Computer Vision 2017FALL HW#4 R05525130 曹珉齊

Python

Import numpy opencv(for reading and writing)

Binary image為之前作業2所生成的圖檔，檔名為lena\_bin\_128.png

所用的kernel為octogonal 3-5-5-5-3 kernel 如下

[0,1,1,1,0

1,1,1,1,1

1,1,1,1,1

1,1,1,1,1

0,1,1,1,0]

hit and miss 所用的j, k kernel則如下

[0,0,0 [0,1,1

1,1,0 0,1,0

1,0,0] 0,0,0]

1. Dilation

首先建立一個與原圖同樣大小全黑圖

i\_dia = np.zeros(img.shape, np.int)

再來對原圖中的每個白點做運算

for i in range(kW/2, iW-kW/2):

for j in range(kH/2, iH-kH/2):

if img[j,i] > 0:

iW, iH 為圖的寬與高

kW, kH 為kernel的寬與高

以那個白點為原點將kernel為白的地方貼上去

for ki in range(kW):

for kj in range(kH):

if kernel[kj,ki] != 0:

i\_dia[j-kH/2+kj, i-kW/2+ki] = kernel[kj,ki]

回傳i\_dia後結束



1. Erosion

步驟類似dilation，不同處為對原圖中與kernel中心處一樣的點做運算

此kernel中心為白點

if img[j,i] == kernel[kH/2,kW/2]:

flag = True

對每個運算的點檢查以kernel中心為原點貼上後是否吻合白點

for ki in range(kW):

for kj in range(kH):

if img[j-kH/2+kj, i-kW/2+ki] != kernel[kj,ki] and kernel[kj,ki] > 0:

flag = False

吻合則將此點設為白色反之黑色

if flag:

i\_ero[j,i] = 255



1. Opening

如講義所說對原圖先作erosion再做dilation即可



1. Closing

如講義所說對原圖先做dilation再做erosion即可



1. hit and miss

分為三個步驟

第一個步驟將原圖與j kernel做erosion

第二個步驟為將原圖的補集與k kernel做erosion

i\_c = -img + 255

第三個步驟將前兩個步驟所得的圖做交集，將兩圖相加除2後等於255的點設 為白點即可。

i\_hnm = ((ij + ik)/2 == 255) \* 255

