

CV hw1 / 電機所 R06921082 陳與賢

Part1:

Description:

利用 python 來處理 bmp 檔，分別是上下顛倒、左右顛倒、主對角線鏡射

Algorithm:

因為這個作業蠻簡單的，沒用到什麼特別的 algorithm
首先將圖檔讀成一個二維陣列，存放每一個 pixel 的灰階值
<上下顛倒> 最下排給到 output 的第一排，倒數第二排給到 output 的第二排...依此類推
<左右顛倒> 最右排給到 output 的最左排，右邊數來第二排給到 output 的左邊數來第二排...依此類推
<主對角線鏡射> x,y 值相反給 output 即可

code 如下：

```
1 from PIL import Image
2
3 img_input = Image.open('lena.bmp')
4 img_output = Image.new(img_input.mode, img_input.size )
5 pixels_input = img_input.load()
6 pixels_output = img_output.load()
7
8 # upside-down
9 for x in range(img_output.height):
10     for y in range(img_output.width):
11         pixels_output[x, y] = pixels_input[x, img_output.width - y - 1]
12 img_output.save('upside-down.bmp')
13
14 # rightside-left
15 for x in range(img_output.height):
16     for y in range(img_output.width):
17         pixels_output[x, y] = pixels_input[img_output.height - x - 1, y]
18 img_output.save('rightside-left.bmp')
19
20 # diagonally mirrored
21 for x in range(img_output.height):
22     for y in range(img_output.width):
23         pixels_output[x, y] = pixels_input[y, x]
24 img_output.save('diagonally-mirrored.png')
25
```

Result:

(一) 上下顛倒



(二) 左右顛倒



(三)主對角線鏡射



Part2:

都使用 PhotoShop 來做以下輸出

(一) 順時針旋轉 45 度

影像旋轉 -> 任 -> 順時針旋轉 45 度



(二) 縮小各半

影像尺寸 -> 長寬調整 50%



(三) 二極化

影像調整 -> 臨界值 -> 臨界值層級128

