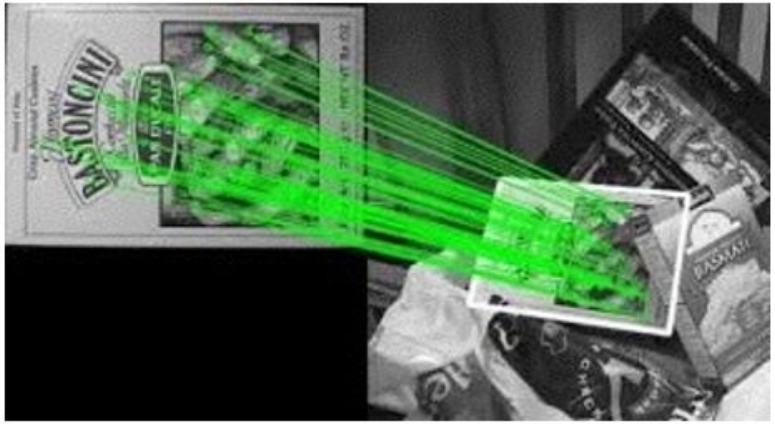
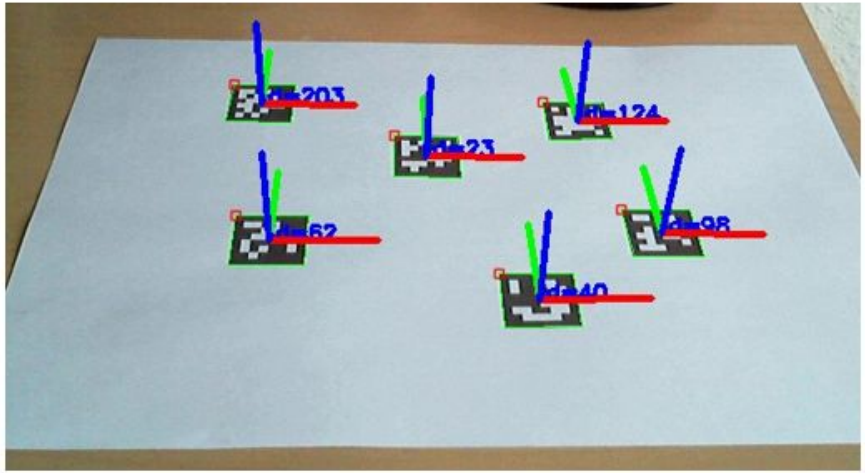
9/19 Lab 01

1. OpenCV introduction
2. Python 3 & opencv installation
3. Lab01

- 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

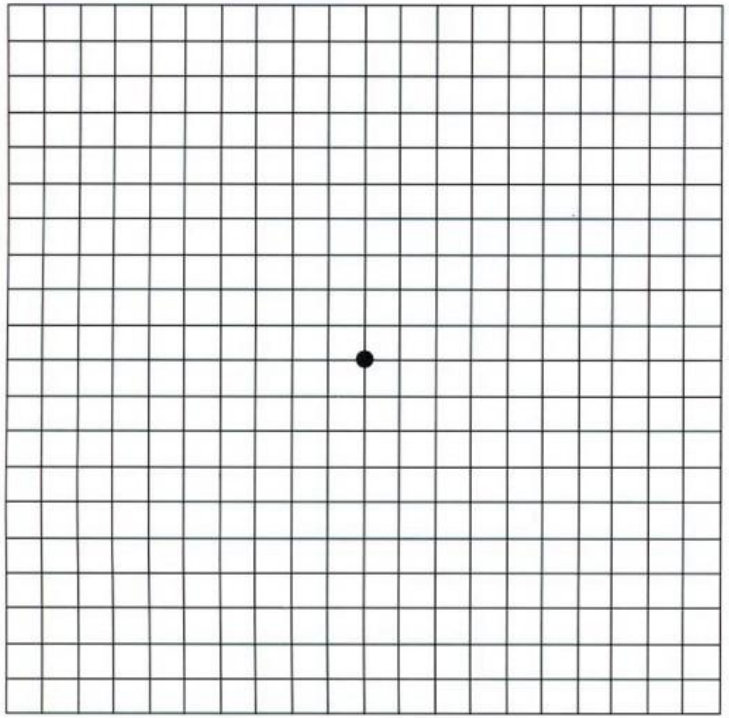
feature detection





pattern recognition

Mat



rows: 長
cols: 寬
type: 像素型態
channels: 通道數

normal:

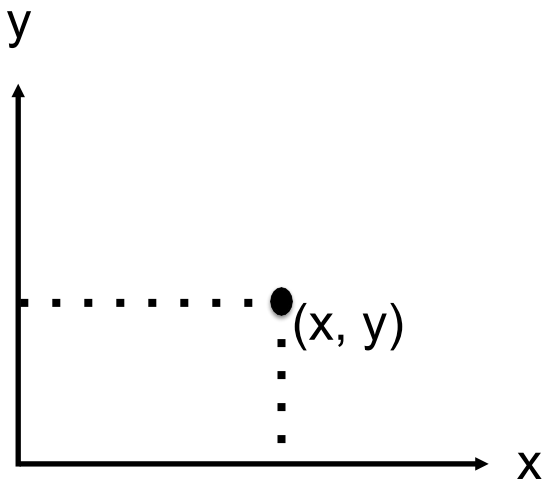


image:

```
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

```
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:

等待按鍵輸入:

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

# 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
複製圖片

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

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

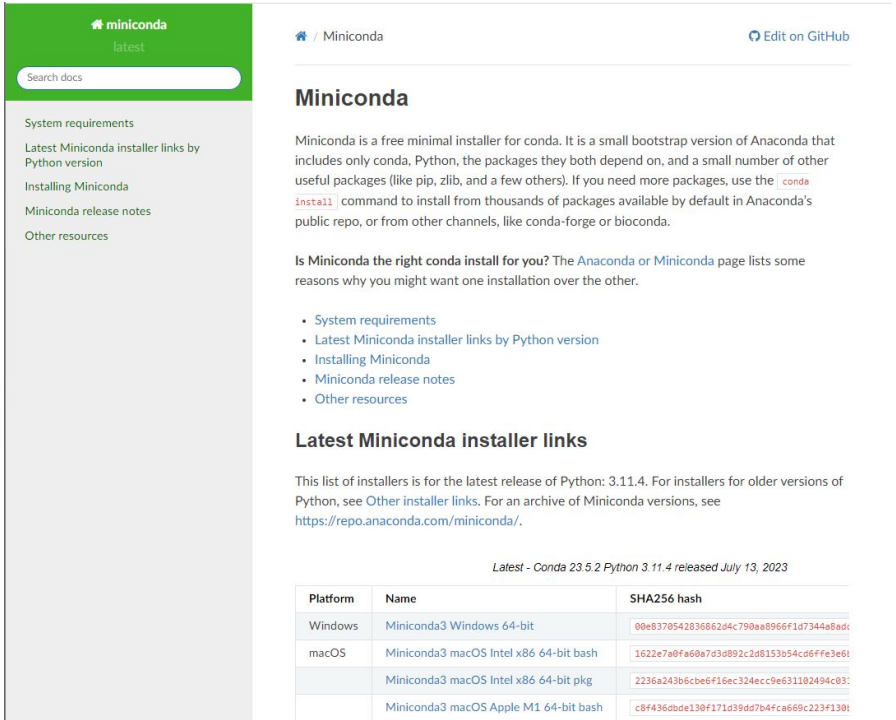
操作像素

image[row, col, channel]

Python 3 & OpenCV Installation

安装conda

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

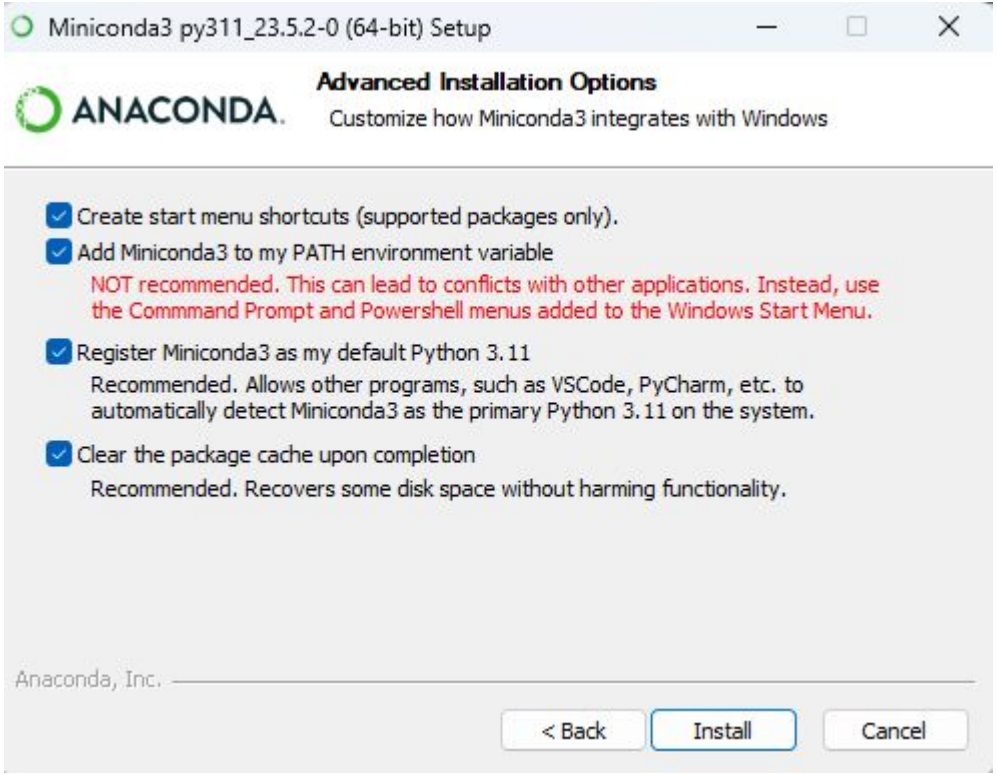


安裝conda

一直按next直到下一頁的畫面

安裝conda

這邊建議4個都打勾



安裝conda

conda activate

-確認是否安裝成功

conda update conda

-更新conda

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>
```

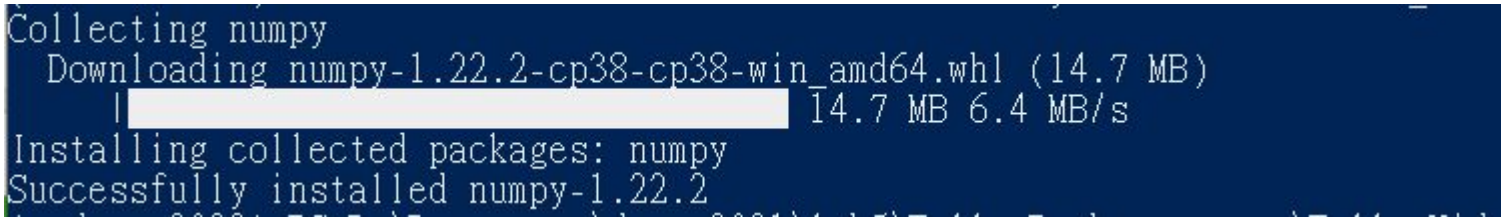
3. 安装opencv

- 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. 安装numpy

- pip install numpy



Lab 01

1. 圖片灰階與顏色濾鏡，對比與亮度
2. Nearest Neighbor Interpolation
3. Bilinear Interpolation

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

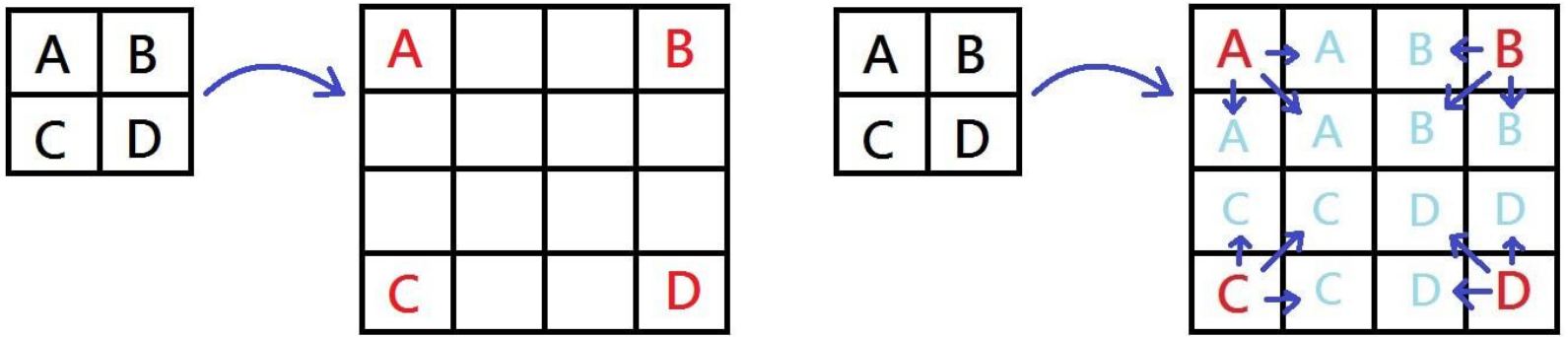
1.2 對比與亮度(10%)

- 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
    - 你可能會用到的函式：np.array(img, dtype=np.int32)、np.clip(img, 0, 255)、np.array(img, dtype=np.uint8)

EX:contrast=100, brightness=40

2. Interpolation - 最近相鄰內插法 (30%)

- 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

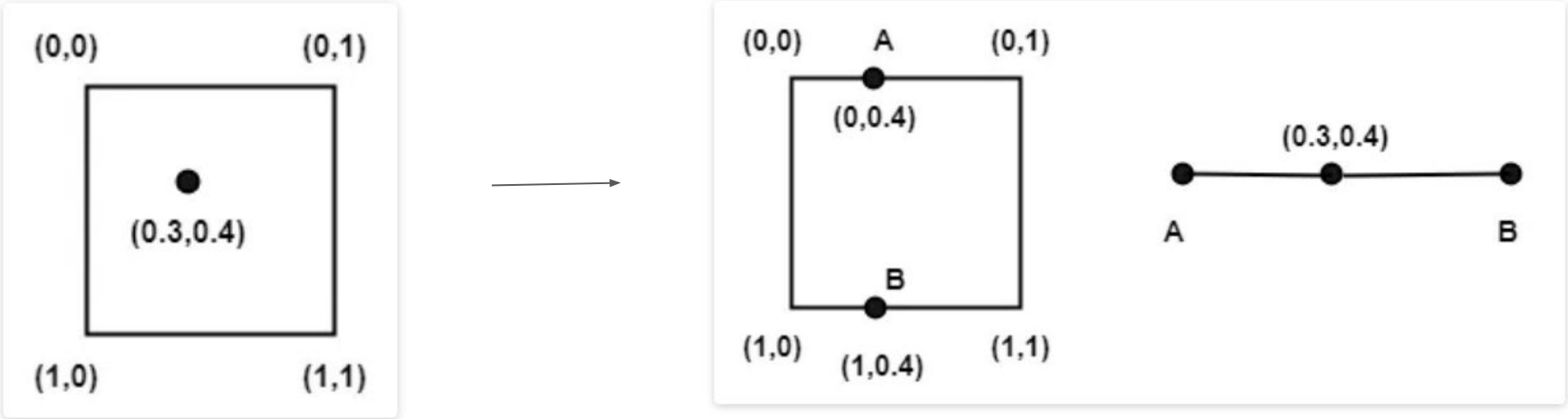


2. Interpolation - 最近相鄰內插法 (30%)

- 將照片放大3倍

3. Interpolation - 雙線性內插法 (40%)

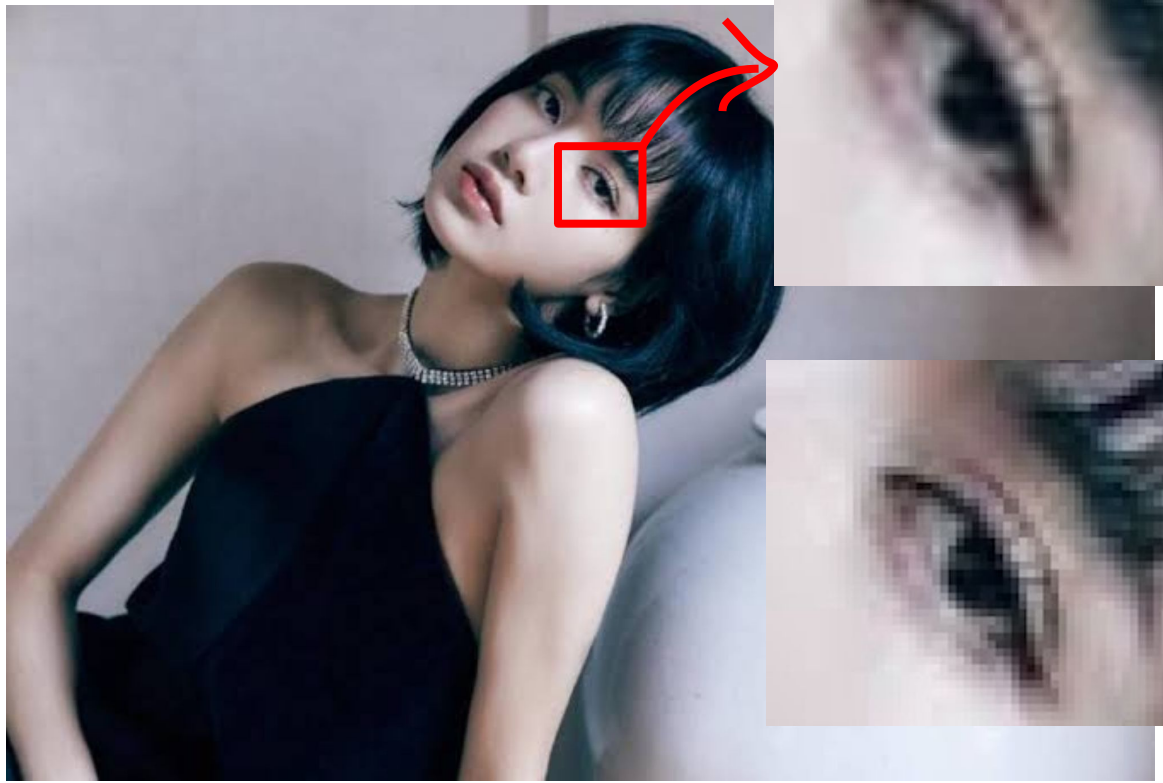
- 根據輸出影像的像素位置，找到輸入影像中最鄰近的四個點 ,再利用雙線性內插法求出輸出影像的像素強度。



3. 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



2, 3. Demo image (test.jpg)