Homework 5 R08921005 黃國郡

# Answer

lena.bmp

(a) dilation_lena.bmp

(b) erosion_lena.bmp

(c) opening_lena.bmp

(d) closing_.bmp

Homework 5 R08921005 黃國郡

# Description

1. Problem Formation

   I. Write programs which do gray-scale morphology on a gray-scale image(lena.bmp):

   (a) Dilation (b) Erosion (c) Opening (d) Closing

   II. Please use the octogonal 3-5-5-5-3 kernel.

2. Method of Algorithms

   (a) Dilation:

   $$(f \oplus k)(x) = \max\{f(x-z) + k(z) \mid z \in K\}$$

   (b) Erosion:

   $$(f \ominus k)(x) = \min\{f(x+z) - k(z) \mid z \in K\}$$

   (c) Opening:

   $$A \circ K = (A \ominus K) \oplus K$$

   (d) Closing:

   $$A \bullet K = (A \oplus K) \ominus K$$

3. 本次作業使用 python, IDE 為 Spyder

4. Source Code [HW5.py]說明如下

```python
# -*- coding: utf-8 -*-
"""
Created on Sun Oct 20 10:32:58 2019

@author: vincent黃國郡
"""
from PIL import Image
import numpy as np
import myMorphology
kernel = np.array([
    [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]])
centerKernel = tuple([x // 2 for x in kernel.shape])

if __name__ == '__main__':
    lena = Image.open("lena.bmp")
    # HW5a)
    dilation_lena = myMorphology.dilation(lena, kernel, centerKernel)
    dilation_lena.save('dilation_lena.bmp')

    # HW5(b)
    erosion_lena = myMorphology.erosion(lena, kernel, centerKernel)
    erosion_lena.save('erosion_lena.bmp')

    # HW5(c)
    opening_lena = myMorphology.opening(lena, kernel, centerKernel)
    opening_lena.save('opening_lena.bmp')

    # HW5(d)
    closing_lena = myMorphology.closing(lena, kernel, centerKernel)
    closing_lena.save('closing_lena.bmp')

```

Import 需要用到的 library

宣告 kernel[35553]

定義 kernel 中心點

呼叫我的 myMorphology.py 副函式進行處理作業

5. Source Code (副程式) [myMorphology.py]說明如下

```python
def dilation(originalImage, kernel, centerKernel):
    dilationImage = Image.new('L', originalImage.size)
    for r in range(originalImage.size[0]):
        for c in range(originalImage.size[1]):
            localMaxPixel = 0
            for ii in range(kernel.shape[0]):
                for jj in range(kernel.shape[1]):
                    if (kernel[ii, jj]):
                        targetX = r + (ii - centerKernel[0])
                        targetY = c + (jj - centerKernel[1])
                        if ((0 <= targetX < originalImage.size[0])
                                and (0 <= targetY < originalImage.size[1])):
                            originalPixel = originalImage.getpixel(
                                (targetX, targetY))
                            localMaxPixel = max(originalPixel, localMaxPixel)
            dilationImage.putpixel((r, c), localMaxPixel)
    return dilationImage


def erosion(originalImage, kernel, centerKernel):
    erosionImage = Image.new('L', originalImage.size)
    for r in range(originalImage.size[0]):
        for c in range(originalImage.size[1]):
            localMinPixel = 255
            for ii in range(kernel.shape[0]):
                for jj in range(kernel.shape[1]):
                    if (kernel[ii, jj]):
                        targetX = r + (ii - centerKernel[0])
                        targetY = c + (jj - centerKernel[1])
                        if ((0 <= targetX < originalImage.size[0])
                                and (0 <= targetY < originalImage.size[1])):
                            originalPixel = originalImage.getpixel(
                                (targetX, targetY))
                            localMinPixel = min(originalPixel, localMinPixel)
            erosionImage.putpixel((r, c), localMinPixel)
    return erosionImage


def opening(originalImage, kernel, centerKernel):
    return dilation(erosion(originalImage, kernel, centerKernel), kernel, centerKernel)


def closing(originalImage, kernel, centerKernel):
    return erosion(dilation(originalImage, kernel, centerKernel), kernel, centerKernel)
```

先產生一張跟 input 一樣大的 8bit original Image

初始一個 localMaxPixel 來抓 kernel 對應到的方框最小值

掃過每一個 pixel kernel 對應上的 each pixel 來抓最大值

最後把 localMaxPixel 值填回去 kenel 中心點作用的 Pixel

如同 dilation 程序，只是換抓取 localmin，最小值填回該中心 pixel 值

先 erosion 再 dilation

先 dilation 再 erosion