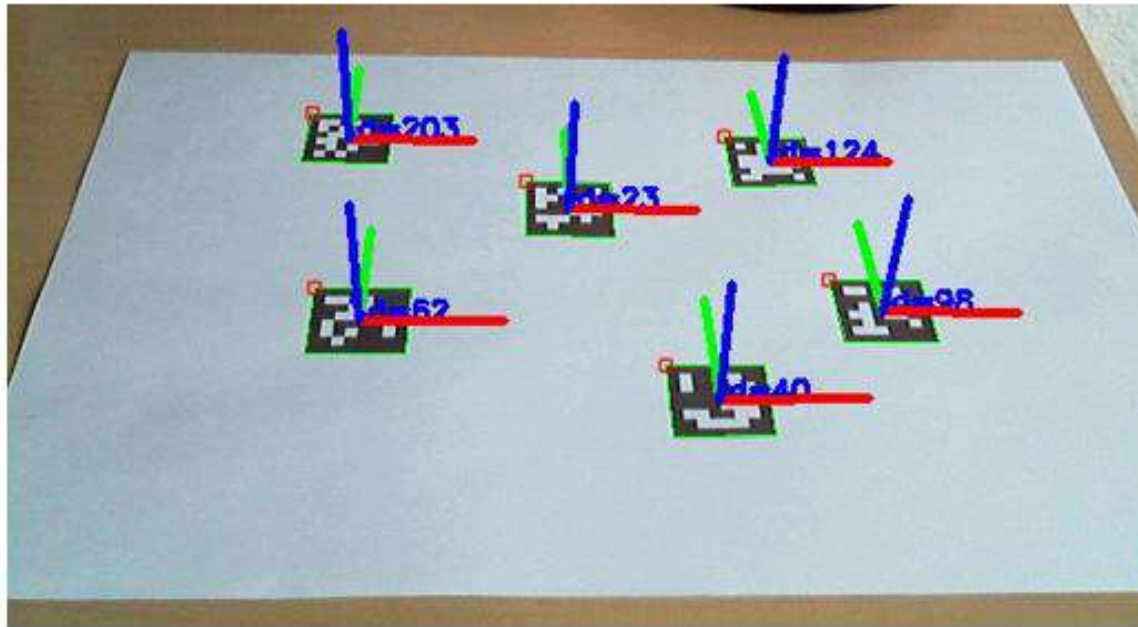
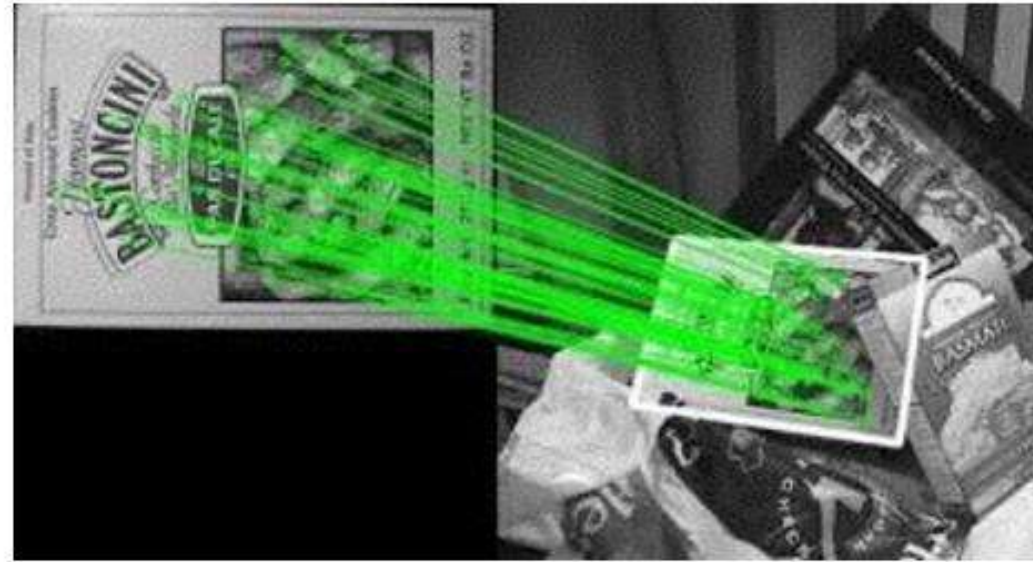


# OPENCV BASICS



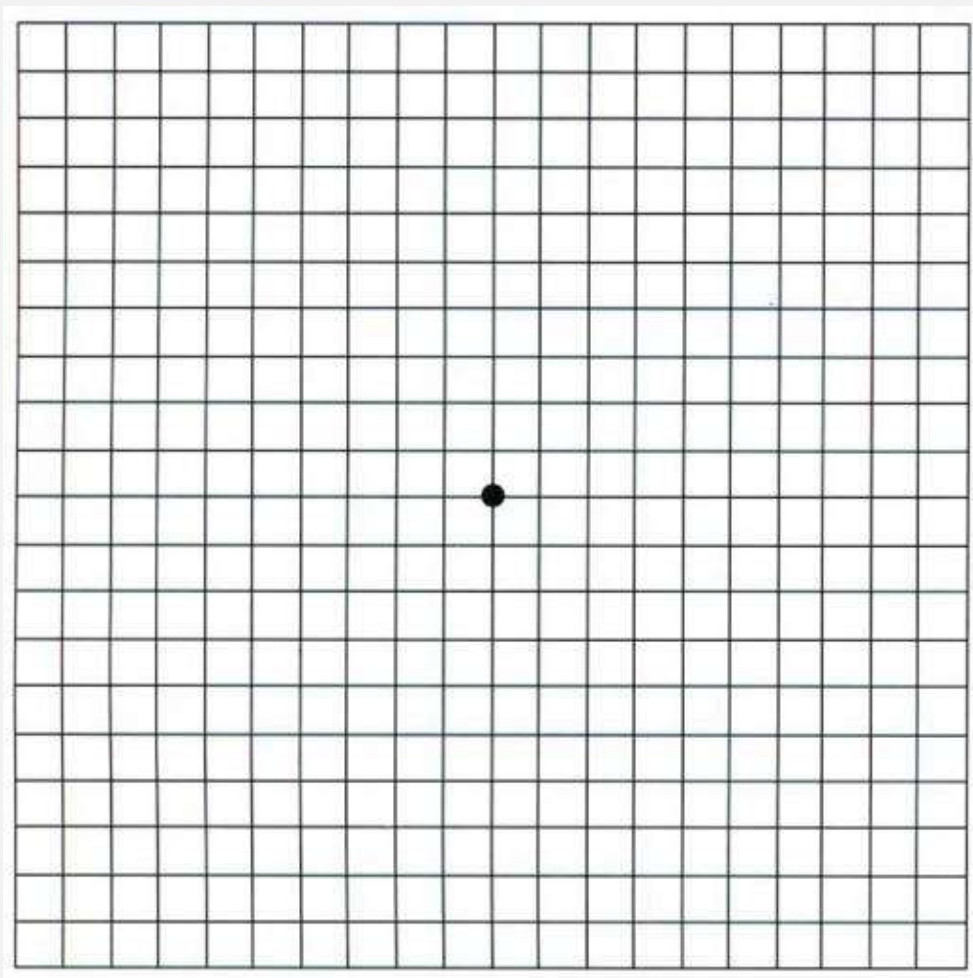
- core. The Core Functionality
- imgproc. Image Processing
- imgcodecs. Image file reading and writing
- videoio. Media I/O
- highgui. High-level GUI and Media I/O
- video. Video Analysis
- calib3d. Camera Calibration and 3D Reconstruction
- features2d. 2D Features Framework
- objdetect. Object Detection
- 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 derived 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:

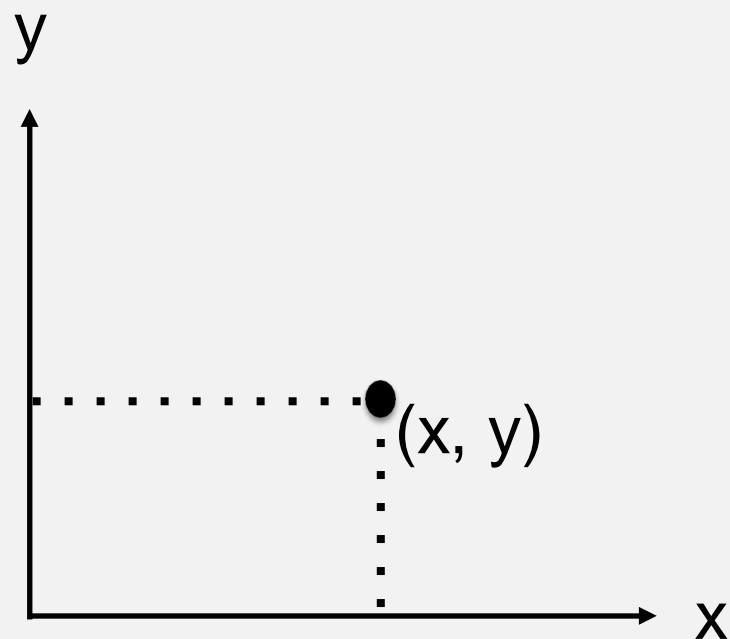
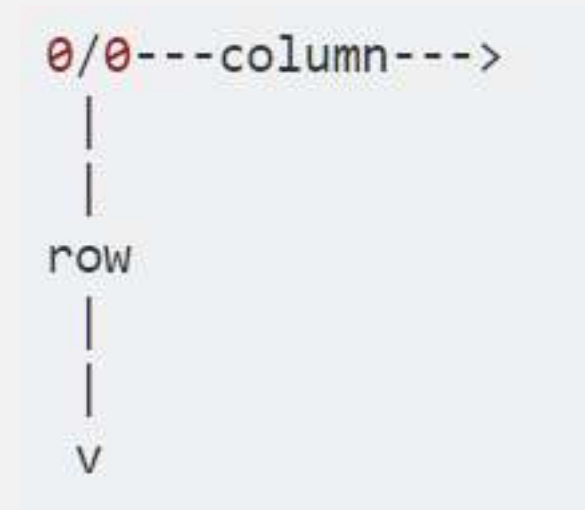
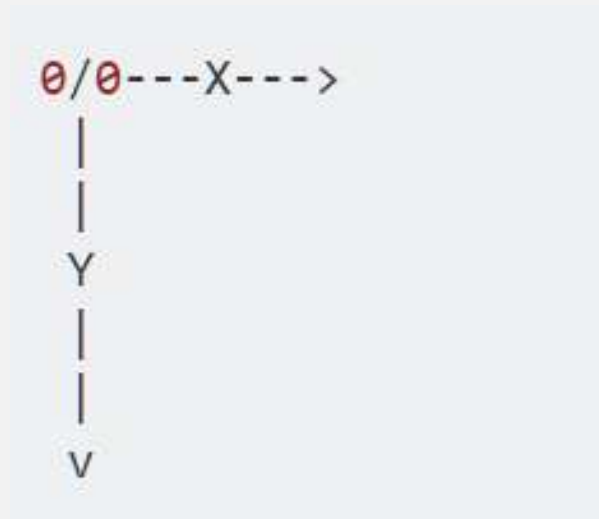


image:



# 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



```
1  import numpy as np
2  import cv2
3
4  #read
5  image = cv2.imread("image.jpg")
6  #show
7  cv2.imshow("My Image", image)
8
9  #按下按鍵關閉顯示視窗
10 cv2.waitKey(0)
11 cv2.destroyAllWindows()
12
13 #save
14 cv2.imwrite("output.jpg", image)
```

# 標頭引入

```
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

顯示影像:

```
# 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)
```

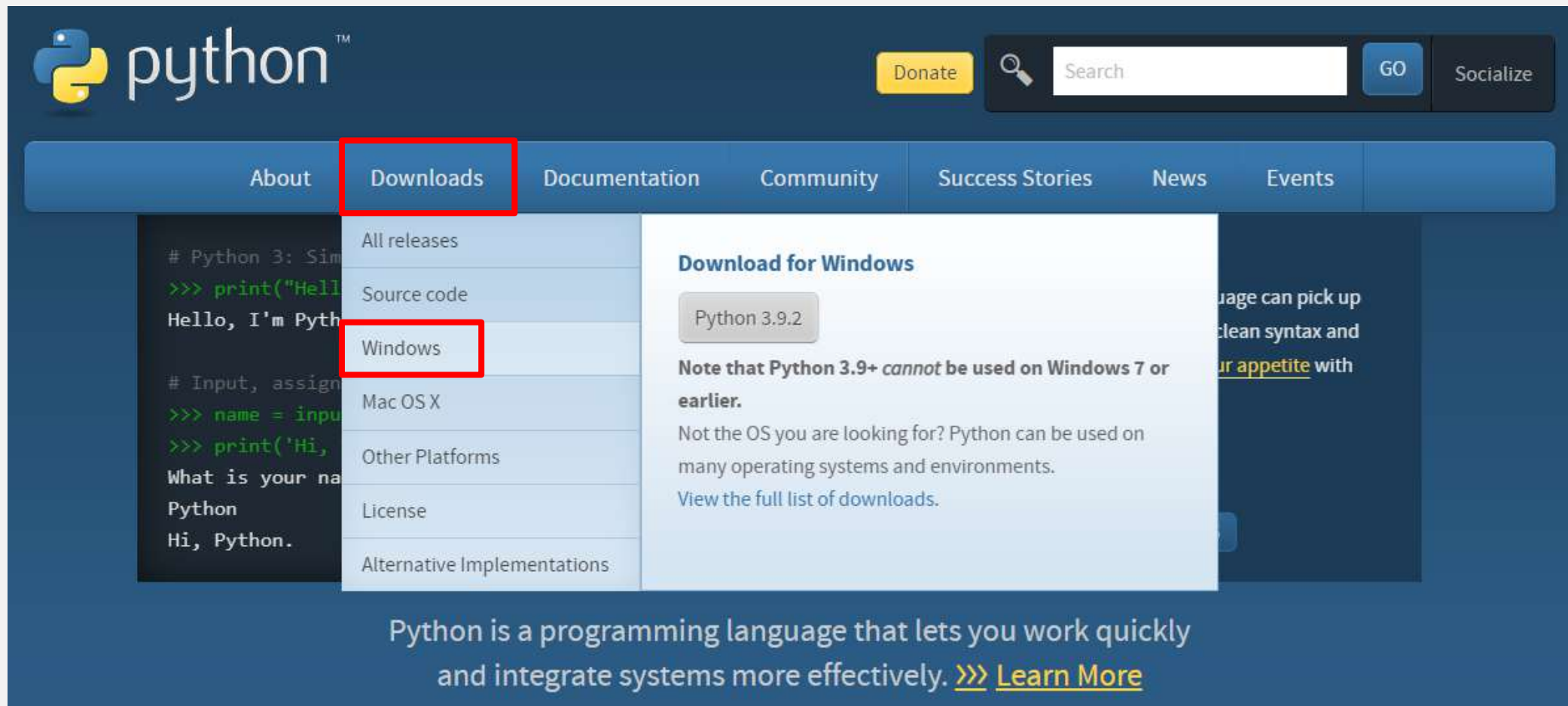
# 操作像素

`image[row, col, channel]`

WINDOWS10  
PYTHON 3 & OPENCV

# I. 下載PYTHON


到官網點選Downloads



# I. 下載PYTHON

選擇python 3

## Python Releases for Windows

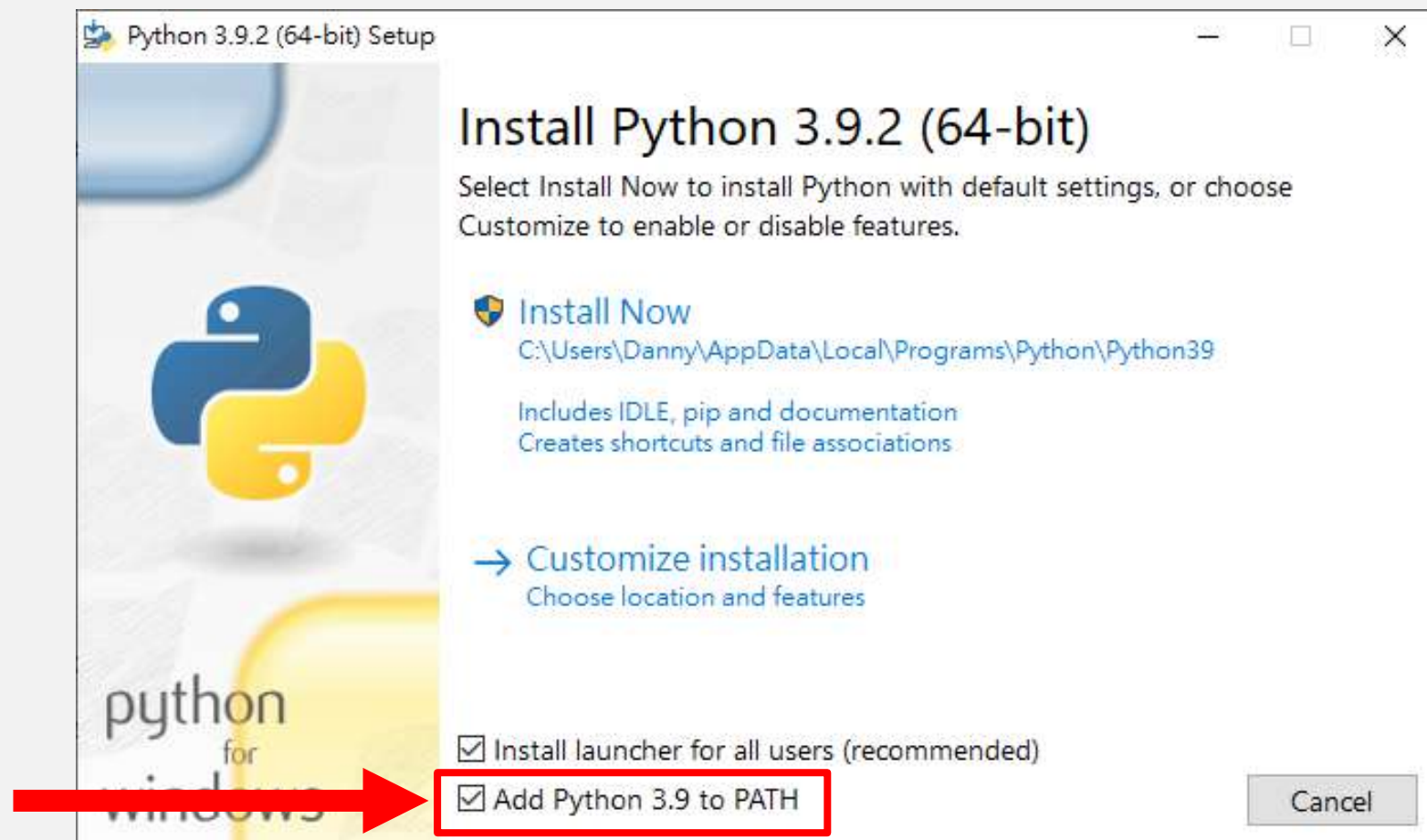
- 
- [Latest Python 3 Release - Python 3.9.2](#)
  - [Latest Python 2 Release - Python 2.7.18](#)



## 2. 安裝軟體

- 點選並安裝
- 加入環境變數

 python-3.9.2-amd64.exe



### 3. 安裝OPENCV

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

```
1  import cv2
2
3  img = cv2.imread('kobe.jpg')
4
5  cv2.imshow('My Image', img)
6  cv2.waitKey(0)
7  cv2.destroyAllWindows()
8  |
```

## 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
```

# HOMEWORK I

# CONTENT

- 使用不同的插值方法對影像中心順時鐘旋轉30度
- 使用不同的插值方法對影像放大2倍
  - Nearest Neighbor Interpolation
  - Bilinear Interpolation
  - Bicubic Interpolation

原圖

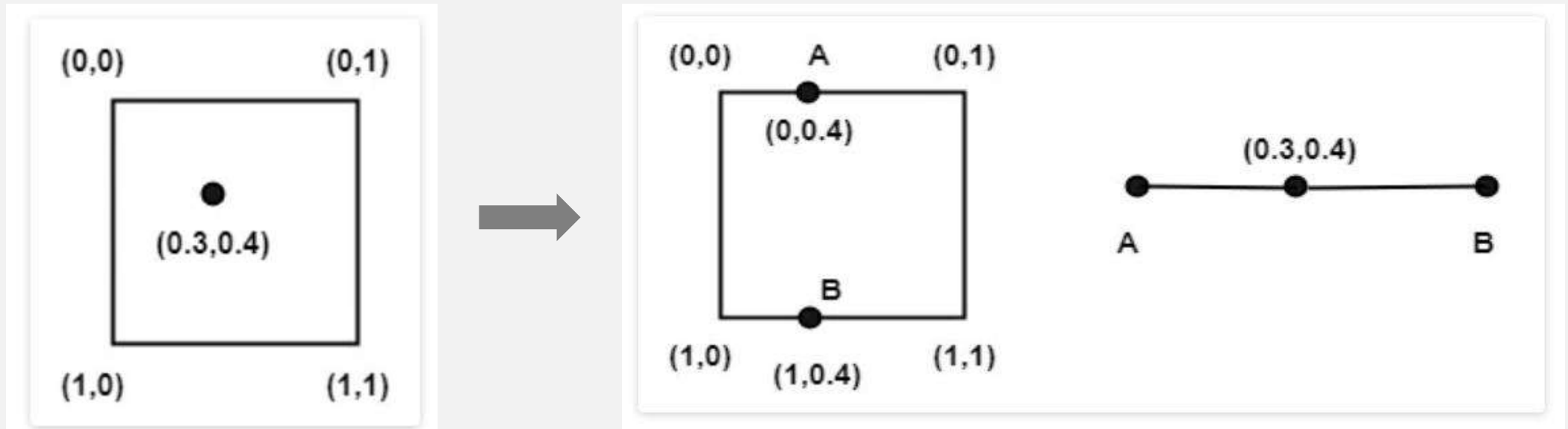


結果示意圖



# BILINEAR INTERPOLATION

- 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。





# BICUBIC INTERPOLATION

- 三次樣條插值 (Cubic spline interpolation)
- 如果一個函數  $f(x)$  在  $x=0$  和  $x=1$  的位置上的函數值已知，利用一個三次方的多項式去計算這條曲線在  $[0,1]$  上的值

$$f(x) = ax^3 + bx^2 + cx + d$$

$$f'(x) = 3ax^2 + 2bx + c$$

$$f(0) = d$$

$$f(1) = a + b + c + d$$

$$f'(0) = c$$

$$f'(1) = 3a + 2b + c$$



$$a = 2f(0) - 2f(1) + f'(0) + f'(1)$$

$$b = -3f(0) + 3f(1) - 2f'(0) - f'(1)$$

$$c = f'(0)$$

$$d = f(0)$$

# BICUBIC INTERPOLATION

- 假設我們有四個值，分別  $p_0, p_1, p_2, p_3$  分別代表  $x=-1, x=0, x=1$  和  $x=2$  位置的函數值, 此時並不知道他的導數，因此用這條直線的斜率來近似替代這個位置導數

$$\begin{aligned}f(0) &= p_1 \\f(1) &= p_2 \\f'(0) &= \frac{p_2 - p_0}{2} \\f'(1) &= \frac{p_3 - p_1}{2}\end{aligned}$$

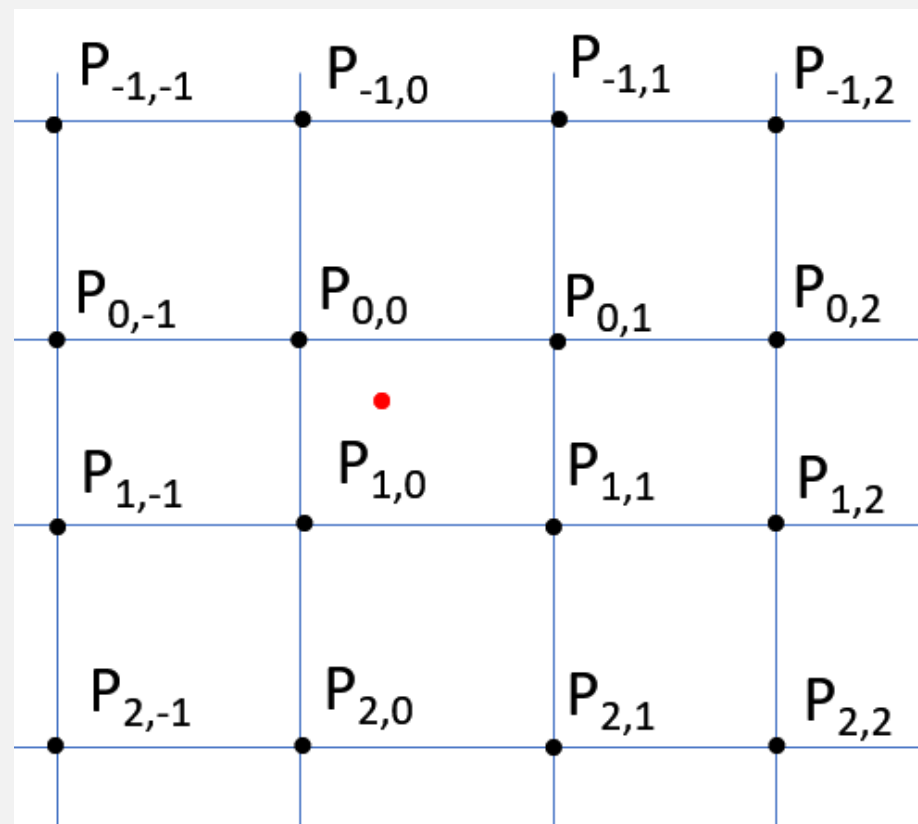


$$\begin{aligned}a &= -\frac{1}{2}p_0 + \frac{3}{2}p_1 - \frac{3}{2}p_2 + \frac{1}{2}p_3 \\b &= p_0 - \frac{5}{2}p_1 + 2p_2 - \frac{1}{2}p_3 \\c &= -\frac{1}{2}p_0 + \frac{1}{2}p_2 \\d &= p_1\end{aligned}$$

$$f(p_0, p_1, p_2, p_3, x) = \left(-\frac{1}{2}p_0 + \frac{3}{2}p_1 - \frac{3}{2}p_2 + \frac{1}{2}p_3\right)x^3 + \left(p_0 - \frac{5}{2}p_1 + 2p_2 - \frac{1}{2}p_3\right)x^2 + \left(-\frac{1}{2}p_0 + \frac{1}{2}p_2\right)x + p_1$$

# BICUBIC INTERPOLATION

- 雙三次插值法就是二維的三次樣條插值
- **Hint:** 記得注意**overflow**的問題
  - `np.clip(img, 0, 255)`
  - `np.array(img, dtype=np.uint8)`



# GRADING POLICY

- Each method – 15%
  - 禁止使用 OpenCV (讀寫照片以外)的函式
- Report – 10%

# SUBMISSION

- Report
  - 包含 Method 、 Result 、 Feedback 三部分
  - 至多3頁
  - Filename: **STUDENT\_ID.pdf**
- Code
  - Filename: **STUDENT\_ID.zip**
- Deadline: **3/22 Fri. 10:10 a.m.**