

Computer Vision HW1 R10921A10 電機碩一 廖彥朋

Part1. Write a program to do the following requirement.

- (a) upside-down lena.bmp
- (b) right-side-left lena.bmp
- (c) diagonally flip lena.bmp

Part2. Write a program or use software to do the following requirement.

- (d) rotate lena.bmp 45 degrees clockwise
- (e) shrink lena.bmp in half
- (f) binarize lena.bmp at 128 to get a binary image

原始圖片:



實作結果:

| a | b | c |
|---|---|---|
| | | |
| d | e | f |
| | | |

程式碼(Python):

```
1  import cv2
2  import numpy as np
3
4  def upside_down(img):
5      img_output = np.zeros(img.shape, int)
6      for i in range(img.shape[0]):
7          img_output[i, :] = img[img.shape[0] - i - 1, :]
8      return img_output
9
10 def right_side_left(img):
11     img_output = np.zeros(img.shape, int)
12     for j in range(img.shape[1]):
13         img_output[:, j] = img[:, img.shape[1] - j - 1]
14     return img_output
15
16 def diagonally_flip(img):
17     img_output = np.zeros(img.shape, int)
18     for i in range(img.shape[0]):
19         for j in range(img.shape[1]):
20             img_output[i, j] = img[img.shape[0] - i - 1, img.shape[1] - j - 1]
21     return img_output
22
23 def rotate45(img, angle = -45, center = None, scale = 1.0):
24     (h, w) = img.shape[:2]
25     if center is None:
26         center = (w / 2, h / 2)
27     M = cv2.getRotationMatrix2D(center, angle, scale)
28     img_output = cv2.warpAffine(img, M, (w, h))
29     return img_output
30
31 def shrink_half(img):
32     img_output = cv2.resize(img, (256, 256))
33     return img_output
34
35 def binarize(img):
36     th, img_output = cv2.threshold(img, 128, 255, cv2.THRESH_BINARY)
37     return img_output
38
39 img = cv2.imread("lena.bmp")
40 img_output = upside_down(img)
41 cv2.imwrite('upside_down.jpg', img_output)
42 img_output = right_side_left(img)
43 cv2.imwrite('right_side_left.jpg', img_output)
44 img_output = diagonally_flip(img)
45 cv2.imwrite('diagonally_flip.jpg', img_output)
46 img_output = rotate45(img)
47 cv2.imwrite('rotate45.jpg', img_output)
48 img_output = shrink_half(img)
49 cv2.imwrite('shrink_half.jpg', img_output)
50 img_output = binarize(img)
51 cv2.imwrite('binarize.jpg', img_output)
52
```

程式碼簡介:

一開始先引入 OpenCV 與 NumPy 的 Python 模組。OpenCV 主要是用於讀取以及寫入圖片檔(cv2.imread/cv2.imwrite)；NumPy 函式庫用以實現快速操作多維陣列的運算。其中，img.shape 指令用於得知圖片的三個維度，前兩個維度依序為圖片的高度與寬度，第三個維度則是圖片的 channel。

a. upside-down lena.bmp

使用 for 迴圈將圖片進行上下顛倒。

b. right-side-left lena.bmp

使用 for 迴圈將圖片進行左右顛倒。

c. diagonally flip lena.bmp

使用 2 層 for 迴圈，先將圖片進行左右顛倒，再將左右顛倒的圖片進行上下顛倒，便實現了將圖片對角線鏡射的目的。

d. rotate lena.bmp 45 degrees clockwise

使用 cv2.getRotationMatrix2D 指令並設定參數以得到旋轉矩陣，第一個參數為旋轉中心(center = (w / 2, h / 2))，第二個參數是旋轉角度(-45 代表順時針轉 45 度)，第三個參數為旋轉後圖片的縮放比例(1.0)。再透過 cv2.warpAffine 指令將得到的旋轉矩陣對圖片進行旋轉。

e. shrink lena.bmp in half

使用 cv2.resize(img,(256,256))指令將原本 512x512 大小的圖片縮小到一半(256x256)。

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

使用 cv2.threshold 指令並設定參數，第一個參數為輸入的灰階影像(img)，第二個參數是用來對像素值進行分類的最小門檻值(128)，第三個參數為最大門檻灰階值(255)，第四個參數為二值化的演算類型(cv2.THRESH_BINARY)。