# Computer Vision Hw #1

B06902058 吳崇維
2019/09/11

## Homework Description

- Part1. Write a program to do the following requirement.
  - (a) upside-down lena.bmp lena.bmp
  - (b) right-side-left lena.bmp
  - (c) diagonally mirrored lena.bmp
- Part2. Write a program or use software to do the following requirement.
  - (d) rotate lena.bmp 45 degrees clockwise
  - (e) shrink lena.bmp in half
  - (f) binarize lena.bmp at 128 to get a binary image

All the problems was completed by **main.py**, which is written by **python**.

---

- **Upside-down**

Simply swap the rows from top to bottom

```
for i in range(int(length / 2)): # length: the image size = 512*512 , arr: the output array
        for j in range(length):
            arr[i][j], arr[length-1-i][j] = arr[length-1-i][j], arr[i][j]
```

- **Right-side-left**

Simply swap the rows from left to right

```
for i in range(int(length / 2)):
        for j in range(length):
            arr[j][i], arr[j][length-1-i] = arr[j][length-1-i], arr[j][i]
```

- **Diagonally mirrored**

Line mirror of top-left to bottom-right.

```
for i in range(512):
        for j in range(512):
            new_arr[511-j][i] = arr[i][j]
```

- **Rotate 45 degrees clockwise**

Create a larger array, then plot the pixels to the corresponding place.

```
rotate_arr = np.zeros((1023, 1023))
    for i in range(1023):
        for j in range(1023):
            rotate_arr[i][j] = 255
    for i in range(512):
        nx = i
        ny = 511-i
        for j in range(512):
            rotate_arr[nx+j][ny+j] = arr[i][j]
    for i in range(511):
        nx = i+1
        ny = 511-i
        for j in range(511):
            cx = nx+j
            cy = ny+j
            rotate_arr[cx][cy] = (
                    rotate_arr[cx+1][cy] +
                    rotate_arr[cx][cy+1] +
                    rotate_arr[cx-1][cy] +
                    rotate_arr[cx][cy-1] ) / 4
```

- **Shrink in half**

Pick half of the pixels of the origin image.

```
new_arr = np.zeros((512, 512))
    for i in range(512):
        for j in range(512):
            new_arr[i][j] = 255
    for i in range(128, 128+255):
        for j in range(128, 128+255):
            new_arr[i][j] = arr[i*2-256][j*2-256]
```

- **Binary Image**

Simple if-else

```
for i in range(length):
        for j in range(length):
            arr[i][j] = 0 if arr[i][j] < 128 else 255
```