電腦視覺作業

1. upside-downlena.bmp:

   第一個是上下顛倒,為了看到像素處理的過程,我的作法是先創建一張同樣大小的圖案,再將 Y 軸向的像素進行顛倒(大小為 512 因此顛倒後為 512-i-1,i 為自訂係數,而系統似乎有規範不可單純 512-I,因此我又多減了一個 1)

```
#(a) upside-down lena.bmp
import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt


img = cv.imread("lena.bmp")
def flipp(img):
    img2 = np.zeros([512,512,3],np.uint8)
    for i in range(512):
        img2[i,:]=img[512-i-1,:]
    return img2

x = flipp(img)

plt.imshow(x)
plt.axis('off')
```

```
(-0.5, 511.5, 511.5, -0.5)
```

2. right-side-left lena.bmp:

   而左右顛倒的作法和上下完全相同,差別只在於像素處理的軸由 Y 軸改完 X 軸

```python
#(b) right-side-left lena.bmp
import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt


img = cv.imread("lena.bmp")
def flipp(img):
    img2 = np.zeros([512,512,3],np.uint16)
    for i in range(512):
        img2[:,i]=img[:,512-i-1]
    return img2

x = flipp(img)

plt.imshow(x)
plt.axis('off')
```

```
(-0.5, 511.5, 511.5, -0.5)
```

3. diagonally flip lena.bmp:

起初對角線翻轉我想了很久,後來發現某條對角線的翻轉即為原圖進行一次上下左右的翻轉,因此我多一個係數分別處理兩個軸的翻轉

```python
#(c) diagonally flip lena.bmp
import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt


img = cv.imread("lena.bmp")
def flipp(img):
    img2 = np.zeros([512,512,3],np.uint8)
    for i in range(512):
        for j in range(512):
            img2[512-i-1,j]=img[i,512-j-1]
    return img2

x = flipp(img)

plt.imshow(x)
plt.axis('off')
```

```
(-0.5, 511.5, 511.5, -0.5)
```

4. rotate lena.bmp 45 degrees clockwise:

這個可以套用 python 內建的 getRozaztionMatrix2D 功能,但做完後總會切到編框,無法完整顯示出菱形的形狀,而後我發現透過改變圖型的縮放係數可以達到完整顯示的作用,因此我將係數調整至 0.7 後終於可以完整顯示出 lena,如下圖所示

```python
#(d) rotate lena.bmp 45 degrees clockwise
import cv2 as cv

import numpy as np

import matplotlib.pyplot as plt

img = cv.imread("lena.bmp")

def rotate(img):
    (h1,w1,d1) = img.shape
    center = (h1//2,w1//2)
    R = cv.getRotationMatrix2D(center,315,0.7)
    img2 = cv.warpAffine(img,R,(512,512))

    return img2
x = rotate(img)
plt.imshow(x)
plt.axis('off')
```

```
(-0.5, 511.5, 511.5, -0.5)
```

5. shrink lena.bmp in half:

這個可以輕易套用 python 內建的 resize 功能,雖然成像看不出來差別,但可以在結果上顯示出目前的像素為(255.5,255.5)

```python
#(e) shrink lena.bmp in half
import cv2 as cv

import numpy as np

import matplotlib.pyplot as ply

img = cv.imread("lena.bmp")

imgresize = cv.resize(img,(256,256))

plt.imshow(imgresize)
plt.axis('off')
```

```
(-0.5, 255.5, 255.5, -0.5)
```

6. binarize lena.bmp at 128 to get a binary image:

這個是透過 python 的 threshold 功能來達到的,起初不曉得這功能也是在網路查了一陣子才找到,而內建的二直化有五種不同的風格,我是將閥值設定在 127,代表在 0~127 的像素會變成白色,而 128~255 則都會變成黑色

```
#(f) binarize lena.bmp at 128 to get a binary image
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
img = cv.imread("lena.bmp",0)

ret,img2 = cv.threshold(img, 127, 255, cv.THRESH_BINARY)

image = [img2]

for i in range(1):
    plt.imshow(img2,'gray')

plt.axis('off')
```

```
(-0.5, 511.5, 511.5, -0.5)
```