Computer Vision 2017FALL HW#1 R05525130 曹珉齊

Python

# construct the argument parse

ap = argparse.ArgumentParser()

ap.add\_argument("image", help="path to the input image")

args = ap.parse\_args()

#執行時輸入 python 程式 圖檔 即可執行

使用openCV處理image I/O

img = cv2.imread(args.image)

cv2.imwrite("upside down.png", udo)

第一小題

1. upside down

# 建立一個跟原圖一樣大的0矩陣

udo = np.zeros(img.shape, np.uint8)

# 將每個column的pixel值上下交換

for i in range(iW):

for j in range(iH):

udo[j,i,0] = img[iH-j-1,i,0]

udo[j,i,1] = img[iH-j-1,i,1]

udo[j,i,2] = img[iH-j-1,i,2]

1. left side right

lro = np.zeros(img.shape, np.uint8)

# 將每個row的pixel值左右交換

for i in range(iW):

for j in range(iH):

lro[j,i,0] = img[j,iW-i-1,0]

lro[j,i,1] = img[j,iW-i-1,1]

lro[j,i,2] = img[j,iW-i-1,2]

1. diagonal mirrored

dmo = np.zeros(img.shape, np.uint8)

# 將每個pixel的x,y值互換

for i in range(iW):

for j in range(iH):

dmo[i,j,0] = img[j,i,0]

dmo[i,j,1] = img[j,i,1]

dmo[i,j,2] = img[j,i,2]

第二小題

1. rotate 45 degree clockwise

# 使用opencv內建function getRotationMatrix

M = cv2.getRotationMatrix2D((iW/2,iH/2),45,1)

# 將矩陣的資料轉進圖檔

output = cv2.warpAffine(img,M,(iW,iH))

1. shrink half

# fx,fy 為長寬放大或縮小的比例

# interpolation的方法是google到推薦的函式

output1 = cv2.resize(img,None,fx=0.5,fy=0.5, interpolation = cv2.INTER\_AREA)

1. binarize at 128

# 讀檔多一個argument是指定讀入gray scale

bimg = cv2.imread(args.image,0)

# 設定threshold

thresh = 128

maxValue = 255

th, output2 = cv2.threshold(bimg, thresh, maxValue, cv2.THRESH\_BINARY);