R09945020 朱彥如

Homework 8

Write a program which does:

(a) Generate noisy images with gaussian noise(amplitude of 10 and 30)
(b) Generate noisy images with salt-and-pepper noise(probability 0.1 and 0.05)
(c) Use the 3x3, 5x5 box filter on images generated by (a)(b)
(d) Use 3x3, 5x5 median filter on images generated by (a)(b)
(e) Use both opening-then-closing and closing-then opening filter (using the octogonal 3-5-5-5-3 kernel, value = 0) on images generated by (a)(b)

You must calculate the signal-to-ratio (SNR) for each instance (4 noisy images and 24 processed images).

The requirements above are accomplished by Python with the help of matplotlib package, cv2 package, numpy package and math package.

(a) Generate noisy images with gaussian noise(amplitude of 10 and 30)

To create the image with gaussian noise, I made the function, gua_noise() with inputs, image array and amplitude. The core calculation of the function is shown below.

$$gua\_noise(image, amp) = \begin{cases} noise = image[r,c] + amp \times random\ .guass(0,1) \\ return\ 255,\ if\ noise > 255 \\ return\ noise,\ otherwise \end{cases}$$

(b) Generate noisy images with salt-and-pepper noise(probability 0.1 and 0.05)

To create the image with salt-and-pepper noise, I made the function, salt_pepper() with inputs, image array and threshold. The core calculation of the function is shown below.

$$salt\_pepper(image, threshold) = \begin{cases} random\ value = random\ .uniform(0,1) \\ return\ 0,\ if\ random\ value \leq threshold \\ return\ 255,\ if\ random\ value \geq threshold \end{cases}$$

(c) Use the 3x3, 5x5 box filter on images generated by (a)(b)

I wrote the function, median_filter(), to mask the median filter on the image.
In the function, for each pixel, it finds the median of pixels around which can be applied within the using kernel. Also, to tackle the out-of-range problems in boundary pixels where kernel applied, I use np.pad() with the width (kernel side length-1)/2 before detecting every pixel. At last, the function uses np.stack() to create a 3D (RGB) filtered image.

(d) Use 3x3, 5x5 median filter on images generated by (a)(b)

I wrote the function, box_filiter(), to mask the box filter on the image.
In the function, for each pixel, it calculates the average of pixels around which can be applied within the using kernel. Also, to tackle the out-of-range problems in boundary pixels where kernel applied, I use np.pad() with the width (kernel side length-1)/2 before

detecting every pixel. At last, the function uses np.stack() to create a 3D (RGB) filtered image.

(e) Use both opening-then-closing and closing-then opening filter

To create opening-then-closing and closing-then opening filter, I use dilation() function and erosion() function from homework 5.

Opening:

$(B \ominus K) \oplus K$

Closing:

$(B \oplus K) \ominus K$

Then I use the combination of opening and closing to create the filter I need.

S/N ratio (signal to noise ratio):

To calculate the S/N ration for every image, I wrote the functions with formula below.

$$SNR = 20 \times log_{10} \frac{\sqrt{VS}}{\sqrt{VN}}$$

$$VS = \frac{\sum_{\forall n} (I(i,j) - \mu_s)^2}{\|n\|}$$

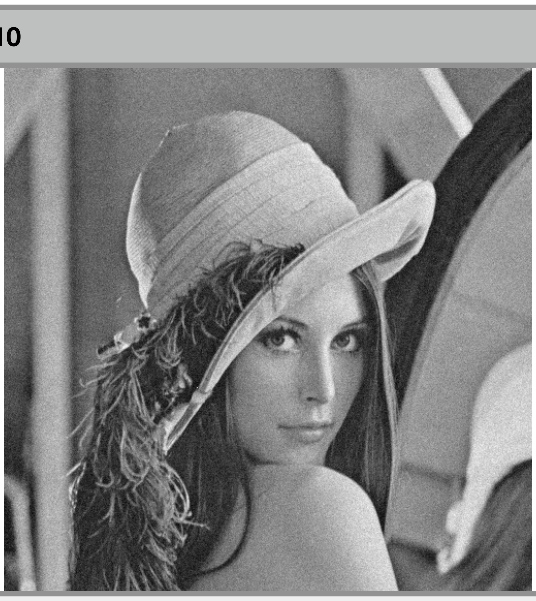
$$VN = \frac{\sum_{\forall n} (I_{noises}(i,j) - I(i,j) - \mu_{noises})^2}{\|n\|}$$

Also, to calculate the log results, I import math package.

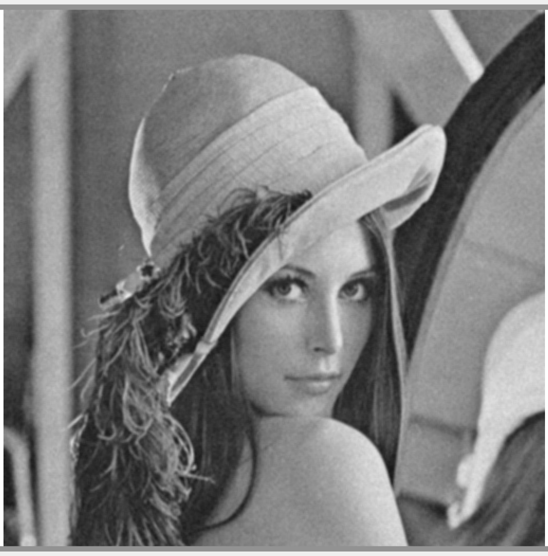
All the S/N ratios of processed images are recorded in the table with the corresponding images.

R09945020 朱彥如

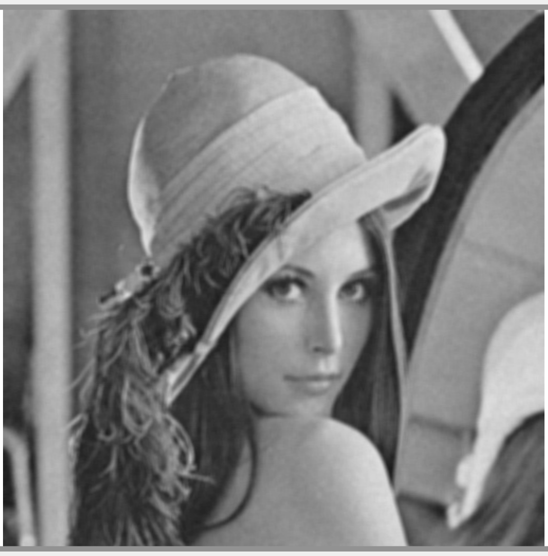
Gaussian noise, amplitude: 10



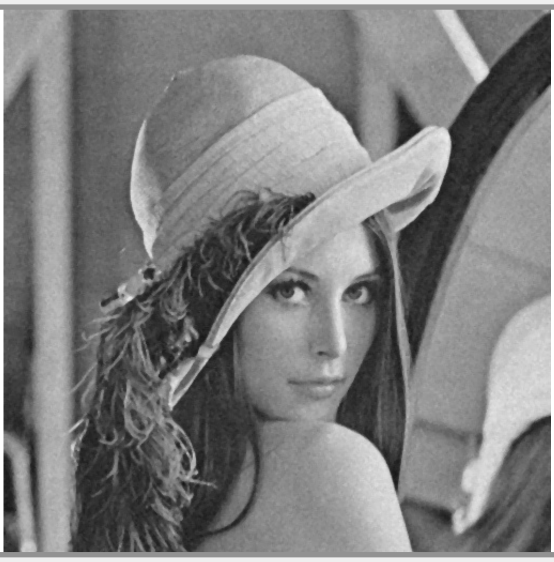
gaussian noise, amplitude: 10, SNR = 13.59



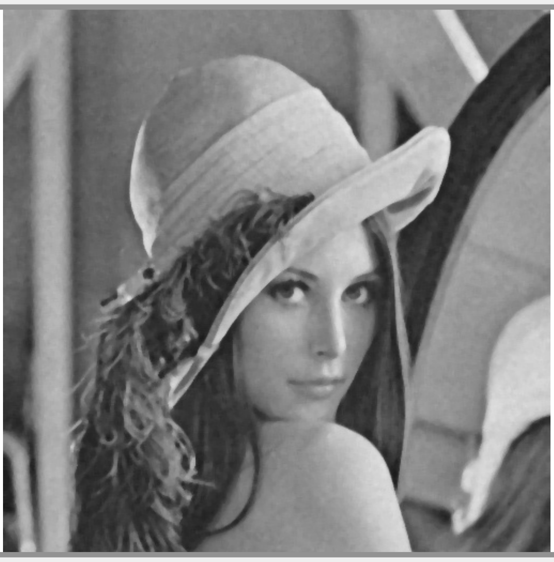
Box 3x3, SNR = 17.72



Box 5x5, SNR = 14.85

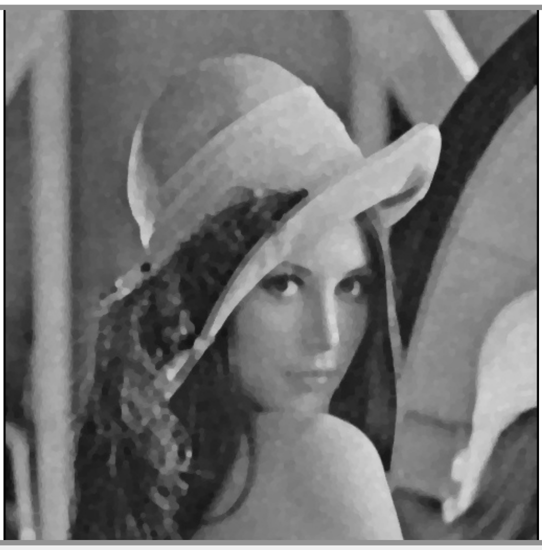


Median 3x3, SNR = 17.65

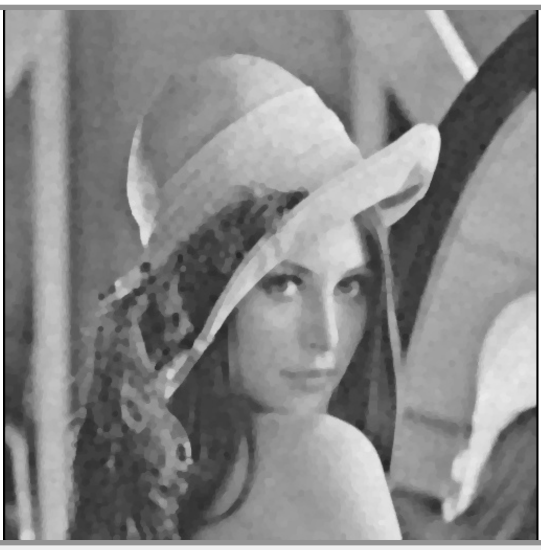


Median 5x5, SNR = 15.98

R09945020 朱彦如

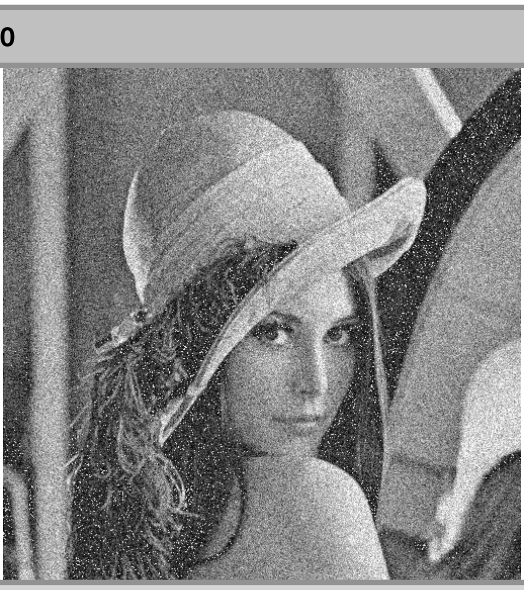


Opening then closing, SNR = 8.61

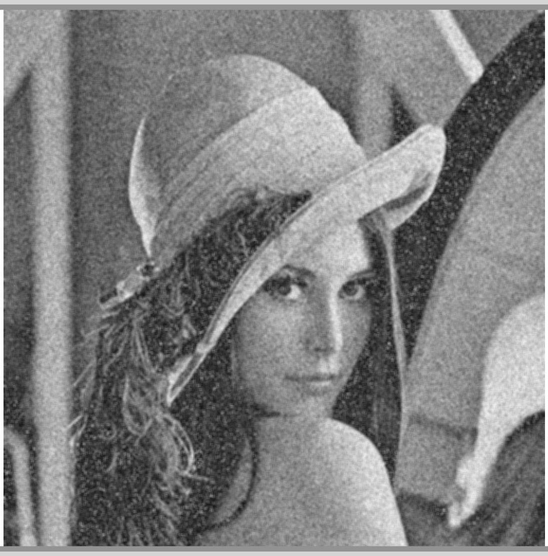


Closing then opening, SNR = 7.64

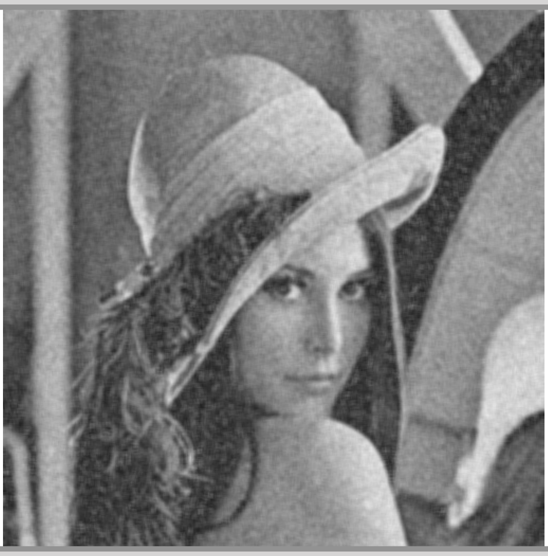
Gaussian noise, amplitude: 30



gaussian noise, amplitude: 10, SNR = 2.77

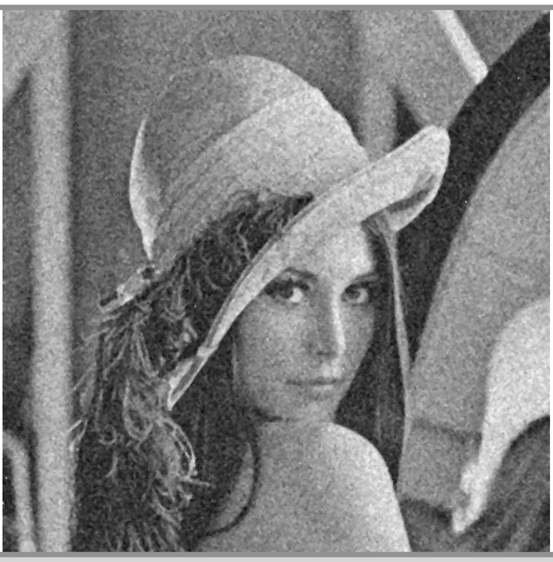
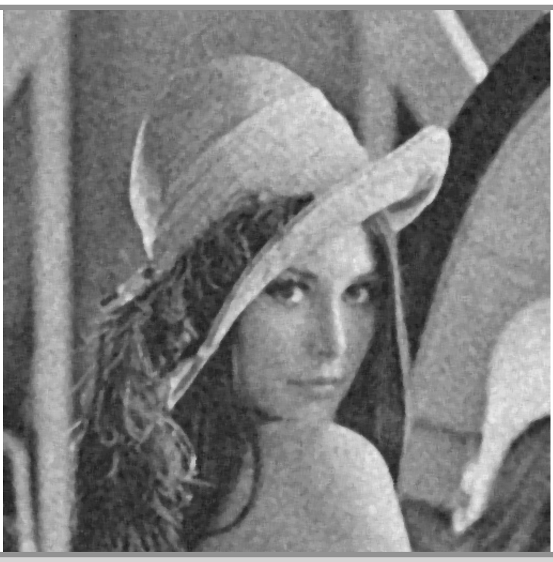
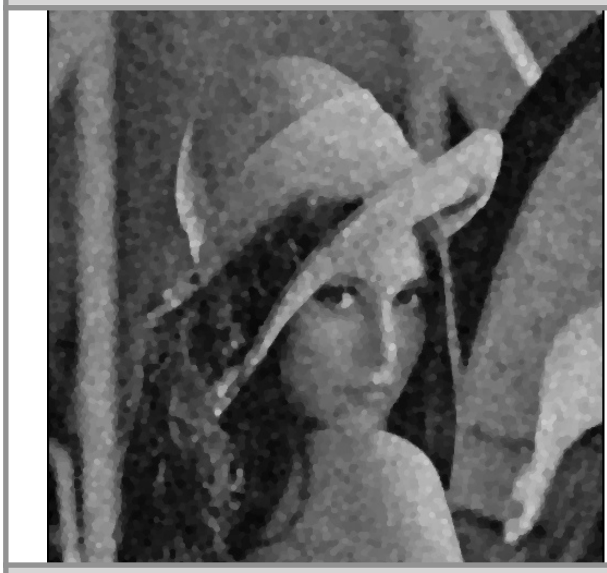
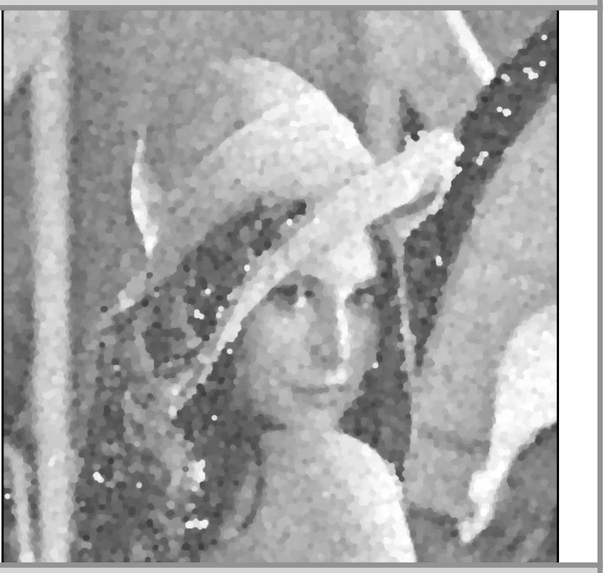


Box 3x3, SNR = 10.72

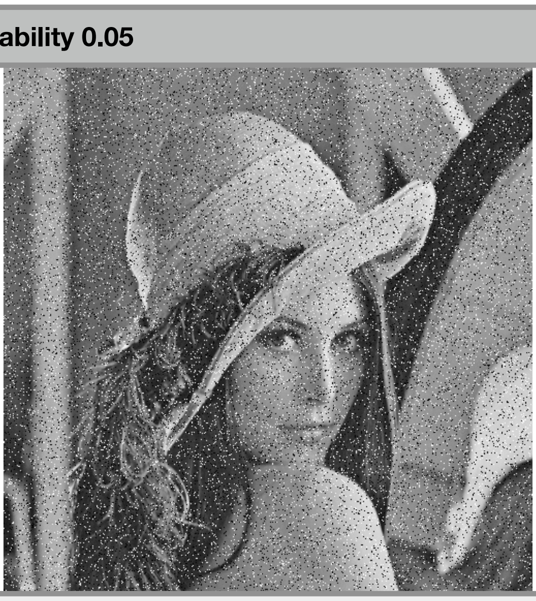


Box 5x5, SNR = 11.69

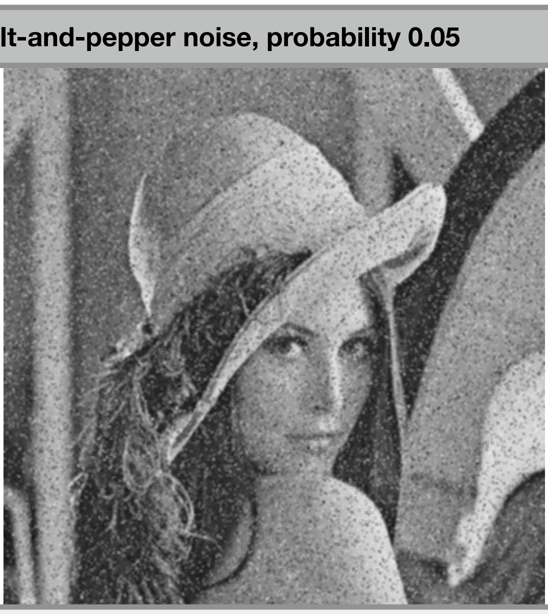
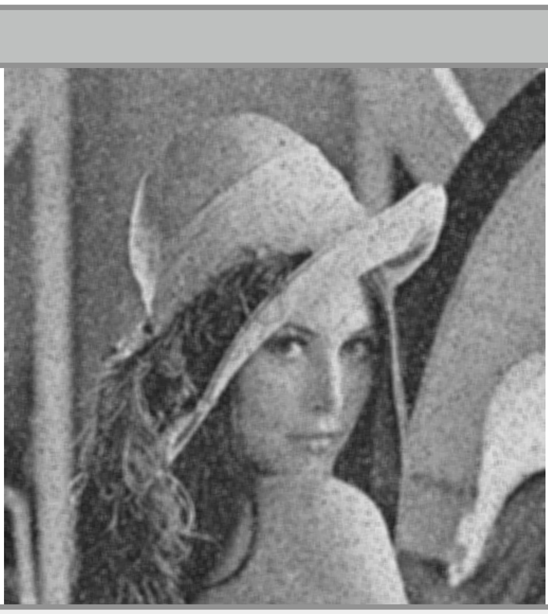
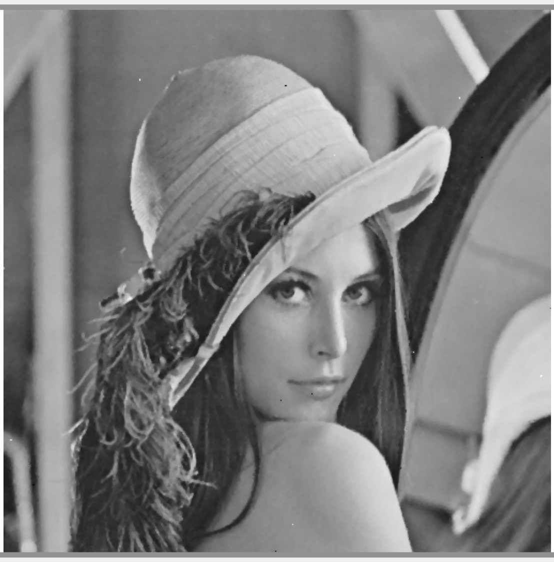
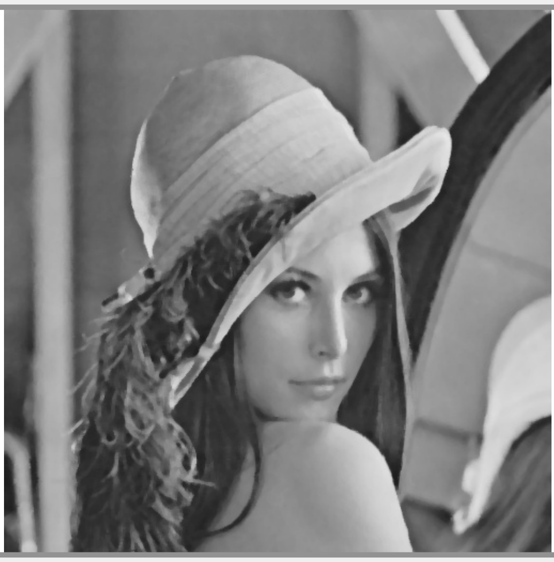
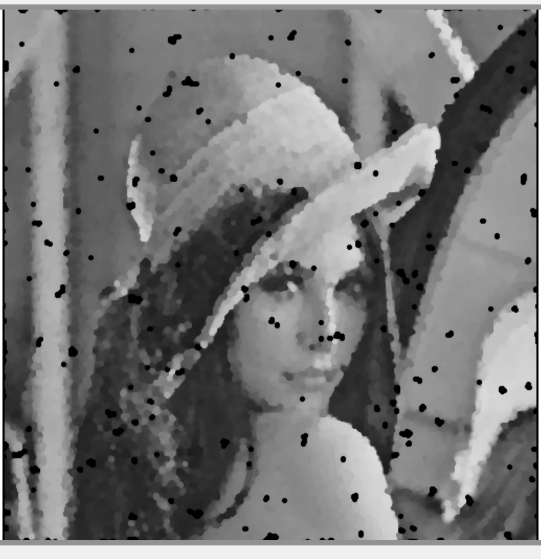
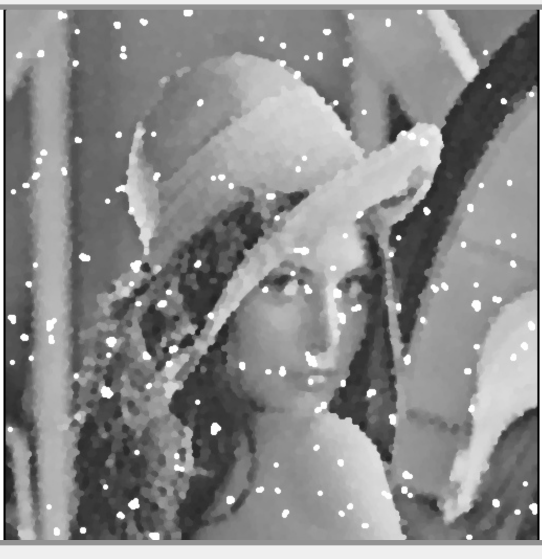
R09945020 朱彦如

	
Median 3x3, SNR = 10.87	Median 5x5, SNR = 12.62
	
Opening then closing, SNR = 8.52	Closing then opening, SNR = 4.57

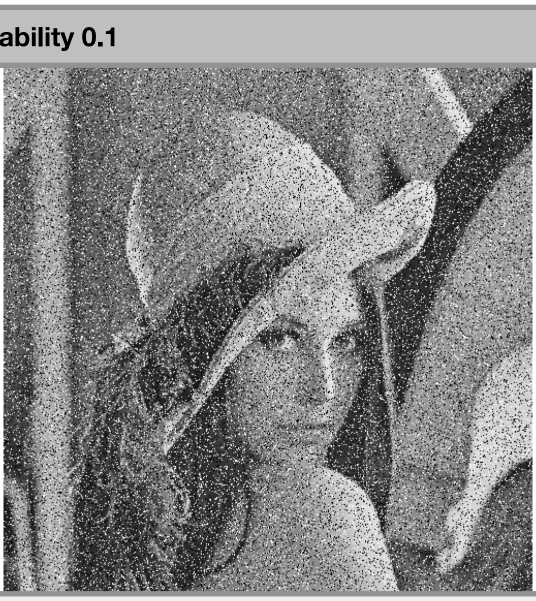
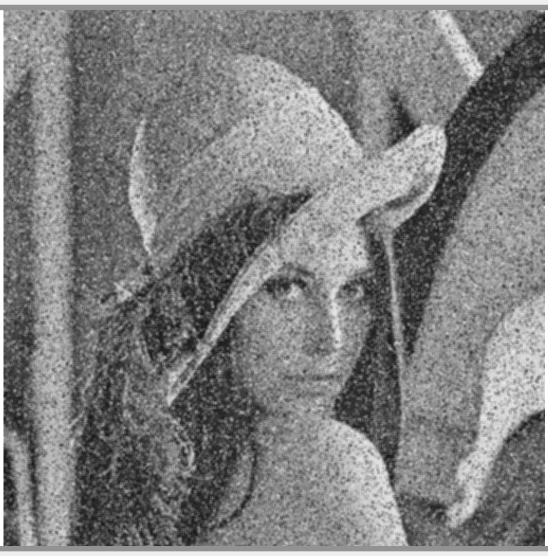
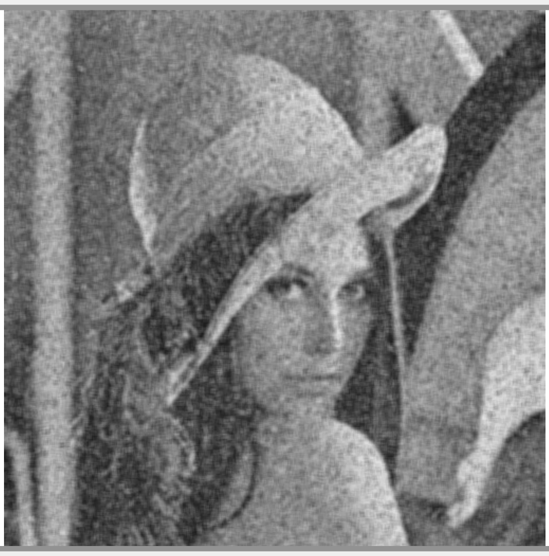
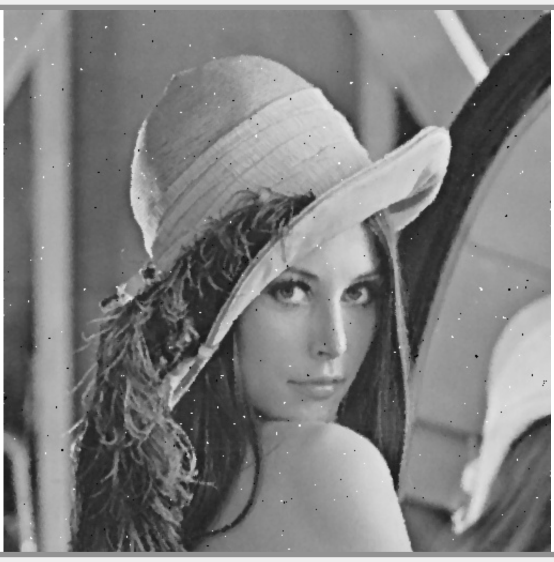
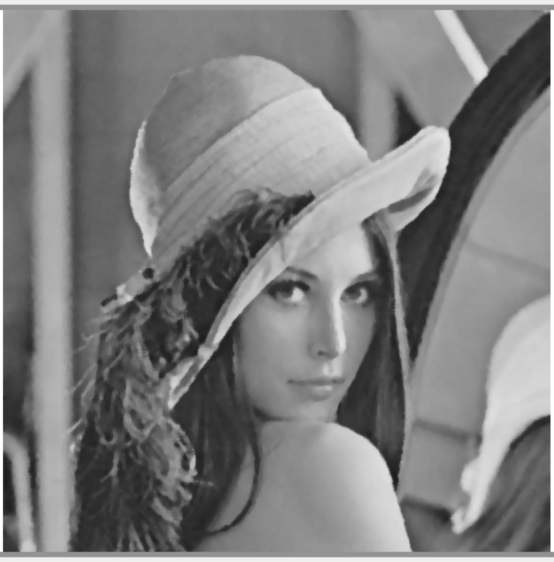
Salt-and-pepper noise, probability 0.05

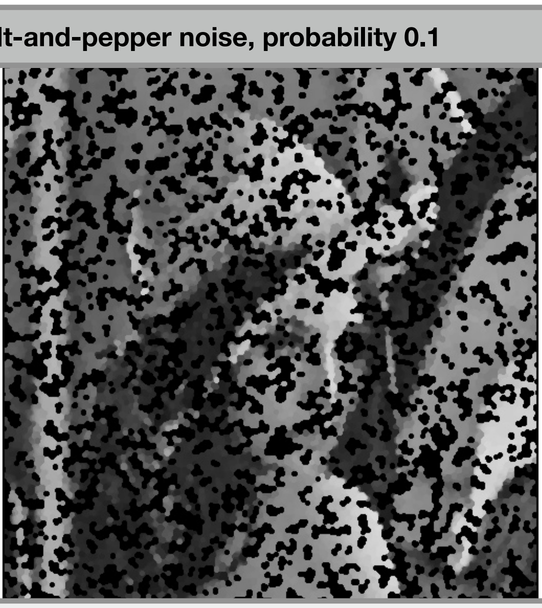
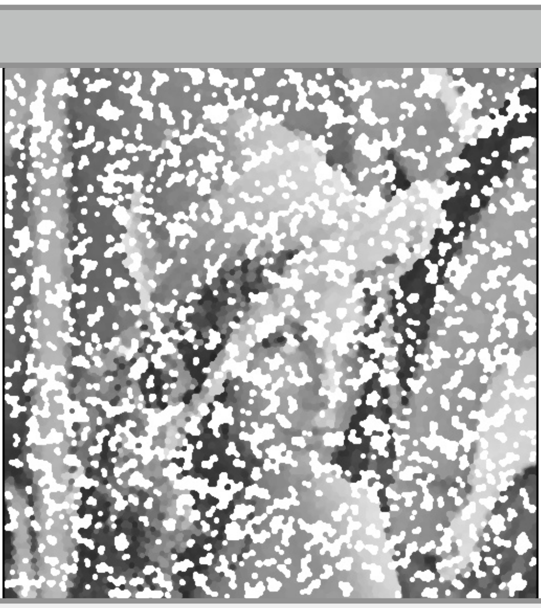


Salt-and-pepper noise, probability 0.05, SNR = 0.97

<table>
<tr><th colspan="2">Salt-and-pepper noise, probability 0.05</th></tr>
<tr><td></td><td></td></tr>
<tr><td>Box 3x3, SNR = 9.49</td><td>Box 5x5, SNR = 11.2</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Median 3x3, SNR = 19.26</td><td>Median 5x5, SNR = 16.4</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Opening then closing, SNR = 4.66</td><td>Closing then opening, SNR = 4.07</td></tr>
</table>

R09945020 朱彥如

<table>
<tr><th colspan="2">Salt-and-pepper noise, probability 0.1</th></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">Salt-and-pepper noise, probability 0.1, SNR = -2.1</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Box 3x3, SNR = 6.35</td><td>Box 5x5, SNR = 8.58</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Median 3x3, SNR = 14.88</td><td>Median 5x5, SNR = 15.76</td></tr>
</table>

Salt-and-pepper noise, probability 0.1	
	
Opening then closing, SNR = -2.22	Closing then opening, SNR = -2.86