Computer Vision I \_2018

Homework assignment #5

R07522717機械所製造組碩一 林温雅

Part1（此次作業僅one part）

Description:

Write programs which do binary morphological dilation, erosion, opening, closing, and hit-and-miss transform on a binary image.

Algorithm:

1. Dilation:

獲得kernel與輸入image的行列數以後，製作一個『外擴』圖檔，為的是處理kernel在尋訪時超出邊界的情況（以image為512\*512, kernel為5\*5來說，暫存圖檔為516\*516，暫存圖檔為往上、往下、往左、往右分別外擴兩列/行，外擴新增的pixel值為0，才不會影響後續取max的計算，中間則就是原本輸入圖檔的值）。

接著再進行計算，kernel設計為[[np.nan,0,0,0,np.nan], [0,0,0,0,0], [0,0,0,0,0], [0,0,0,0,0], [np.nan,0,0,0,np.nan]]。在計算時會忽略遇到nan的運算，因此計算結果即為3553形狀kernel中的最大值，將最大值填入kernel中心覆蓋的某p點。

特別的是，在dilation進行convolution，因此kernel須先flip 180度，但作業指定的是點對稱kernel，有沒有flip不對結果造成影響。

1. Erosion:

獲得kernel與輸入image的行列數以後，製作一個『外擴』圖檔，為的是處理kernel在尋訪時超出邊界的情況（以image為512\*512, kernel為5\*5來說，暫存圖檔為516\*516，暫存圖檔為往上、往下、往左、往右分別外擴兩列/行，外擴新增的pixel值為255，才不會影響後續取min的計算，中間則就是原本輸入圖檔的值。特別注意這邊外擴pixel的值為255，而在dilation function內是0）。

接著再進行計算，kernel設計為[[np.nan,0,0,0,np.nan], [0,0,0,0,0], [0,0,0,0,0], [0,0,0,0,0], [np.nan,0,0,0,np.nan]]。在計算時會忽略遇到nan的運算，因此計算結果即為3553形狀kernel中的最小值，將最小值填入kernel中心覆蓋的某p點。

1. Opening:

對影像進行GrayScale\_Erosion後再進行GrayScale\_Dilation

1. Closing:

對影像進行GrayScale\_Dilation後再GrayScale\_Erosion

Parameters:

1. In function “dilation”:

img\_rows, img\_columns #輸入圖檔的行列數

ker\_rows, ker\_columns #kernel的行列數

row\_dist, column\_dist #計算kernel中心距離邊界有多遠，主要目的是看原始圖檔要擴大多少

temp\_img #原始圖檔擴大後的暫存圖檔

kernel\_flip #flip 180度後的kernel

new\_img #新圖檔準備接受dilation後的圖

i,j #迴圈計數用參數

1. In function “erosion”:

img\_rows, img\_columns #輸入圖檔的行列數

ker\_rows, ker\_columns #kernel的行列數

row\_dist, column\_dist #計算kernel中心距離邊界有多遠，主要目的是看原始圖檔要擴大多少

temp\_img #原始圖檔擴大後的暫存圖檔

new\_img #新圖檔準備接受Erosion後的圖

i,j #迴圈計數用參數

1. Outside of function

original\_img #讀取原始圖檔

kernel #作業指定會用到的kernel矩陣

Principal code fragment:

def GrayScale\_Dilation(img, ker):

#獲得輸入圖檔之行列數

img\_rows, img\_columns = img.shape

#獲得kernel之行列數

ker\_rows, ker\_columns = ker.shape

#計算kernel中心距離邊界有多遠，為的是擴大原始圖檔，方便後續迴圈處理

row\_dist, column\_dist = int((ker\_rows-1)/2), int((ker\_columns-1)/2)

#根據上述計算，製作一個比原始圖檔大的暫存圖檔，以img為512\*512, kernel為5\*5來說，暫存圖檔為516\*516，暫存圖檔為往上、往下、往左、往右分別外擴兩列/行，外擴新增的pixel值另為0，中間則就是原本輸入圖檔的值

#dilation要找最大的，所以外擴的填0

temp\_img = np.zeros((img\_rows+2\*row\_dist, img\_columns+2\*column\_dist), np.int)

temp\_img[row\_dist:img\_rows+row\_dist, column\_dist:img\_columns+column\_dist] = img

#製作一個新圖檔準備接受dilation後的圖

#為了for迴圈裡面index好寫，這邊一樣把new\_img改成擴大後的，之後再來裁，和hw4做法有一點點不一樣

new\_img = np.zeros((img\_rows+2\*row\_dist, img\_columns+2\*column\_dist), np.int)

#為了矩陣相乘，先flip kernel，erosion不用這樣

kernel\_flip = np.flip(ker)

#進行dilation計算

for i in range(row\_dist, img\_rows+row\_dist):

for j in range(column\_dist, img\_columns+column\_dist):

new\_img[i, j] = np.nanmax(temp\_img[i-row\_dist: i+row\_dist+1, j-column\_dist: j+column\_dist+1]+kernel\_flip)

new\_img = new\_img[row\_dist:img\_rows+row\_dist, column\_dist:img\_columns+column\_dist]

return new\_img

def GrayScale\_Erosion(img, ker):

#獲得輸入圖檔之行列數

img\_rows, img\_columns = img.shape

#獲得kernel之行列數

ker\_rows, ker\_columns = ker.shape

#計算kernel中心距離邊界有多遠，為的是擴大原始圖檔，方便後續迴圈處理

row\_dist, column\_dist = int((ker\_rows-1)/2), int((ker\_columns-1)/2)

#根據上述計算，製作一個比原始圖檔大的暫存圖檔，以img為512\*512, kernel為5\*5來說，暫存圖檔為516\*516，暫存圖檔為往上、往下、往左、往右分別外擴兩列/行，外擴新增的pixel值另為0，中間則就是原本輸入圖檔的值

#erosion要找最小的，所以外擴的填255

temp\_img = 255 \* np.ones((img\_rows+2\*row\_dist, img\_columns+2\*column\_dist), np.int)

temp\_img[row\_dist:img\_rows+row\_dist, column\_dist:img\_columns+column\_dist] = img

#製作一個新圖檔準備接受dilation後的圖

#為了for迴圈裡面index好寫，這邊一樣把new\_img改成擴大後的，之後再來裁，和hw4做法有一點點不一樣

new\_img = 255\*np.ones((img\_rows+2\*row\_dist, img\_columns+2\*column\_dist), np.int)

#進行erosion計算

for i in range(row\_dist, img\_rows+row\_dist):

for j in range(column\_dist, img\_columns+column\_dist):

new\_img[i, j] = np.nanmin(temp\_img[i-row\_dist: i+row\_dist+1, j-column\_dist: j+column\_dist+1]-ker)

new\_img = new\_img[row\_dist:img\_rows+row\_dist, column\_dist:img\_columns+column\_dist]

Resulting images

gray\_scale\_dilation



gray\_scale\_erosion



gray\_scale\_opening



gray\_scale\_closing

