Computer Vision homework.1

電機所 R07921101 劉定岳

Discription:

Part1 為上下反轉、左右反轉、對角線反轉處理，圖片輸出於 result 目目錄中，使用 pillow 進行基礎 IO 處理，code 檔為「hw1.py」。

Part2 為順時針 45 旋轉度、圖片長寬縮小一半、binarize 實做，其中旋轉使用 Pinta 軟體達成，縮小與 binarize 以 Python 達成，code 檔為「hw1_part2.py」。

處理後的圖片皆置於 result 資料夾中。

Part.1

How to use: 1. 請確保 hw1.py 與 result 目錄、lena.bmp 位於同目錄中。
2. 執行 $ python hw1.py。

(a) upside-down

algorithm: 建立一個新的 Image 物件(pic_ud)，利用雙層迴圈依序將 pixel 以上下相反的方式置入其中。

principal code fragment:

```
def upside_down(pic):
    width, height = pic.size
    pic_ud = Image.new(pic.mode, pic.size)
    pic_load = pic.load()
    pic_ud_load = pic_ud.load()
    for i in range(0, height):
        for j in range(0, width):
            pic_ud_load[i, j] = pic_load[i, height-1-j]
    try:
        pic_ud.save('./result/upside_down.bmp')
    except:
        print('Error! 請確認含 hw1.py 的目錄下存在 result 目錄!')
        exit()
```

Illustration 1-a: upside-down

(b) right-side-left

algorithm: 建立一個新的 Image 物件(pic_rsl)，利用雙層迴圈依序將 pixel 以左右相反的方式置入其中。

principal code fragment:

```
def right_side_left(pic):
    width, height = pic.size
    pic_rsl = Image.new(pic.mode, pic.size)
    pic_load = pic.load()
    pic_rsl_load = pic_rsl.load()
    for i in range(0, height):
        for j in range(0, width):
            pic_rsl_load[i, j] = pic_load[width-1-i, j]
    try:
        pic_rsl.save('./result/right_side_left.bmp')
    except:
        print('Error! 請確認含 hw1.py 的目錄下存在 result 目錄!')
        exit()
```

Illustration 1-b: right-side-left

(c) diagonally-mirrored
algorithm: 建立一個新的 Image 物件(pic_dm)，利用雙層迴圈依序將 pixel 以對角線相反的方式置入其中。

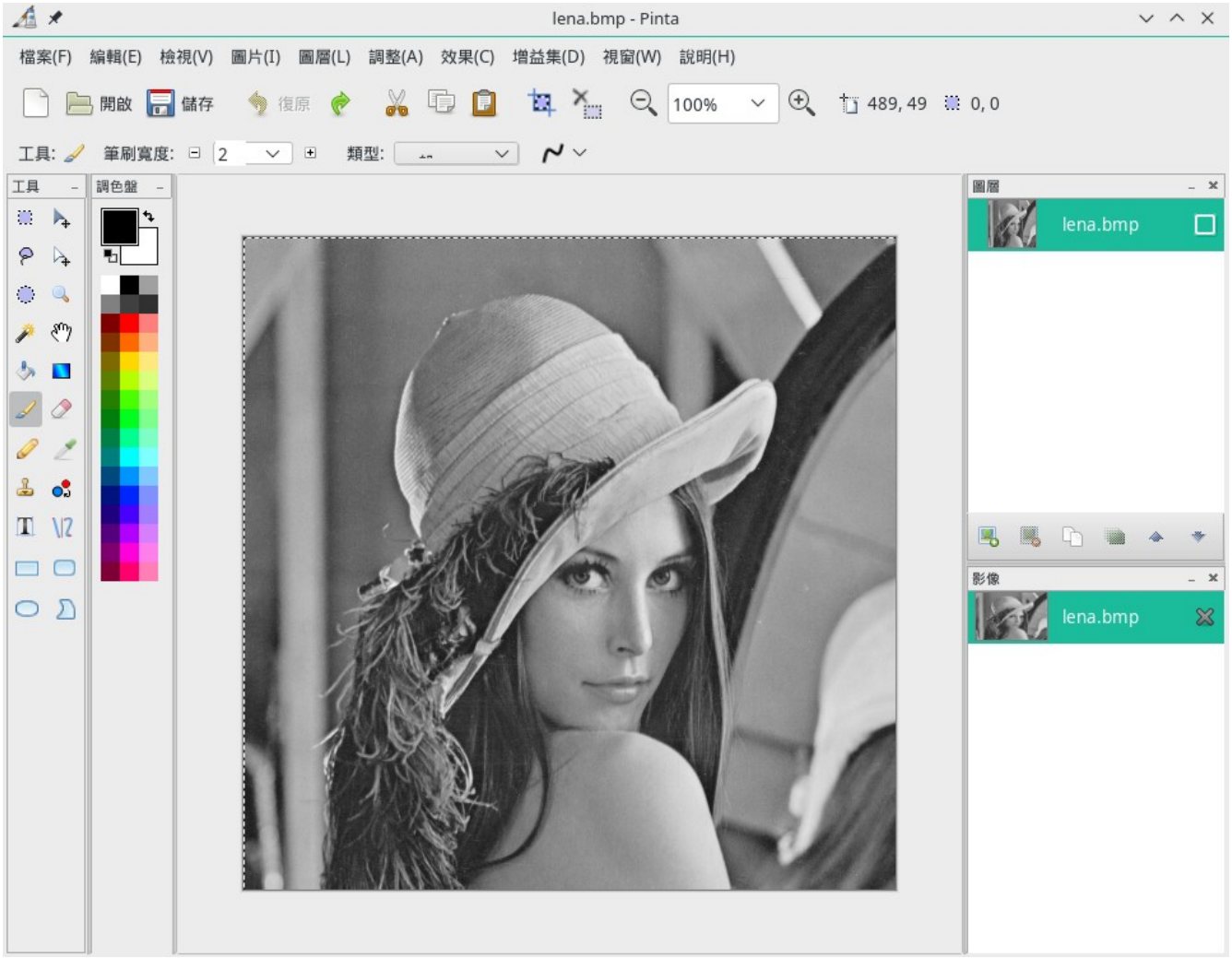
Principal code fragment:

```
def diagonally_mirrored(pic):
    width, height = pic.size
    pic_dm = Image.new(pic.mode, pic.size)
    pic_load = pic.load()
    pic_dm_load = pic_dm.load()
    for i in range(0, height):
        for j in range(0, width):
            pic_dm_load[i, j] = pic_load[j, i]
    try:
        pic_dm.save('./result/diagonally_mirrored.bmp')
    except:
        print('Error! 請確認含 hw1.py 的目錄下存在 result 目錄!')
        exit()
```

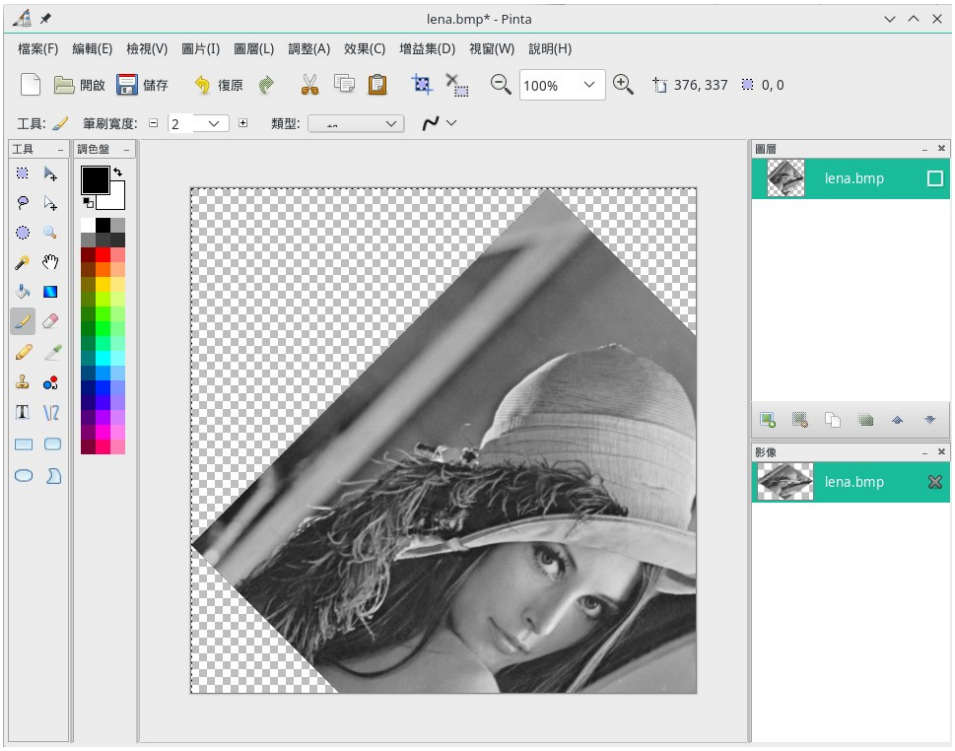
Illustration 1-c: diagonally-mirrored

Part.2

(a) rotate 45 degrees clockwise
使用 Pinta 實做，步驟如下。

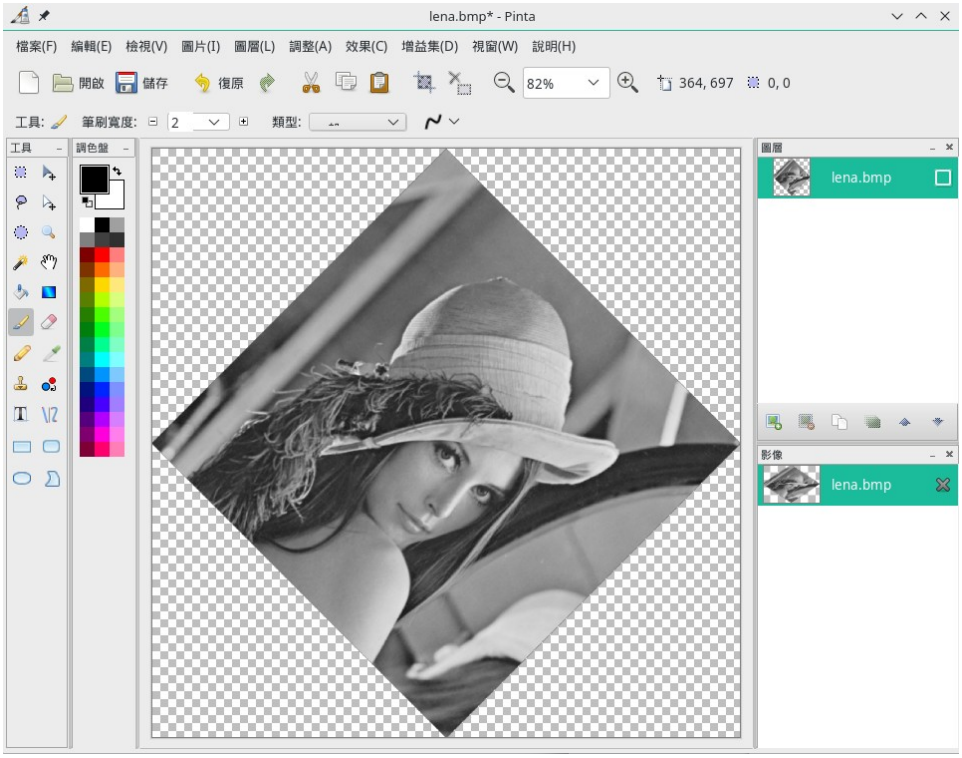


圖層(L) → 旋轉/縮放圖層 → 輸入「45」 → 確定



此時圖片超出畫布範圍，需調整畫布大小。

圖片(I) → 自動裁切



存檔即得結果。

Illustration 2-a: rotate 45 degrees clockwise

(b) shrink lena in half
使用 python 的 PIL 的 resize 函式達成，source code 附在 hw1_part2.py 中。

step1.讀取

```
wife = Image.open('lena.bmp')
```

step2.利用 resize 函式縮小

```
h, w = wife.size
shrink_wife = wife.resize((h//2, w//2))
```

step3.存檔

```
shrink_wife.save('./result/shrink.bmp')
```

Illustration 2-b: shrink

(c)binarize lena at 128 to get a binary image
使用 python 的 PIL 讀取圖檔後，利用雙層迴圈遍歷整張圖進行修改，source code 附在 hw1_part2.py 中。

step1.讀取

```
wife = Image.open('lena.bmp')
```

step2.將圖檔與設定傳入 binary 函式

```
binarize(wife, 128)
```

step3.為接下來的運算初始化參數

```
width, height = pic.size
pic_binarize = Image.new(pic.mode, pic.size)
pic_load = pic.load()
pic_binarize_load = pic_binarize.load()
```

step4.檢查全部 pixel 顏色，若>=T(此題中 T = 128)即設為白(255)，反之設為黑(0)

```
for i in range(0, height):
    for j in range(0, width):
        if(pic_load[i, j]>=T):
            pic_binarize_load[i, j]=255
        else:
            pic_binarize_load[i, j]=0
```

step5.存檔

```
try:
    pic_binarize.save('./result/binarize.bmp')
except:
    print('Error! 請確認含 hw1.py 的目錄下存在 result 目錄!')
    exit()
```

Illustration 2-c: binarize