

Homework 8

資工所碩一 R05922068 彭宇劭

Write the following programs

1. Generate additive white Gaussian noise
2. Generate salt-and-pepper noise
3. Run box filter (3x3, 5x5) on all noisy images
4. Run median filter (3x3, 5x5) on all noisy images
5. Run opening followed by closing or closing followed by opening

Source code: [hw8.py](#)

執行方式：[python hw8.py](#)

版本：[Python 2.7.10](#)

Output(bmp folder) :

Noise:

```
lena_gaussian10.bmp --> Gauss Noise with amplitude=10  
lena_gaussian30.bmp --> Gauss Noise with amplitude=30  
lena_salt_pepper_005.bmp --> salt-and-pepper noise (threshold=0.05)  
lena_salt_pepper_010.bmp --> salt-and-pepper noise (threshold=0.10)
```

Box Filter:

```
lena_box33_ga10.bmp --> 3*3 box filter on lena_gaussian10.bmp  
lena_box33_ga30.bmp --> 3*3 box filter on lena_gaussian30.bmp  
lena_box55_ga10.bmp --> 5*5 box filter on lena_gaussian10.bmp  
lena_box55_ga30.bmp --> 5*5 box filter on lena_gaussian30.bmp  
lena_box33_sp005.bmp --> 3*3 box filter on lena_salt_pepper_005.bmp  
lena_box33_sp010.bmp --> 3*3 box filter on lena_salt_pepper_010.bmp  
lena_box55_sp005.bmp --> 5*5 box filter on lena_salt_pepper_005.bmp  
lena_box55_sp010.bmp --> 5*5 box filter on lena_salt_pepper_010.bmp
```

Median Filter:

```
lena_med33_ga10.bmp --> 3*3 median filter on lena_gaussian10.bmp  
lena_med33_ga30.bmp --> 3*3 median filter on lena_gaussian30.bmp  
lena_med55_ga10.bmp --> 5*5 median filter on lena_gaussian10.bmp  
lena_med55_ga30.bmp --> 5*5 median filter on lena_gaussian30.bmp  
lena_med33_sp005.bmp --> 3*3 filter on lena_salt_pepper_005.bmp  
lena_med33_sp010.bmp --> 3*3 filter on lena_salt_pepper_010.bmp  
lena_med55_sp005.bmp --> 5*5 filter on lena_salt_pepper_005.bmp  
lena_med55_sp010.bmp --> 5*5 filter on lena_salt_pepper_010.bmp
```

Opening follow by Closing (closing-opening)

```
lena_cl_op_ga10.bmp --> closing-opening on lena_gaussian10.bmp  
lena_cl_op_ga30.bmp --> closing-opening on lena_gaussian30.bmp  
lena_cl_op_sp005.bmp --> closing-opening on lena_salt_pepper_005.bmp  
lena_cl_op_sp010.bmp --> closing-opening on lena_salt_pepper_010.bmp
```

Closing follow by Opening (opening-closing)

```
lena_op_cl_ga10.bmp --> opening-closing on lena_gaussian10.bmp  
lena_op_cl_ga30.bmp --> opening-closing on lena_gaussian30.bmp  
lena_op_cl_sp005.bmp --> opening-closing on lena_salt_pepper_005.bmp  
lena_op_cl_sp010.bmp --> opening-closing on lena_salt_pepper_010.bmp
```

簡述：

1. Define mask and Kernel

先在main function中定義mask大小權重以及kernel的形式 (3-5-5-5-3)

在mask中有三個元素，前兩個代表位置的偏移量，第三個為權重

```
# [i, j, w] : [row, col, weight]
box33 = [
    [-1, -1, 1], [0, -1, 1], [1, -1, 1],
    [-1, 0, 1], [0, 0, 1], [1, 0, 1],
    [-1, 1, 1], [0, 1, 1], [1, 1, 1]
]
box55 = [
    [-2, -2, 1], [-1, -2, 1], [0, -2, 1], [1, -2, 1], [2, -2, 1],
    [-2, -1, 1], [-1, -1, 1], [0, -1, 1], [1, -1, 1], [2, -1, 1],
    [-2, 0, 1], [-1, 0, 1], [0, 0, 1], [1, 0, 1], [2, 0, 1],
    [-2, 1, 1], [-1, 1, 1], [0, 1, 1], [1, 1, 1], [2, 1, 1],
    [-2, 2, 1], [-1, 2, 1], [0, 2, 1], [1, 2, 1], [2, 2, 1]
]
kernel =[ [-2, -1], [-2, 0], [-2, 1],
           [-1, -1], [-1, 0], [-1, 1], [-1, 2],
           [0, -1], [0, 0], [0, 1], [0, 2],
           [1, -1], [1, 0], [1, 1], [1, 2],
           [2, -1], [2, 0], [2, 1]]
```

2. Gaussian Noise

利用兩個for-loops跑過所有pixel，在每個pixel上加入 (Gaussian distribution的偏移值*定義的Amplitude)，當加入該偏移值後超過255 則將該點設為255

```
def gaussian_noise(img, amplitude):
    noise_img = np.zeros((img.shape[0], img.shape[1]), dtype=int)
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            new_pixel = img[i][j] + amplitude*np.random.normal()
            noise_img[i][j] = new_pixel if new_pixel < 255 else 255
    return noise_img
```

3. Salt-and-Pepper Noise

在這function中，會傳入一個參數-threshold，同樣利用兩個for-loops跑過所有pixel，在每個pixel上都會利用uniform distribution抽取一個 0-1之間的數字，當此數字小於傳入的threshold，則將此點設為0 (pepper)，若數字大於1-threshold則將此點設為255 (salt)

```
def salt_and_pepper(img, threshold):
    noise_img = np.zeros((img.shape[0], img.shape[1]), dtype=int)
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            uni = np.random.uniform()
            if uni < threshold:
                noise_img[i][j] = 0
            elif uni > (1-threshold):
                noise_img[i][j] = 255
            else:
                noise_img[i][j] = img[i][j]
    return noise_img
```

4. Box Filter

在box_filter function中，跑過所有pixel，在每個pixel上，利用傳入的3*3 或是5*5的box罩上去，若該點在圖片範圍內，則將該點的pixel值乘上權重累加(pixel_val)，最後再將累加的值除以權重數(normalize)，所得出來的數值為該點的新值，跑完所有pixel輸出圖片

```
def box_filter(img, box):
    filtered_img = img.copy()
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            pixel_val = 0
            normalize = 0
            for [m, n, l] in box:
                if (i+m>=0 and i+m<img.shape[0] and j+n>=0 and j+n<img.shape[1]):
                    normalize += l
                    pixel_val += filtered_img[i+m][j+n]*l
            filtered_img[i][j] = pixel_val/normalize
    return filtered_img
```

5. Median Filter

前面同上述方法，跑過所有pixel，在每個pixel上，利用傳入的3*3 或是5*5的box罩上去，若該點在圖片範圍內，則將該點的pixel值乘上權重，但在這邊不做累加的動作，而使把該點用一個暫時的list存起來，跑完一組mask之後，從該暫存的list中取出中間值並設為該點的新值，同樣，跑完所有pixel後輸出圖片

```
def median_filter(img, box):
    filtered_img = img.copy()
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            counter = 0
            tmp_list = []
            for [m, n, l] in box:
                if (i+m>=0 and i+m<img.shape[0] and j+n>=0 and j+n<img.shape[1]):
                    counter += l
                    tmp_list.append(img[i+m][j+n])
            tmp_list.sort()
            if (counter % 2 == 1):
                filtered_img[i][j] = tmp_list[counter/2]
            else:
                filtered_img[i][j] = (tmp_list[counter/2-1]+tmp_list[counter/2])/2
    return filtered_img
```

6. Opening follow by Closing & Closing follow by Opening

同Hw5，將圖片先opening再closing或先closing再opening

```
def opening(img, kernel):
    return dilation(erosion(img, kernel), kernel)

def closing(img, kernel):
    return erosion(dilation(img, kernel), kernel)
```

結果：

Noise



[lena_gaussian10.bmp](#)



[lena_gaussian30.bmp](#)



[lena_salt_pepper_005.bmp](#)



[lena_salt_pepper_010.bmp](#)

Box Filter



左上 lena_box33_ga10.bmp
右上 lena_box33_ga30.bmp
左下 lena_box33_sp005.bmp
右下 lena_box33_sp010.bmp



左上 lena_box55_ga10.bmp
右上 lena_box55_ga30.bmp
左下 lena_box55_sp005.bmp
右下 lena_box55_sp010.bmp

Median Filter



左上 lena_med33_ga10.bmp
右上 lena_med33_ga30.bmp
左下 lena_med33_sp005.bmp
右下 lena_med33_sp010.bmp



左上 lena_med55_ga10.bmp
右上 lena_med55_ga30.bmp
左下 lena_med55_sp005.bmp
右下 lena_med55_sp010.bmp

Opening and Colsing



Opening follow by Closing

左上 lena_cl_op_ga10.bmp

右上 lena_cl_op_ga30.bmp

左下 lena_cl_op_sp005.bmp

右下 lena_cl_op_sp010.bmp



Closing follow by Opening

左上 lena_op_cl_ga10.bmp

右上 lena_op_cl_ga30.bmp

左下 lena_op_cl_sp005.bmp

右下 lena_op_cl_sp010.bmp