

Homework 1

Basic Image Manipulation

— B05902118 陳盈如

Part 1.

A. upside-down lena.bmp

```
7  upside_down = np.zeros(image.shape, int)
8  for i in range(image.shape[0]):
9      upside_down[i] = image[image.shape[0]-1-i]
```

Seeing that we only want to put the picture upside down, don't need to worry about the pixels from left to right. Just let them be the same and make the last line of pixels become the first line, the first line become the last line.

RESULTS:



B. right-side-left lena.bmp

```
11 right_side_left = np.zeros(image.shape, int)
12 for i in range(image.shape[0]):
13     for j in range(image.shape[1]):
14         right_side_left[i, j] = image[i, image.shape[1]-1-j]
```

Seeing that it only requires left and right reversed, don't need to change the order from top to down, that is to say, the order of i . Then, make the pixel from the most left side to the most right place, step by step, left to right, and it's done.

RESULTS:



C. diagonally flip lena.bmp

```
19 diagonally_flip = np.zeros(image.shape, int)
20 for i in range(image.shape[0]):
21     for j in range(image.shape[1]):
22         diagonally_flip[i, j] = image[j, i]
```

Flip the picture by a diagonal line ($x = -y$), which we only need to exchange the values of x and y .

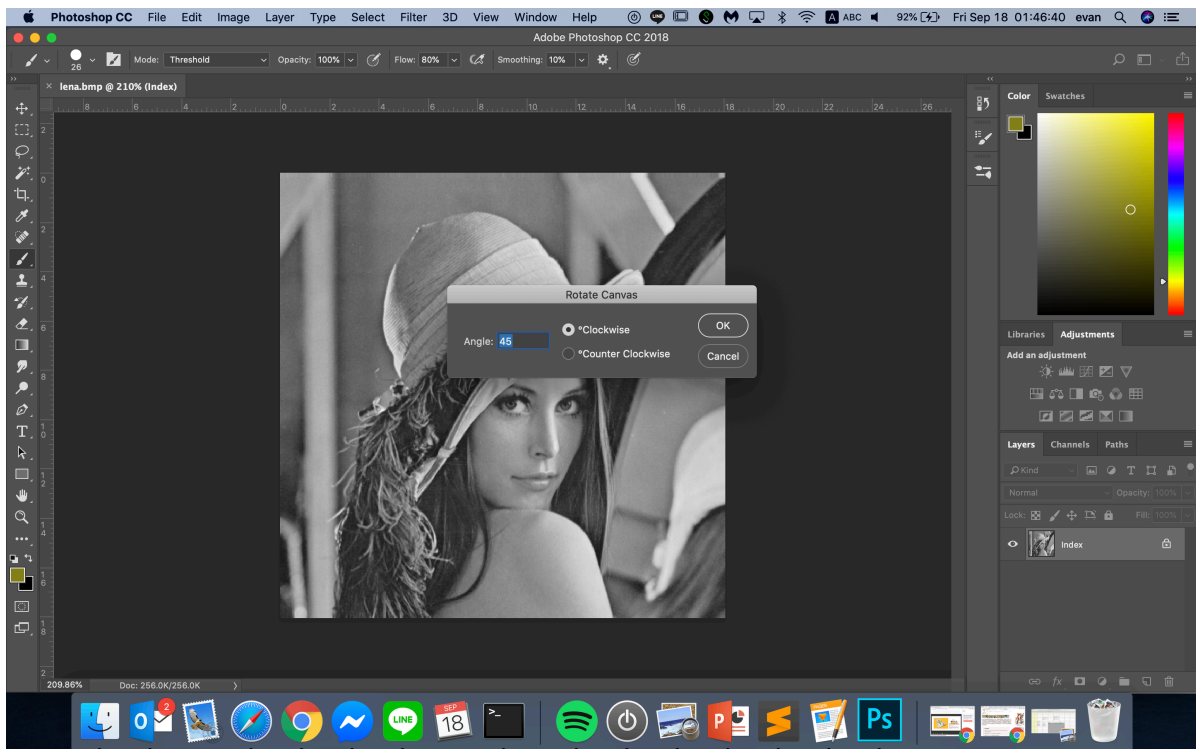
RESULTS:



Part 2.

D. rotate lena.bmp 45 degrees clockwise

Use Photoshop CC 2018 to rotate the picture.



RESULTS:



E. shrink lena.bmp in half

```
37 half_size = tuple([int(image.shape[0]/2), int(image.shape[1]/2), image.shape[2]])
38 half = np.zeros(half_size, int)
39 for i in range(int(image.shape[0]/2)):
40     for j in range(int(image.shape[1]/2)):
41         half[i, j] = image[i*2, j*2]
```

Shrink the picture in half will also reduce half of the information in the picture. If the value of x and y can be divided by 2, its new place will become $(x/2, y/2)$. If they can't, this pixel doesn't matter.

RESULTS:



Due to the fact that I shrink all results of lena pictures to the same size in this report, we cannot see the difference through the picture here. But if you click the information of the result picture, you can see the dimension of picture becomes half.

F. binarize lena.bmp at 128 to get a binary image

```
43 binarize = np.zeros(image.shape, int)
44 for i in range(image.shape[0]):
45     for j in range(image.shape[1]):
46         if (image[i][j][0] * 0.299 + image[i][j][1] * 0.587 + image[i][j][2] * 0.114) > 128:
47             binarize[i][j] = 255
48         else:
49             binarize[i][j] = 0
```

Actually, the picture is already greyscale, so we don't need to transfer the color into greyscale again. But I still did it on line 46. Then, if the value is more than 128, change it to white, if it less than 128, change it to black.

RESULTS:

