

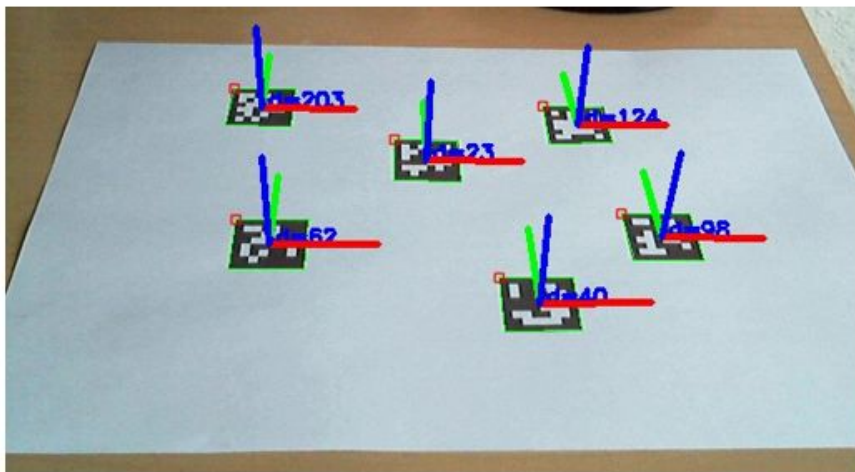
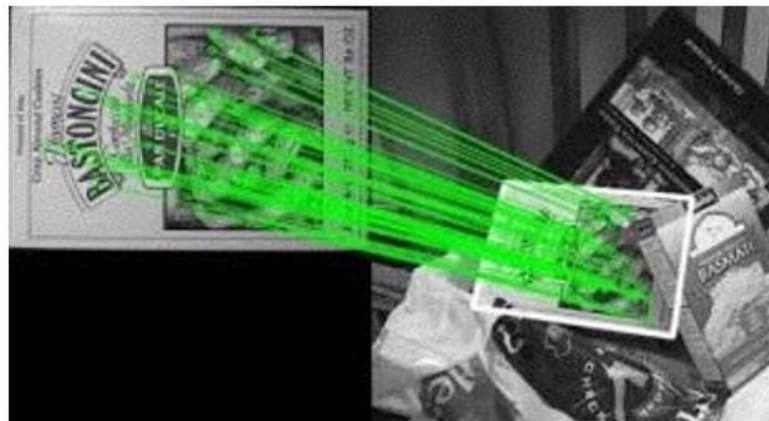
9/19 Lab 01

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



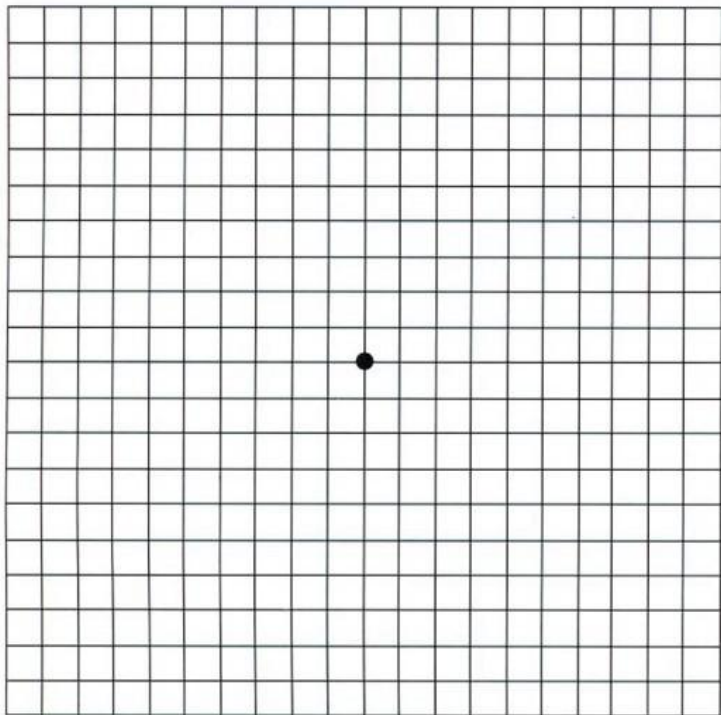
- core. The Core Functionality
- imgproc. Image Processing
- imgcodecs. Image file reading and writing
- videoio. Media I/O
- highgui. High-level GUI and Media I/O
- video. Video Analysis
- calib3d. Camera Calibration and 3D Reconstruction
- features2d. 2D Features Framework
- objdetect. Object Detection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

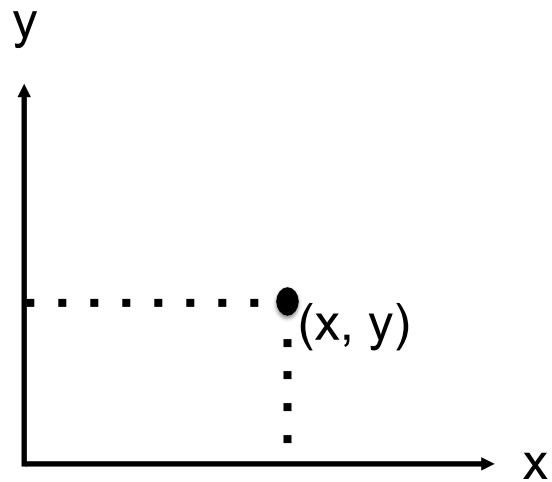
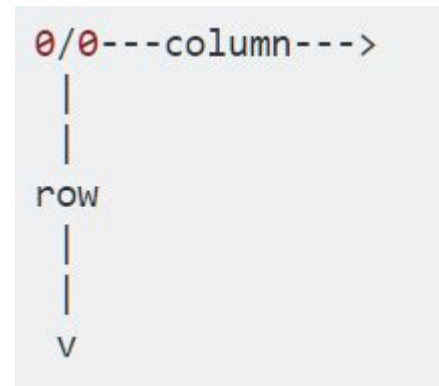
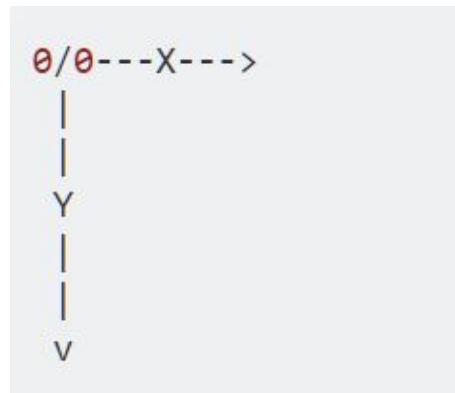


image:



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

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 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

安裝conda

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

miniconda
latest

Search docs

System requirements

Latest Miniconda installer links by Python version

Installing Miniconda

Miniconda release notes

Other resources

/ Miniconda

Edit on GitHub

Miniconda

Miniconda is a free minimal installer for conda. It is a small bootstrap version of Anaconda that includes only conda, Python, the packages they both depend on, and a small number of other useful packages (like pip, zlib, and a few others). If you need more packages, use the `conda install` command to install from thousands of packages available by default in Anaconda's public repo, or from other channels, like conda-forge or bioconda.

Is Miniconda the right conda install for you? The [Anaconda](#) or [Miniconda](#) page lists some reasons why you might want one installation over the other.

- System requirements
- Latest Miniconda installer links by Python version
- Installing Miniconda
- Miniconda release notes
- Other resources

Latest Miniconda installer links

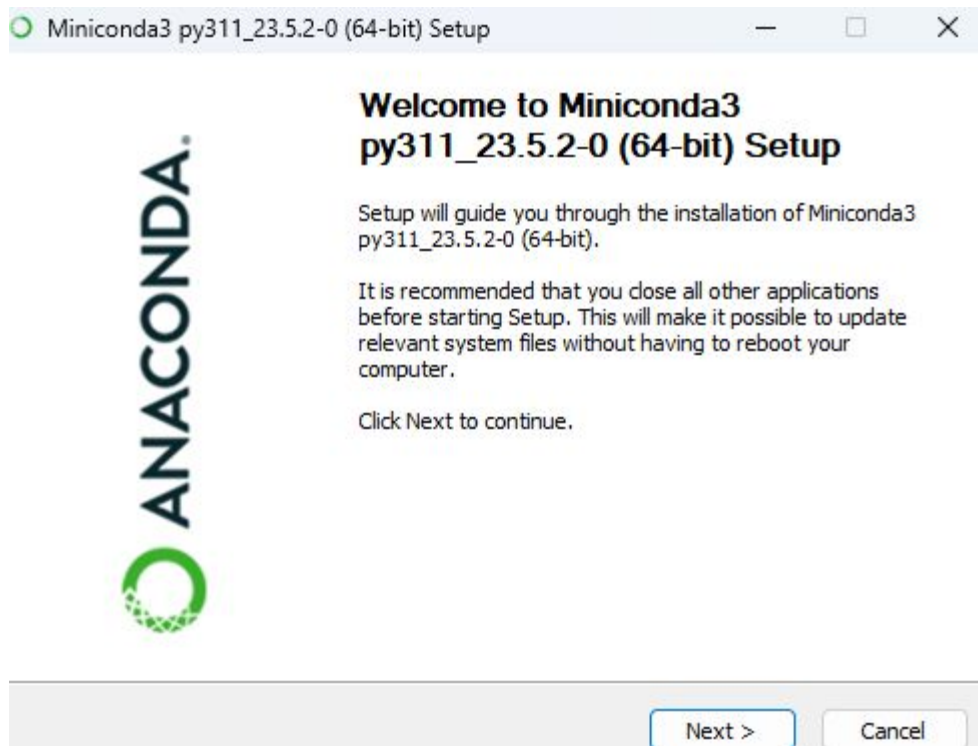
This list of installers is for the latest release of Python: 3.11.4. For installers for older versions of Python, see [Other installer links](#). For an archive of Miniconda versions, see <https://repo.anaconda.com/miniconda/>.

Latest - Conda 23.5.2 Python 3.11.4 released July 13, 2023

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	00e837854283686264c790aa8966f1d7344a8ad:
macOS	Miniconda3 macOS Intel x86 64-bit bash	1622e7a0fa60a7d3d892c2d8153b54c06ffe3e6l
	Miniconda3 macOS Intel x86 64-bit pkg	2236a243b6cbe6f16ec324ecc9e631102494c03:
	Miniconda3 macOS Apple M1 64-bit bash	c8f436dbde136f171d39dd7b4fca669c223f130l

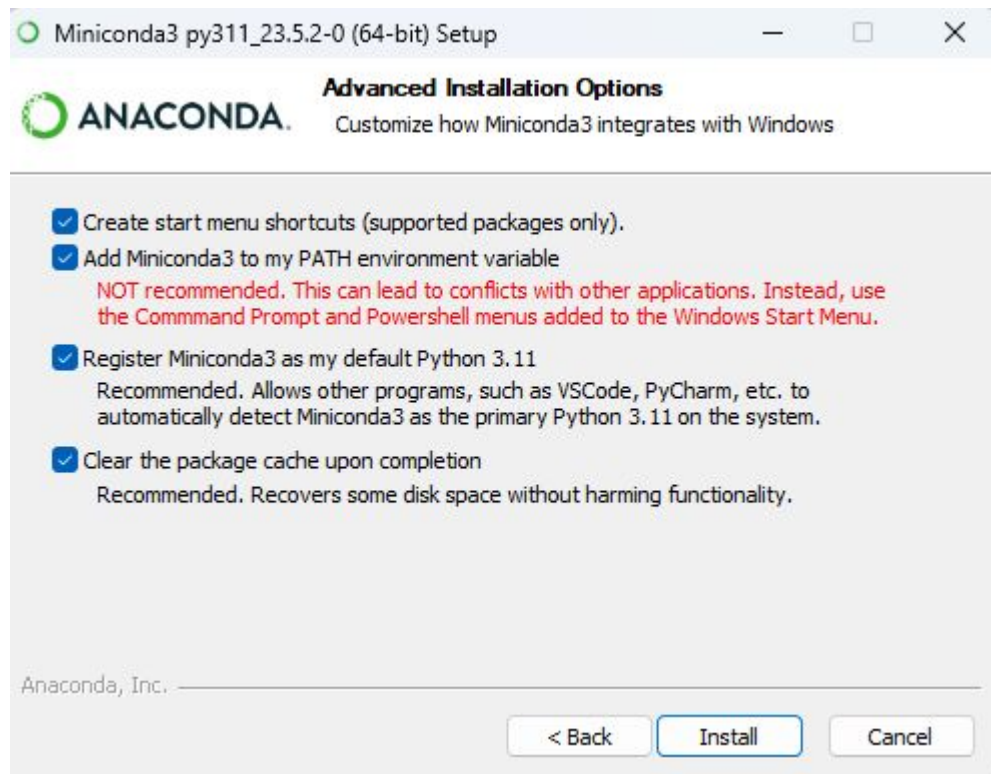
安裝conda

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

```
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>|
```

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
```

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%)

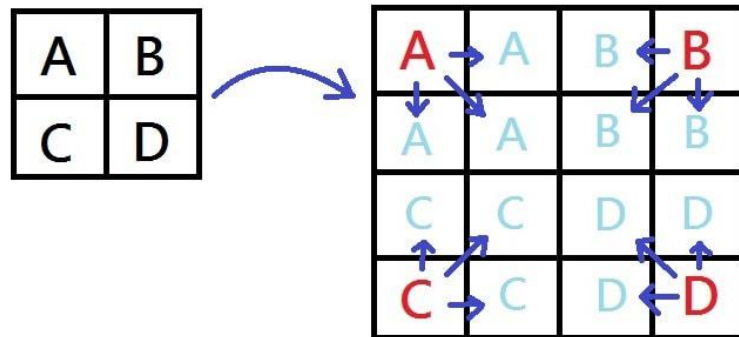
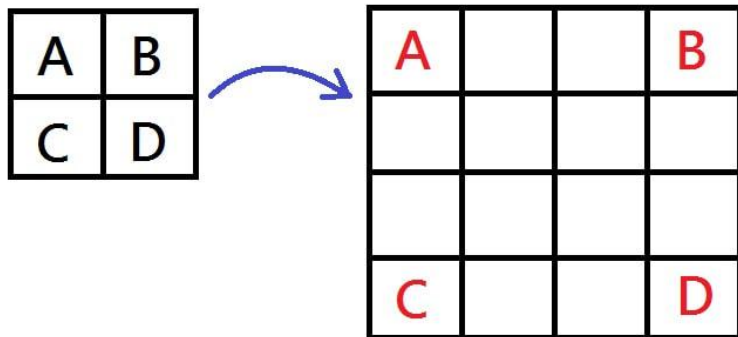
- 更改原始圖片中的「藍點與黃點」像素的對比與亮度，其餘像素保持原樣
- Hint: $(B + G) * 0.3 > R$
- $new_image = (old_image - 127) \times (contrast/127 + 1) + 127 + brightness$
 - Hint: 記得注意overflow的問題
 - 你可能會用到的函式：np.array(img, dtype=np.int32)、np.clip(img, 0, 255)、np.array(img, dtype=np.uint8)



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



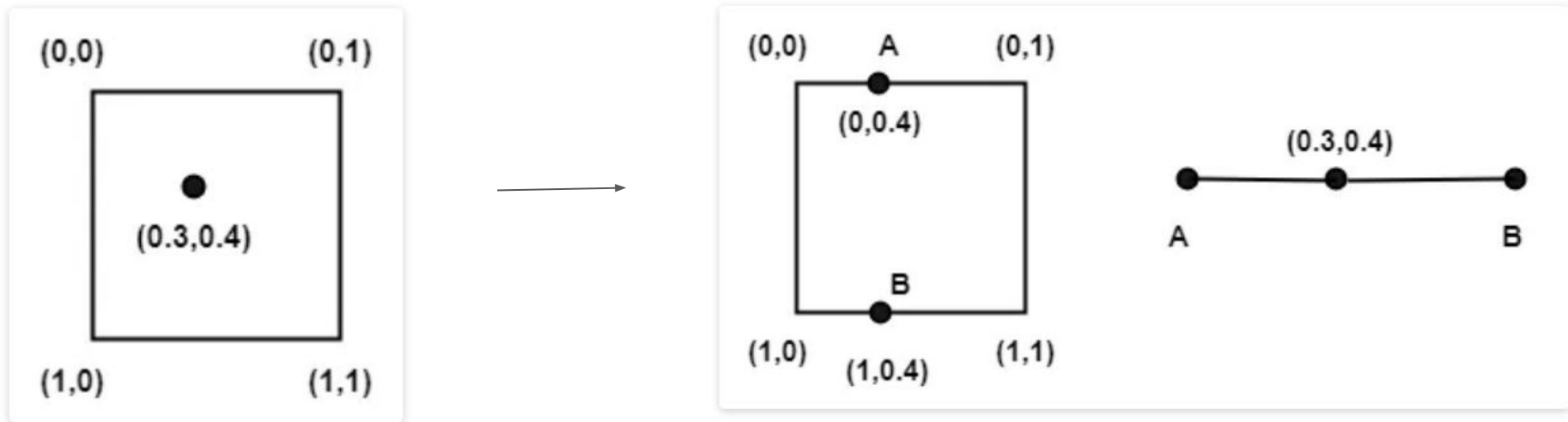
2. Interpolation - 最近相鄰內插法 (30%)

- 將照片放大3倍



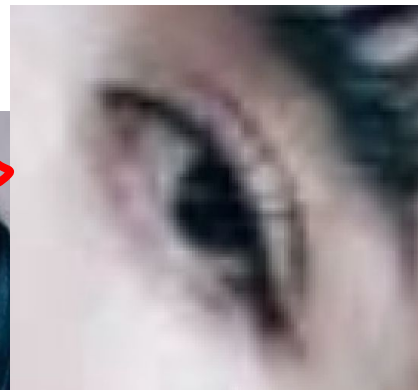
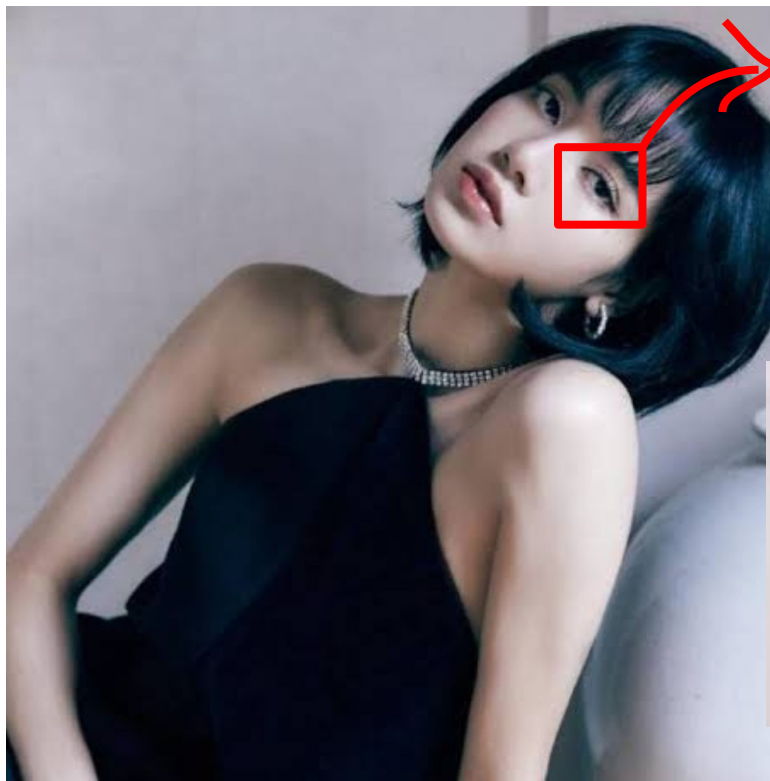
3. Interpolation - 雙線性內插法 (40%)

- 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)

