電腦視覺作業報告

**Homework 8**

學號：D03922030

姓名：林君憲

中華民國103年12月16日

程式語言：Python

程式名稱：NoiseRemoval.py

說明：

1. 本程式會讀入lena.bmp影像檔，將其做Gaussian noise及Salt and pepper noise，接著依序將上開兩個有雜訊的影像，kernel採3x3及5x5，分別做box filter、median filter、opening followed by closing及closing followed by opening。
2. 引用Python之Pillow影像程式庫（http://pillow.readthedocs.org/en/latest/index.html）來處理圖檔讀寫的工作。
3. 生成雜訊的函式有兩個，一個是對每個像素，依據Gaussian分佈產生介於0到1之間的亂數值，然後將它分別放大10倍及30倍，再分別加到原始圖片中。程式碼片段如下：

def **GaussNoiseImage**(image, amplitude):

im = image.copy()

width = im.size[0]

height = im.size[1]

noiseIm = Image.new(*"L"*,(width, height), 0)

for i in range(width):

for j in range(height):

pix\_val = im.getpixel((i,j))

gn\_pix = pix\_val + (random.gauss(0,1) \* amplitude)

if gn\_pix < 0:

gn\_pix = 0

if gn\_pix > 255:

gn\_pix = 255

noiseIm.putpixel((i,j), gn\_pix)

return noiseIm

1. 另一個則是隨機產生黑和白的雜訊點，雜訊點是依據傳入的閾值與所產生的亂數來決定是白點、黑點或是保留原強度。程式碼片段如下：

def **SNPNoiseImage**(image, threshold):

im = image.copy()

width = im.size[0]

height = im.size[1]

noiseIm = Image.new(*"L"*,(width, height), 0)

for i in range(width):

for j in range(height):

rnd = random.uniform(0, 1)

pix\_value = im.getpixel((i,j))

if threshold > rnd:

noiseIm.putpixel((i,j), 0)

else:

if 1 - threshold < rnd:

noiseIm.putpixel((i,j), 255)

else:

noiseIm.putpixel((i,j), pix\_value)

return noiseIm

1. 由於Python的range()函數會產生的列舉項目會從0開始到n – 1，所以在設定box邊界時，上界值要加1。否則，會因為少估計一個pixel而導致輸出圖片亮度變暗。Box Filter的程式碼片段如下：

def **BoxFilter**(image, n):

im =image.copy()

halfN = n / 2

normalizeCoefficient = n \* n

width = im.size[0]

height = im.size[1]

W = width - halfN

H = height - halfN

filteredIm = Image.new(*"L"*,(width, height), 255)

for i in range(halfN, H):

for j in range(halfN, W):

boxSum = 0

for x in range(-halfN, halfN+1):

for y in range(-halfN, halfN+1):

boxSum = boxSum + im.getpixel((x+i,y+j)) / normalizeCoefficient

filteredIm.putpixel((i,j), boxSum)

return filteredIm

1. Median filter程式碼如下。與box filter相同的是，因為Python語言的特性，所以在設定上界邊緣時，記得要加1，以免少算一個pixel導致輸出圖片會較暗。

def **medianFilter**(image, n):

im =image.copy()

halfN = n / 2

kernelLen = n \* n

width = im.size[0]

height = im.size[1]

W = width - halfN

H = height - halfN

filteredIm = Image.new(*"L"*,(width, height), 255)

for i in range(halfN, H):

for j in range(halfN, W):

boxList = []

for x in range(-halfN, halfN+1):

for y in range(-halfN, halfN+1):

boxList.append(im.getpixel((x+i,y+j)))

boxList.sort()

filteredIm.putpixel((i,j), boxList[kernelLen/2])

return filteredIm

1. Opening followed by closing及closing followed by opening則是沿用Homework 5的gray scale erosion及gray scale dilation函式來產生。Kernel是3-5-5-5-3。
2. 結果圖片共28張，如下所列。

Gaussian noise image, amplitude = 10



Gaussian noise image, amplitude = 30



Gaussian noise image (amplitude = 10) after 3x3 box filter



Gaussian noise image (amplitude = 10) after 5x5 box filter



Gaussian noise image (amplitude = 10) after 3x3 median filter



Gaussian noise image (amplitude = 10) after 5x5 median filter

Gaussian noise image (amplitude = 10) after opening followed by closing.



Gaussian noise image (amplitude = 10) after closing followed by opening.



Gaussian noise image (amplitude = 30) after 3x3 box filter



Gaussian noise image (amplitude = 30) after 5x5 box filter



Gaussian noise image (amplitude = 30) after 3x3 median filter



Gaussian noise image (amplitude = 30) after 5x5 median filter



Gaussian noise image (amplitude = 30) after opening followed by closing.



Gaussian noise image (amplitude = 30) after closing followed by opening.



Salt and pepper noise image, threshold = 0.05



Salt and pepper noise image, threshold = 0.1



Salt and pepper noise image (threshold = 0.05) after 3x3 box filter.



Salt and pepper noise image (threshold = 0.05) after 5x5 box filter.



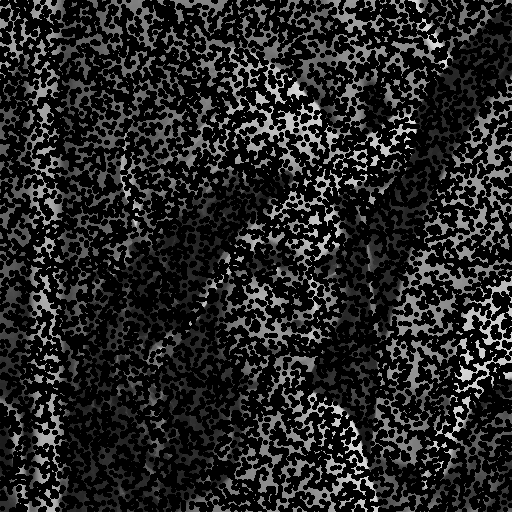
Salt and pepper noise image (threshold = 0.05) after 3x3 median filter.



Salt and pepper noise image (threshold = 0.05) after 5x5 median filter.



Salt and pepper noise image (threshold = 0.05) after closing followed by opening.



Salt and pepper noise image (threshold = 0.05) after opening followed by closing.



Salt and pepper noise image (threshold = 0.1) after 3x3 box filter.



Salt and pepper noise image (threshold = 0.1) after 5x5 box filter.



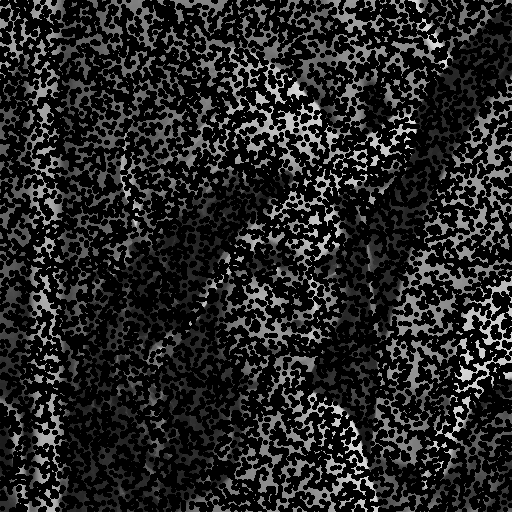
Salt and pepper noise image (threshold = 0.1) after 3x3 median filter.



Salt and pepper noise image (threshold = 0.1) after 5x5 median filter.



Salt and pepper noise image (threshold = 0.1) after closing followed by opening.



Salt and pepper noise image (threshold = 0.1) after opening followed by closing.

