Computer Vision
HW8

R10921104 王怡堯

November 29, 2021

1 Primary Procedures

For Opening and Closing , I'll use the code in HW5.

1.1 Gaussian Noise

```
def gaussian_noise(lena,amplitude):
    m=len(lena)
    n=len(lena[0])
    res=[[0]*n for i in range(m)]
    for i in range(m):
        for j in range(n):
            res[i][j]=max(0,min(255,lena[i][j]+amplitude*random.gauss(0,1)))
    return res
```

1.2 Salt and Pepper Noise

```
def s_and_p_noise(lena,theshold):
    m=len(lena)
    n=len(lena[0])
    res=[[0]*n for i in range(m)]
    for i in range(m):
        for j in range(n):
            tmp=random.random()
            if tmp<theshold:
                res[i][j]=0
            elif tmp>1-theshold:
                res[i][j]=255
            else:
                res[i][j]=lena[i][j]
    return res
```

1.3 Box Filter (3*3)

```
def box_filter3(lena):
    m=len(lena)
    n=len(lena[0])
    lena=expan(lena)
    res=[[0]*n for i in range(m)]
    for i in range(m):
        tmp=lena[i][0]+lena[i+1][0]+lena[i+2][0]
        tmp+=lena[i][1]+lena[i+1][1]+lena[i+2][1]
        tmp+=lena[i][2]+lena[i+1][2]+lena[i+2][2]
        res[i][0]=tmp//9
        for j in range(1,n):
            tmp-=lena[i][j-1]+lena[i+1][j-1]+lena[i+2][j-1]
            tmp+=lena[i][j+2]+lena[i+1][j+2]+lena[i+2][j+2]
            res[i][j]=tmp//9
    return res
```

1.4 Box Filter (5*5)

```
def box_filter5(lena):
    m=len(lena)
    n=len(lena[0])
    lena=expan(lena)
    lena=expan(lena)
    res=[[0]*n for i in range(m)]
    for i in range(m):
        tmp=lena[i][0]+lena[i+1][0]+lena[i+2][0]+lena[i+3][0]+lena[i+4][0]
        tmp+=lena[i][1]+lena[i+1][1]+lena[i+2][1]+lena[i+3][1]+lena[i+4][1]
        tmp+=lena[i][2]+lena[i+1][2]+lena[i+2][2]+lena[i+3][2]+lena[i+4][2]
        tmp+=lena[i][3]+lena[i+1][3]+lena[i+2][3]+lena[i+3][3]+lena[i+4][3]
        tmp+=lena[i][4]+lena[i+1][4]+lena[i+2][4]+lena[i+3][4]+lena[i+4][4]
        res[i][0]=tmp//25
        for j in range(1,n):
            tmp-=lena[i][j-1]+lena[i+1][j-1]+lena[i+2][j-1]+lena[i+3][j-1]+lena[i+4][j-1]
            tmp+=lena[i][j+4]+lena[i+1][j+4]+lena[i+2][j+4]+lena[i+3][j+4]+lena[i+4][j+4]
            res[i][j]=tmp//25
    return res
```

1.5 Median Filter (3*3)

```
def med_filter3(lena):
    m=len(lena)
    n=len(lena[0])
    lena=expan(lena)
    res=[[0]*n for i in range(m)]
    for i in range(m):
        for j in range(n):
            res[i][j]=sorted(lena[i][j:j+3]+lena[i+1][j:j+3]+lena[i+2][j:j+3])[4]
    return res
```

1.6 Median Filter (5*5)

```
def med_filter5(lena):
    m=len(lena)
    n=len(lena[0])
    lena=expan(lena)
    lena=expan(lena)
    res=[[0]*n for i in range(m)]
    for i in range(m):
        for j in range(n):
            tmp=sorted(lena[i][j:j+5]+lena[i+1][j:j+5]+lena[i+2][j:j+5]+lena[i+3][j:j+5]+lena[i+4][j:j+5])
            res[i][j]=tmp[12]
    return res
```

1.7 Computing Signal Noise Rate (SNR)

```
def SNR(origin,lena):
    m=len(origin)
    n=len(origin[0])
    mu=0
    mu_n=0
    for i in range(m):
        for j in range(n):
            mu+=origin[i][j]
            mu_n+=lena[i][j]-origin[i][j]
    mu/=m*n
    mu_n/=m*n
    vs=0
    vn=0
    for i in range(m):
        for j in range(n):
            vs+=(origin[i][j]-mu)**2
            vn+=(lena[i][j]-origin[i][j]-mu_n)**2
    vs/=m*n
    vn/=m*n
    return 20*np.log10((vs**0.5)/(vn**0.5))
```

2 Result

Gaussian Noise (amplitude=10)
SNR=13.6130506044

Box Filter 3*3
SNR=15.4130851200

Box Filter 5*5
SNR=14.870670493

Median Filter 3*3
SNR=17.6862701444

Median Filter 5*5
SNR=15.9874514287

Opening-then-Closing
SNR=6.4587138667

Closing-then-Opening
SNR=5.29204632897

Gaussian Noise (amplitude=30)
SNR=4.1657734748

Box Filter 3*3
SNR=12.5972301851

Box Filter 5*5
SNR=13.3031734220

Median Filter 3*3
SNR=11.0682457441

Median Filter 5*5
SNR=12.8832447983

Opening-then-Closing
SNR=7.0260368184

Closing-then-Opening
SNR=3.85076320620

Salt and Pepper Noise (threshold=0.05)
SNR=0.9043074295

Box Filter 3*3
SNR=9.4436221358

Box Filter 5*5
SNR=11.120614263

Median Filter 3*3
SNR=19.311863677

Median Filter 5*5
SNR=16.393028419

Opening-then-Closing
SNR=3.64708160609

Closing-then-Opening
SNR=2.75776652710

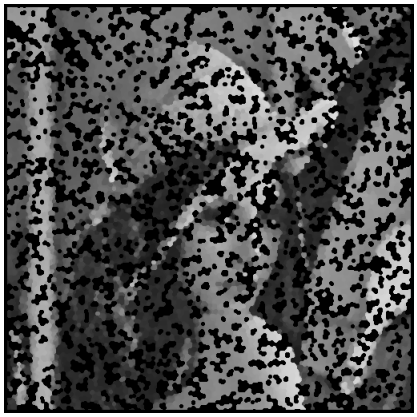
Salt and Pepper Noise (threshold=0.1)
SNR=-2.0969583441

Box Filter 3*3
SNR=6.352908846

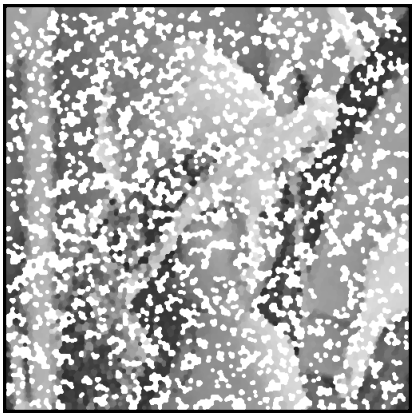
Box Filter 5*5
SNR=8.5315483297

Median Filter 3*3
SNR=15.0759910952

Median Filter 5*5
SNR=15.745725733



Opening-then-Closing
SNR=-2.3192932549702623



Closing-then-Opening
SNR=-3.2127249492740066