

Computer Vision HW8

資工碩一 胡嘉祐 R07922162

1 Principle code

Part 1. Generate additive white Gaussian & noise salt-and-pepper noise

```
def additive_white_Gaussian_noise(img , amplitude):
    use_img = new_img(img)
    randnum = []
    length = len(use_img)

    for i in range (length):
        for j in range (length):

            rand = np.random.normal(0, amplitude)
            rand = int(rand)
            randnum.append(rand)
            temp = rand+img[i][j][0]
            if(temp>255):
                temp=255
            elif(temp<0):
                temp=0
            use_img[i][j]= (temp,temp,temp)

    return use_img
```

```
def salt_and_pepper (img,threshold):
    use_img = new_img(img)
    x = 1/threshold
    x = int (x)
    length = len(use_img)
    for i in range (length):
        for j in range (length):
            rand = random.randint(1,x)
            if(rand==1):
                use_img[i][j]=(0,0,0)
            elif (rand==x):
                use_img[i][j]=(255,255,255)
            else:
                use_img[i][j]=img[i][j]

    return use_img
```

Part 2. Box filter and Median filter

```
def box_filter (img,kernel):
    k=0
    if(kernel == 3):
        k=1
    else:
        k=2

    expand_img = expand(img,k)
    new_i = np.zeros((len(img),len(img[0]),3), np.uint8)

    for i in range (k,len(img)+k):
        for j in range (k,len(img)+k):
            mean = 0
            for ii in range (i-k,i+k+1):
                for jj in range(j-k,j+k+1):
                    mean+=expand_img[ii][jj][0]
            mean = mean/(kernel*kernel)
            if(mean-int(mean)>=0.5):
                mean = int(mean)+1
            else:
                mean = int(mean)
            new_i[i-k][j-k]=(mean,mean,mean)

    return new_i
```

```
def median_filter (img,kernel):
    k=0
    m=0
    if(kernel == 3):
        k=1
        m=5
    else:
        k=2
        m=13

    expand_img = expand(img,k)
    new_i = np.zeros((len(img),len(img[0]),3), np.uint8)

    for i in range (k,len(img)+k):
        for j in range (k,len(img)+k):
            median = []
            for ii in range (i-k,i+k+1):
                for jj in range(j-k,j+k+1):
                    median.append(expand_img[ii][jj][0])
            new_i[i-k][j-k]=(median[m],median[m],median[m])

    return new_i
```

2. Result

part 1 four Noise Image

additive white Gaussian noise

amplitude :10



amplitude :30



salt-and-pepper noise

threshold :0.05



threshold :0.1



Part 2 additive white Gaussian noise amplitude : 10

Box filter 3x3



Box filter 5x5



median filter 3x3



median filter 5x5



Opening then closing



Closing then opening



Part 3 additive white Gaussian noise amplitude : 30

Box filter 3x3



Box filter 5x5



median filter 3x3



median filter 5x5



Opening then closing



Closing then opening



Part 4 salt-and-pepper noise threshold 0.05

Box filter 3x3



Box filter 5x5



median filter 3x3



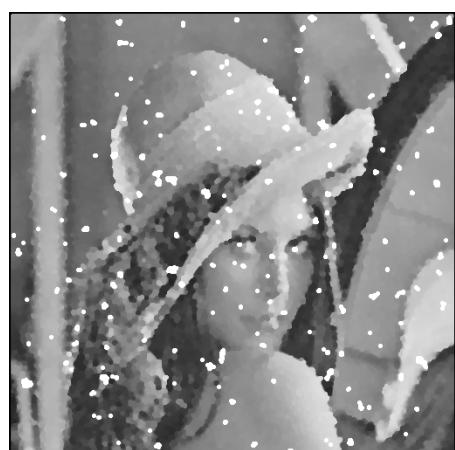
median filter 5x5



Opening then closing



Closing then opening



Part 5 salt-and-pepper noise threshold:0.1

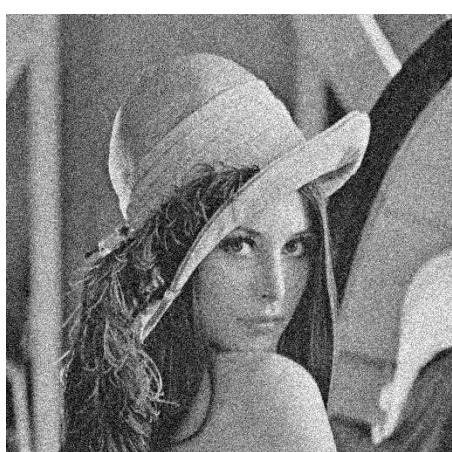
Box filter 3x3



Box filter 5x5



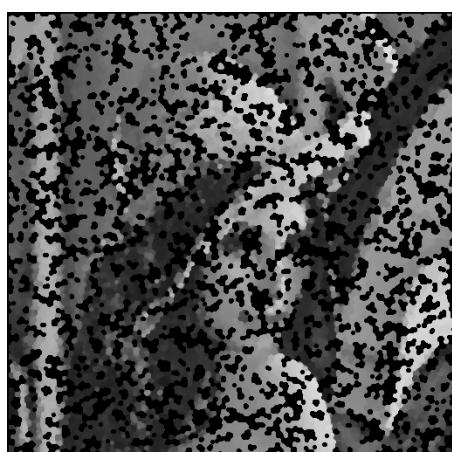
median filter 3x3



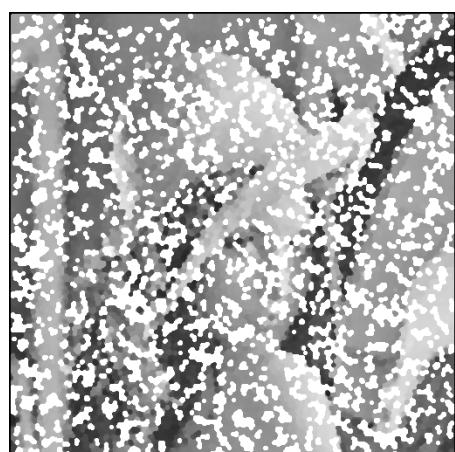
median filter 5x5



Opening then closing



Closing then opening



3 SVR

	Box 3x3	Box 5x5	Median 3x3	Median5x5	Open then close	Close then open
Gaussian :10	17.3	14.8	17.1	13.4	8.82	7.62
Gaussian :30	12.4	13.7	11.9	9.3	8.64	6.5
Salt&pepper:0.05	8.9	11.4	18.4	13.8	5.3	4.51
Salt&pepper:0.1	6.3	8.6	14.3	11.8	-2.7	-4.6