

# HW1 report

R10922082 林育駿

本次作業中使用 python opencv 幫助圖片的 I/O 一些處理部分。

(a) upside-down lena.bmp



在程式中作法是將原圖 pixel 將每一個 row 上下交換獲得，第一個 row 搬到最後一個 row 做交換。

```
# manipulate pixels
for i in range(imgHeight):
    for j in range(imgWidth):
        udimg[i][j]= img
```

(b) right-side-left lena.bmp



左右交換則是將原圖 pixel 每一個 column 左右交換獲得，第一個 column 搬到最後一個 row 做交換。

```
# manipulate pixels
for i in range(imgHeight):
    for j in range(imgWidth):
        rlimg[i][j]= img[i][imgWidth-j-1]
```

(c) diagonally flip lena.bmp



對原圖的 pixel 的 x,y 兩位置交換，如同對  $x=y$  的對角線做鏡射。

```
# manipulate pixels
for i in range(imgHeight):
    for j in range(imgWidth):
        diaimg[i][j]= img[j][i]
```

(d) rotate lena.bmp 45 degrees clockwise



對原圖做順時針 45 度旋轉，利用 opencv 的 api 生成旋轉矩陣再對每個點做矩陣乘法來旋轉。

```
# rotate matrix  
M = cv2.getRotationMatrix2D(center, angle, scale)  
rotated = cv2.warpAffine(img, M, (imgWidth, imgHeight))
```

(e) shrink lena.bmp in half



使用 opencv api resize 直接將圖片長寬縮小兩倍，使用內差的方式得到縮小時的 pixel 值。

```
# resize image  
resized = cv2.resize(img, dim, interpolation=cv2.INTER_AREA)
```

(f) binarize lena.bmp at 128 to get a binary image



設定門檻值當每個 pixel 的 rgb 平均大於 128 與否直接輸出為 255 或 0，達到 binarize 的效果。

```
#####
# manipulate pixels
for i in range(imgHeight):
    for j in range(imgWidth):
        if img[i][j].mean() >=128:
            binimg[i][j]= np.array([255,255,255])
        else:
            binimg[i][j]= np.array([0,0,0])
```