OpenCV - I

Python OpenCV Image Processing

【110上】嵌入式系統技術實驗課程

TA: 陳翰群 hanz1211.ee09@nycu.edu.tw

Introduction to OpenCV

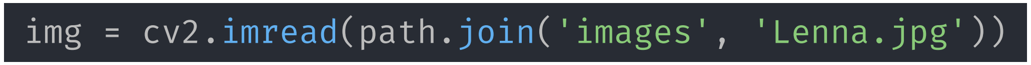
- OpenCV stands for Open Source Computer Vision
- A free computer vision library that allows you to manipulate images and videos to accomplish a variety of tasks.
- Build with C/C++, Python API is also available
- Offer some basic GUI functionality.

Related libraries

- NumPy: This library is a dependency of OpenCV's Python bindings. It provides numeric computing functionality, including efficient arrays.
- SciPy: This library is a scientific computing library that is closely related to NumPy. It is not required by OpenCV, but it is useful if you wish to manipulate data in OpenCV images.

Image I/O

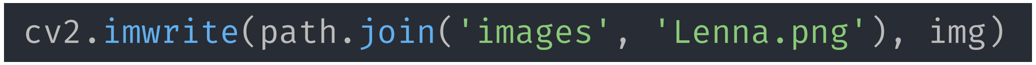
- OpenCV provides the imread function to load an image from a file and the imwrite function to write an image to a file.
- The supported formats vary—as formats can be added or removed in a custom build of OpenCV—but normally BMP, PNG, JPEG, and TIFF.
- imageIO.py shows the basic image reading and writing method.
  - Read image to matrix



```
img = cv2.imread(path.join('images', 'Lenna.jpg'))
```

Using join to make sure
different OS can find this file

  - Save matrix as image

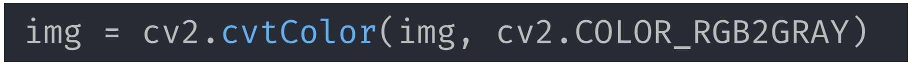


```
cv2.imwrite(path.join('images', 'Lenna.png'), img)
```

OpenCV will do the image formatting for you when specified

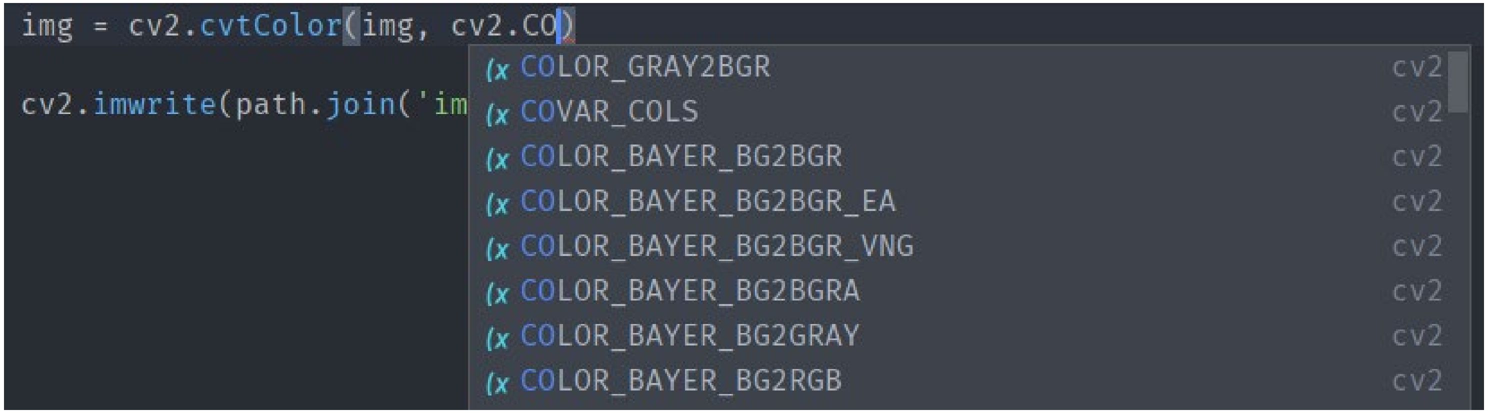
Convert Color Space

- imageIO.py
- cvtColor will accept a Mat input and a conversion method
  - This will change img to a grayscale image



```
img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
```

  - Hint: your IDE may show some suggestions as you type, passing them will cause different color conversion behaviors



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

cv2.imwrite(path.join('im
```

```
(x COLOR_GRAY2BGR            cv2
(x COVAR_COLS                cv2
(x COLOR_BAYER_BG2BGR        cv2
(x COLOR_BAYER_BG2BGR_EA     cv2
(x COLOR_BAYER_BG2BGR_VNG    cv2
(x COLOR_BAYER_BG2BGRA       cv2
(x COLOR_BAYER_BG2GRAY       cv2
(x COLOR_BAYER_BG2RGB        cv2
```

Color Space

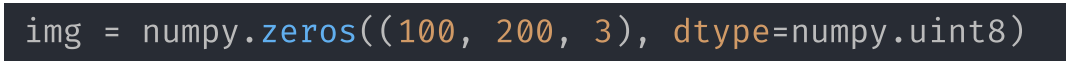
- Default color image in OpenCV is stored as BGR matrix, not RGB
- Other options are HSV/HSL, LAB, LUV color space, we may cover them later.

- In imageIO.py, you can check the image dimension with img.shape, and the pixel data type with img.dtype
  - These are the properties of numpy.ndarray
- The terminal output shows the original image has 3 channels (BGR), after converting to grayscale, drop to 1 (Brightness).
- uint8 accept value from 0~255 (0: black, 255: white)

Declare an Empty Image

- In Python, image can be edit as Numpy ndarray

- edit.py
- You can declare an empty black image by



```
img = numpy.zeros((100, 200, 3), dtype=numpy.uint8)
```

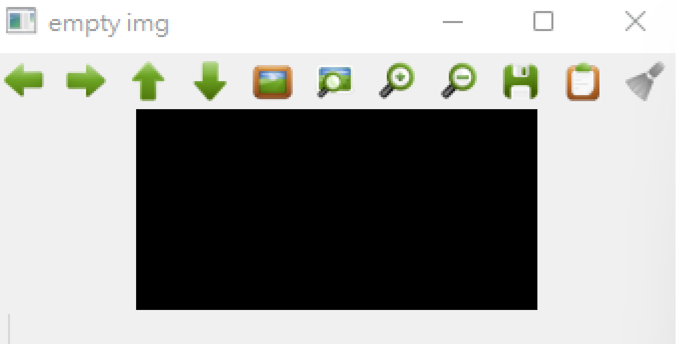
  - This image has height 100px, width 200px, and 3 channels

Display Image

- edit.py
- To show an image, use imshow(<window title>, <matrix>)
- To prevent window close immediately, add waitKey() will make program halt and wait for further input.

```
black = numpy.zeros((100, 200, 3), dtype=numpy.uint8)
cv2.imshow('empty img', black)
cv2.waitKey()
```

Press your keyboard to close window, don't click X



- Remember to clear all windows at the end of your program

```
cv2.destroyAllWindows()
```

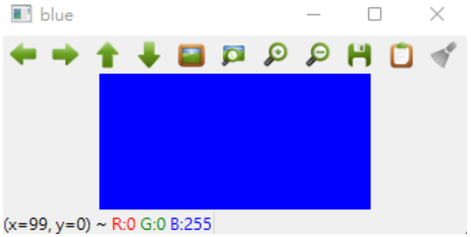
Edit Image

- edit.py
- Python object are pointers, if you want to copy and edit an image without change the original, use copy()

```
blue = black.copy()
```

- You can edit a certain value in matrix, or a part of it using list slice
  - This will change all pixels in blue channel to 255, make this image blue

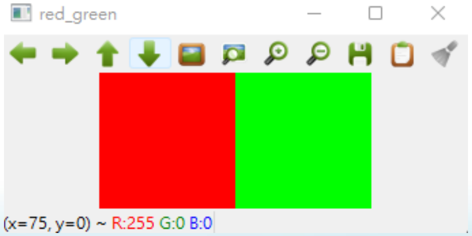
```
blue[:, :, 0] = 255
```



(x=99, y=0) ~ R:0 G:0 B:255

  - This will make a half red, half green image

```
red_green[:, :100, 2] = 255
red_green[:, 100:, 1] = 255
```



(x=75, y=0) ~ R:255 G:0 B:0

Defining regions of interest

- roi.py
- ROI can be obtained by slice as well
  - Copy the image at ROI x=100~239, y=100~239

```
face = img[100:240, 100:240, :]
```

  - Overwrite the ROI at x=0~139, y=0~139.

```
img[0:140, 0:140, :] = face
```

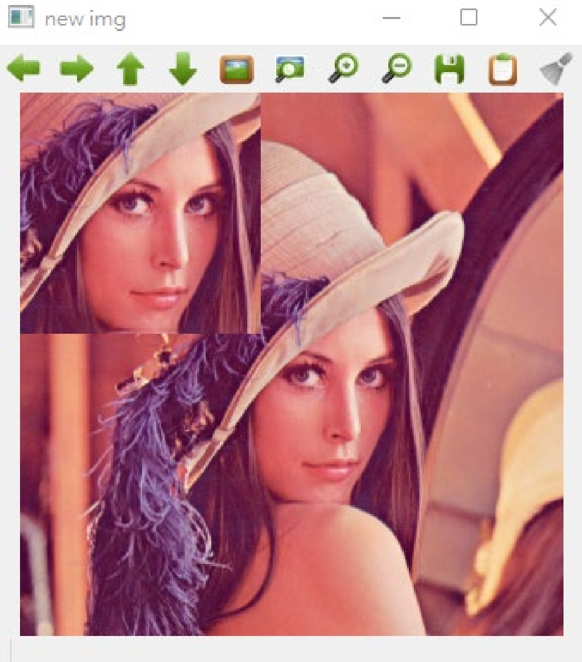


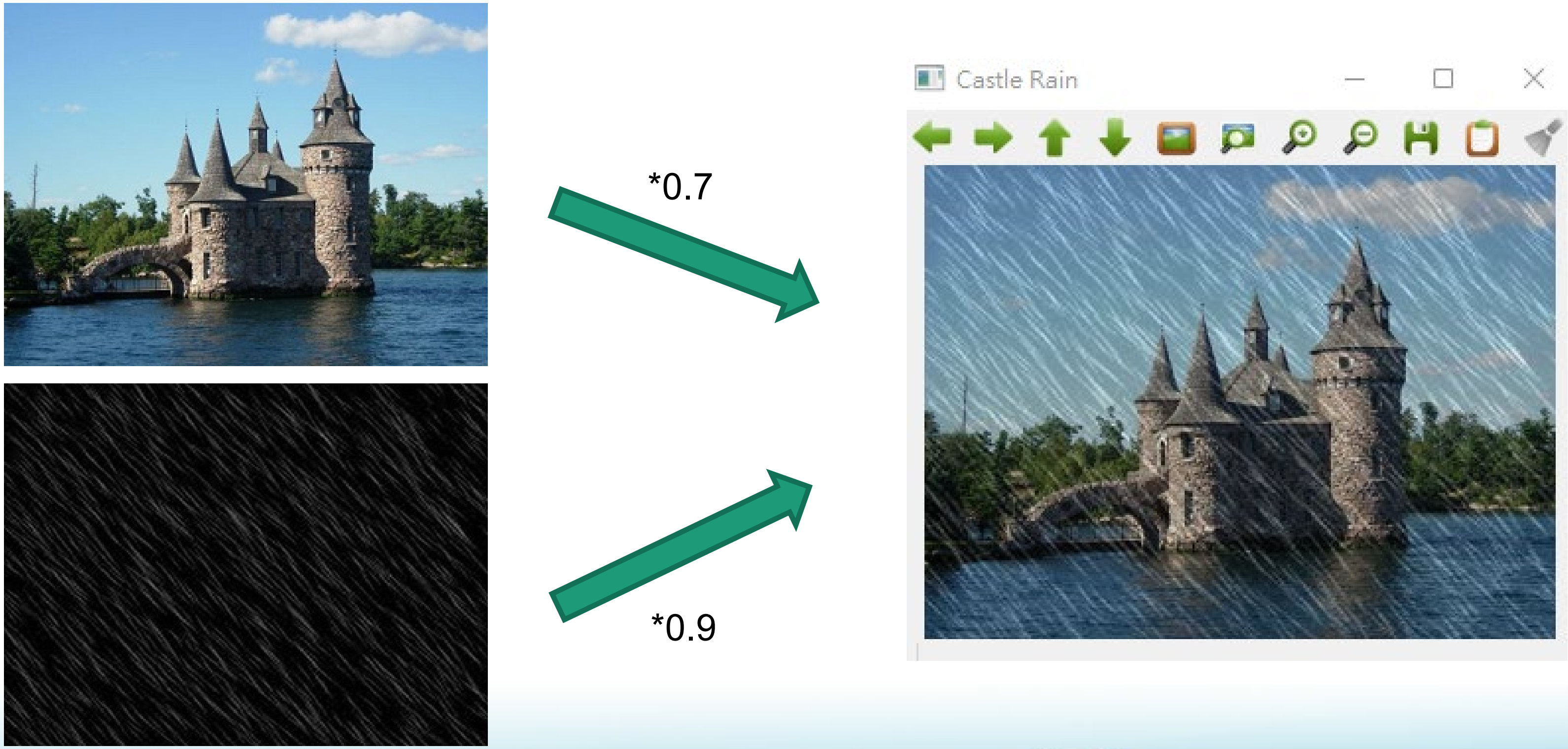
Image Calculation

- Two image must be the same size, or using ROI
- imageC = cv2.add(imageA, imageB)
  - c[i]= a[i]+b[i]
- imageC = cv2.addWeighted(imageA, k1, imageB, k2, k3)
  - c[i]= k1*a[1]+k2*b[i]+k3
- There are cv2.subtract, cv2.subtractWeighted too

Image Calculation

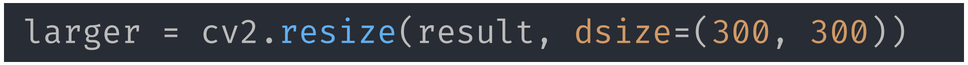
- add_image.py

```
result = cv2.addWeighted(castle, 0.7, rain, 0.9, 0.)
```



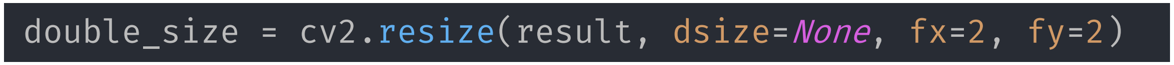
Resize Image

- add_image.py
- Use cv2.resize() to resize an image.
- You can give a specify size you want



```
larger = cv2.resize(result, dsize=(300, 300))
```

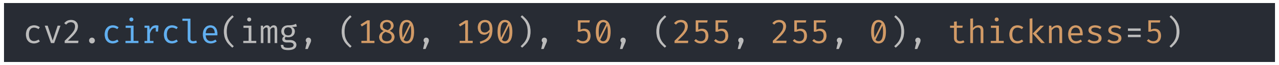
- Or give a ratio (you need to pass dsize=None)



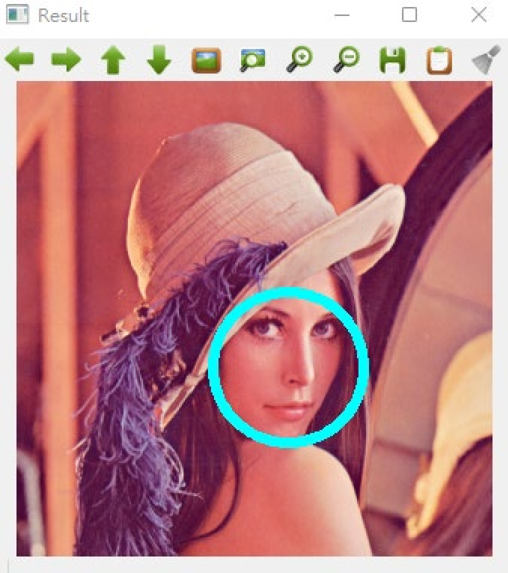
```
double_size = cv2.resize(result, dsize=None, fx=2, fy=2)
```

Drawing circles

- circle.py
- cv2.circle(img, center, radius, color, thickness=None, lineType=None, shift=None)

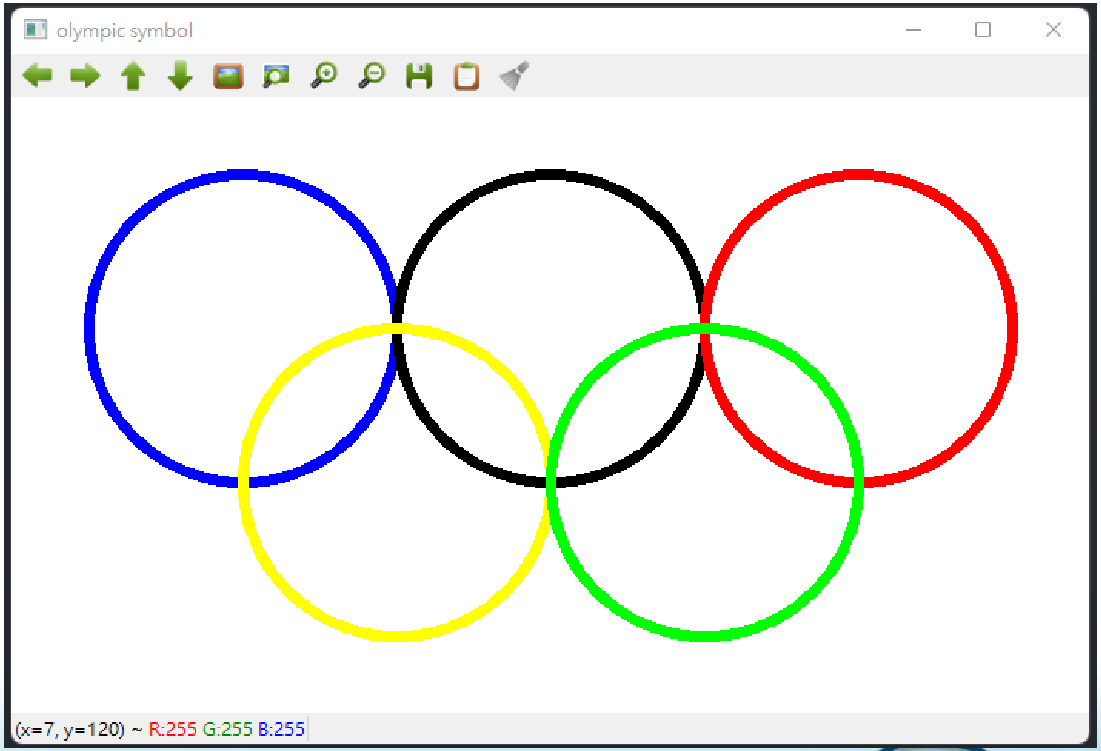


```
cv2.circle(img, (180, 190), 50, (255, 255, 0), thickness=5)
```



Lab3-1: Olympic Symbols

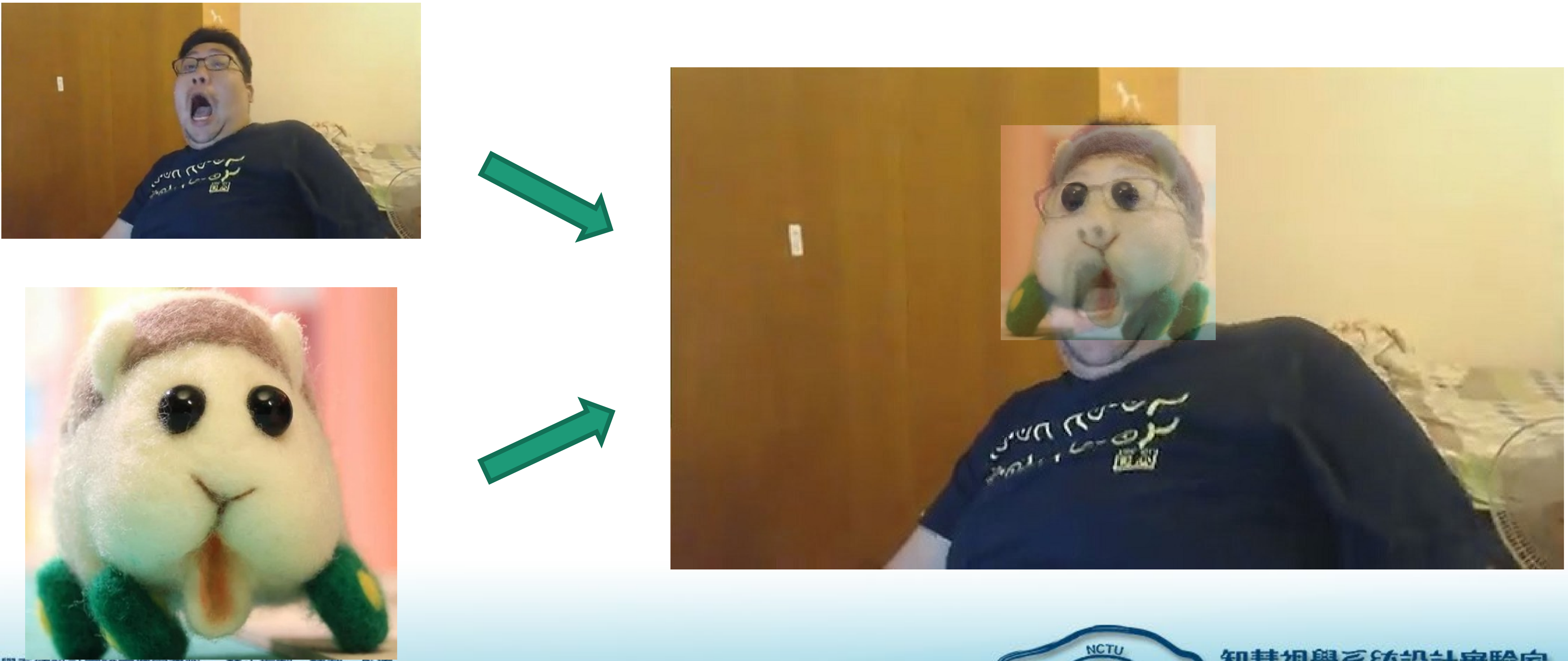
- lab3_olympic.py
- Draw an Olympic symbols with white background
  - Colors must match
  - Use whatever image size you like, but the symbols must be centered properly



(x=7, y=120) ~ R:255 G:255 B:255

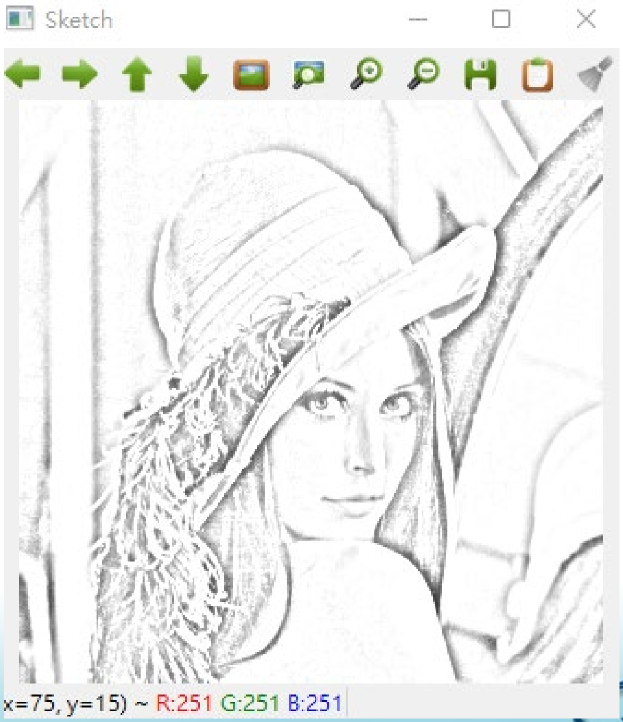
Lab3-2: Image Calculation

- lab3_imageCalc.py
- Draw an Olympic symbols with white background
  - Flip puipui first
  - Find a proper resize parameter and ROI, stack with 50% transparency



Lab3-3: Pencil Sketch Filter

- lab3_pencil.py
- Create a nice pencil sketch filter with the following steps
  - First, convert the color image to grayscale.
  - Second, apply a Gaussian blur to the grayscale image.
  - Finally, blend the grayscale image (from step 1) with the blurred image (from step 2) by using color dodge.



```
x=75, y=15) ~ R:251 G:251 B:251
```

Lab3-3: Pencil Sketch Filter

- Hint:
- Color dodge can be done by divide the image
- Use cv2.GaussianBlur() to blur image, for this pencil filter, use 21x21 kernel size is recommended
- In order to show the quotient image properly, you need to scale up the dividend value by 256, look up cv2.divide() doc for more details.

OpenCV Official Doc

- https://docs.opencv.org/4.5.3/
- These are C++ documents, but Python API should be similar

Demo

- 本次Lab以個人為單位
- 配分
  - Lab3-1 : 40%
  - Lab3-2 : 40%
  - Lab3-3 : 20%
- Demo
  - 完成Lab後，請進視訊會議舉手呼叫助教們demo
  - 多個小題可以分次demo
  - 根據助教要求呈現程式執行結果
- 最後登記時間：21:20