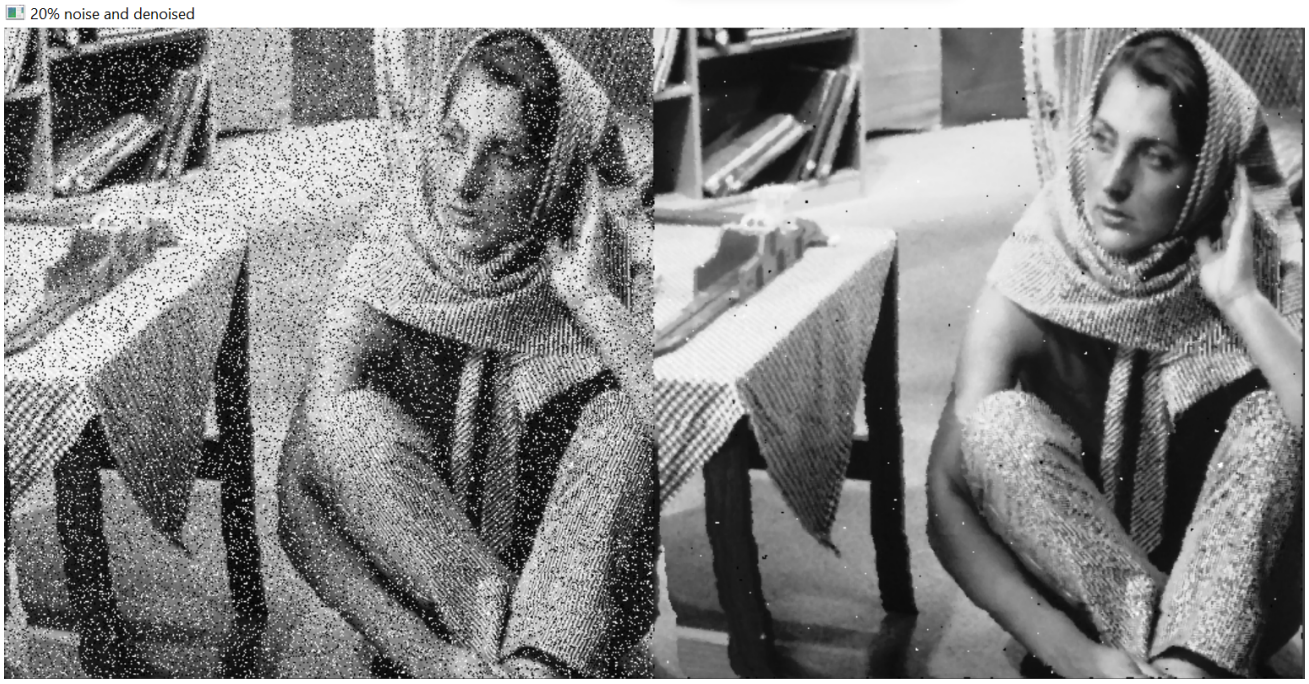
Name: Sahas Marwah
Roll Number: 2020237
DIP Assignment 2

Q1:

NOTE: Median Filtering is done on 5x5 and 3x3 kernels. But here, images of 5x5 are shown as they had the better-denoised output.
This does not mean they had the better PSNR as it was relatively lower than the PSNR when a 3x3 median filter was used.

For example in the 3x3 case, we got the denoised image with some noise:



Hence, the **parameters for denoising** used here are 5x5.

Comparison of PSNRs:
5x5 Median Filter:

```
PSNR 5%: 31.61333516687002
PSNR 15%: 31.487283977956075
PSNR 20%: 31.410493401375508
PSNR 25%: 31.36738812280634
```

3x3 Median Filter:

```
PSNR 5%: 32.83039207191129
PSNR 15%: 32.514098635226944
PSNR 20%: 32.387945657811564
PSNR 25%: 32.25886054673095
```

A. 5% noise:

PSNR = ~31.61192871731332

B. 15% noise:

PSNR = ~31.481557191731255

C. 20% noise:

PSNR = ~31.43082985039085

D. 25% noise:

PSNR = ~31.35627773376776

Observation:
As the noise increases, PSNR decreases.

Q2:

The given image was to be reduced by 16 times.
Hence, the length was divided by 4, and the breadth was also divided by 4.

Input Image:

Resized Image (128x128):

A. Nearest Neighbour:

Nearest Neighbour

PSNR: ~33.6055806020707

B. Bilinear:

BiLinear

PSNR: ~33.53990280460686

Similarly, 3 more kernels have been made:

```python
#ORDER 2 Kernels
def bell_kernel(x):
    if abs(x) < 1:
        ans = 0.75 - x**2
    elif 0.5 < abs(x) < 1.5:
        ans = 0.5 * (abs(x) - 1.5)**2
    else:
        ans = 0
    return ans

#ORDER 3 Kernels
def hermite_kernel(x):
    if abs(x) <= 1:
        ans = 2 * abs(x)**3 - 3 * abs(x)**2 + 1
    else:
        ans = 0
    return ans

def bicubic_kernel(x, a):
    if abs(x) <= 1:
        ans = (a+2) * abs(x)**3 - (a+3) * abs(x)**2 + 1
    elif 1 < abs(x) < 2:
        ans = a * abs(x)**3 - 5*a * abs(x)**2 + 8*a * abs(x) - 4*a
    else:
        ans = 0
    return ans
```