**Computer Vision HW8**

R08922a27 資工系 人工智慧碩士班 李吉昌

I use python 3.7 to implement all image processing requirements. Reading .bmp file by **PIL**, and then processing through **NumPy** array.

* **1. Results**

**Gaussian noise with amplitude of 10**

****

**Gaussian noise with amplitude of 30**

****



**Salt-and-Pepper noise with probability 0.1**





**Salt-and-Pepper noise with probability 0.05**

****

* **2. Code fragment**

def gaussian\_noise\_transform(img, amp):

return img + amp \* np.random.normal(0, 1, img.shape)

def salt\_and\_pepper\_noise\_transform(img, thr):

prob\_map = np.random.uniform(0, 1, img.shape)

salt\_idx = prob\_map > 1 - thr

pepper\_idx = prob\_map < thr

img\_sp = img.copy()

img\_sp[salt\_idx] = 255

img\_sp[pepper\_idx] = 0

return img\_sp

def padding(img, filter\_size):

img\_pad = np.zeros((img.shape[0] + filter\_size // 2 \* 2, img.shape[1] + filter\_size // 2 \* 2), np.int)

for i in np.arange(filter\_size // 2, img.shape[0] + filter\_size // 2):

for j in np.arange(filter\_size // 2, img.shape[1] + filter\_size // 2):

img\_pad[i, j] = img[i - filter\_size // 2, j - filter\_size // 2]

return img\_pad

def box\_filter(img, filter\_size):

img\_mean = np.zeros(img.shape)

img\_pad = padding(img, filter\_size)

for i in range(img\_mean.shape[0]):

for j in range(img\_mean.shape[1]):

img\_mean[i, j] = img\_pad[i: i + filter\_size, j: j + filter\_size].mean()

return img\_mean

def median\_filter(img, filter\_size):

img\_med = np.zeros(img.shape)

img\_pad = padding(img, filter\_size)

for i in range(img\_med.shape[0]):

for j in range(img\_med.shape[1]):

img\_med[i, j] = np.median(img\_pad[i: i + filter\_size, j: j + filter\_size])

return img\_med

def snr(img\_org, img\_pro):

noise = img\_pro - img\_org

return np.log10(img\_org.var()/noise.var()) \* 10

* **3. Brief Description**

All the assigned operators' implementation details follow the course's lecture slides. The **np.random** was used to simulate all the assigned noise distribution.The filters were implemented by using looping to determine each processed pixel's value. Please note that the boundary pixel's values of the opening and closing operation's outputs are different, so that the SNR values would be a little different from the reference.