Name: **Iman Mousaei**
Course: **Image Processing**
Assignment **2**

Student Number: **98222099**
Date: May 4, 2023

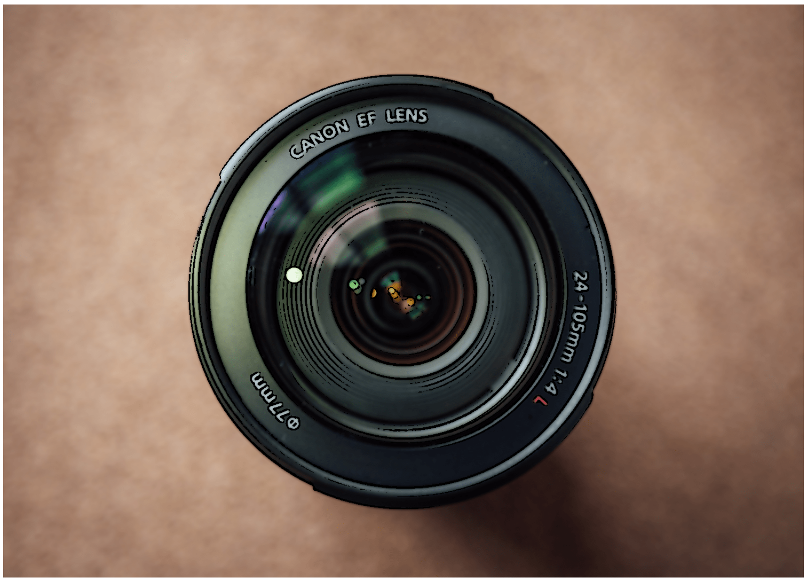
The full code with outputs can be found here: https://github.com/imanmousaei/Image-Processing-HW/tree/main/HW2

1 Cartoonize

We used multiple methods to Cartoonize the input image, which we'll explain next.

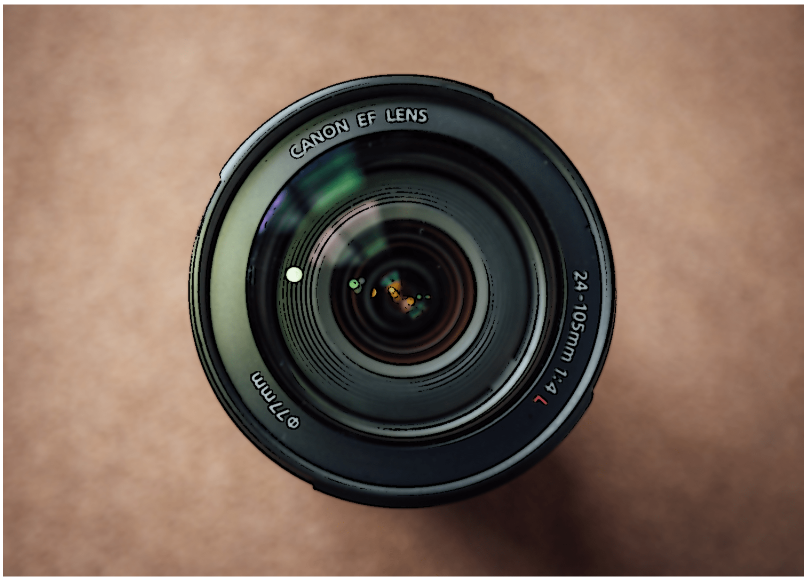
1.1 Method 1

Our first method is to first use canny edge detection to detect edges. This method colors edges as white so we reverse the pixels so they would be black. Then we multiply this image with its weight and add it to the GaussianBlur of the input image with $kernel_size = 3$. You can see the output below:



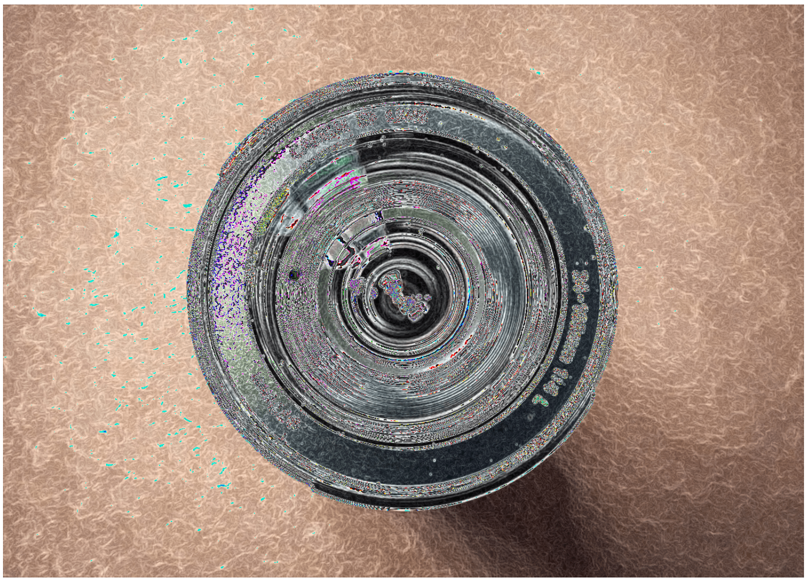
1.2 Method 2

In this method, we first use threshold edge detection. Then we again multiply it by its weight and bitwise_and it with the smoothed image of the input(using bilateralFilter). The result is this:



1.3 Method 3

Here we use sobel edge detection and multiply it by its weight and add it to the smoothed image of the input(using bilateralFilter). In the image below you can see the output:



2 Exercise 2

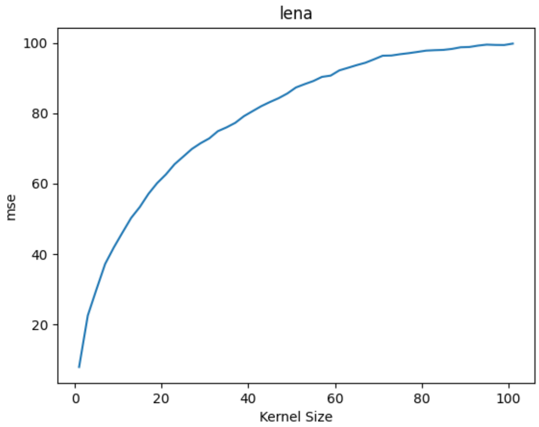
In this exercise, we add Gaussian Noise with a mean of 0 and different variances to these 3 images. Then we apply GaussianBlur with different kernel sizes to them and get the loss function so we can compare them.

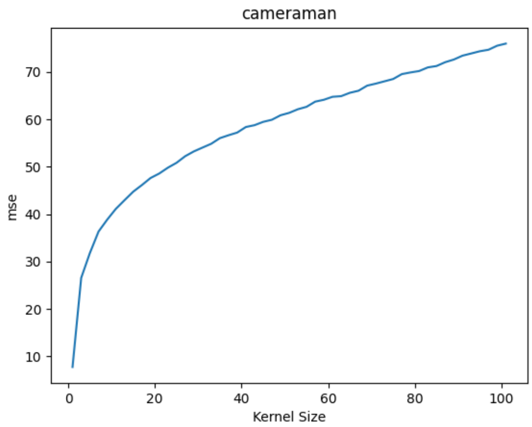
2.1 A

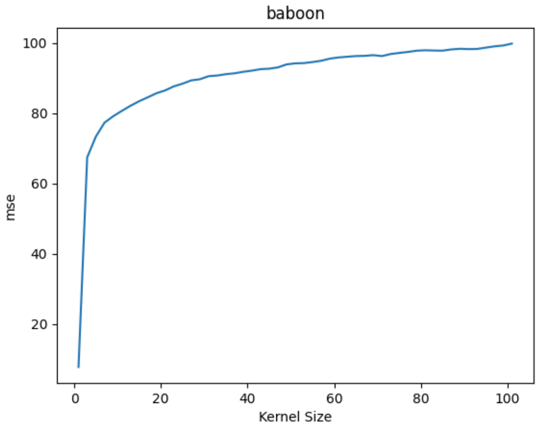
2.1.1 variance=10

First, we use $variance = 10$ for Gaussian Noise. You can see the noisy images here:

We plotted the MSE loss for different kernel sizes for each image:





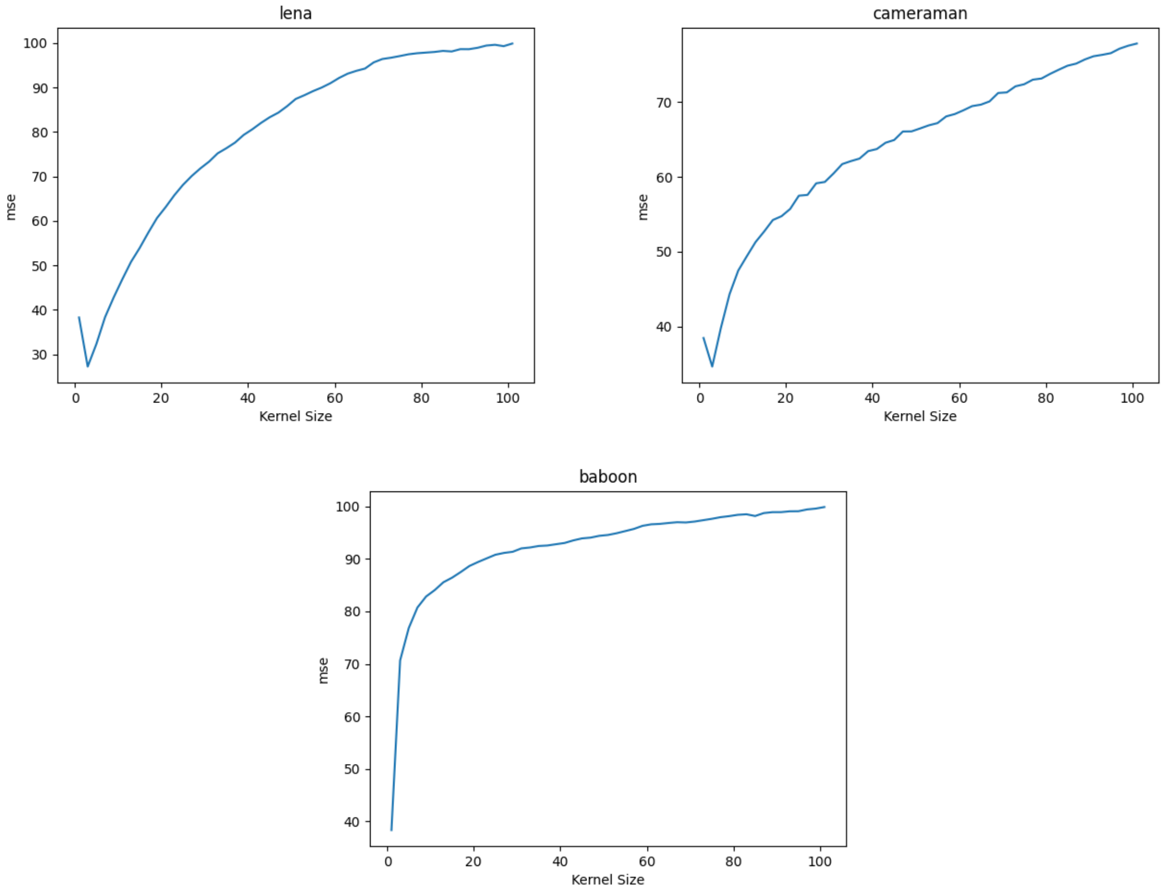


Obviously, the larger the kernel size is, the blurer the image becomes. Therefore MSE grows larger.

2.1.2 variance=50

Then, we use $variance = 50$ for Gaussian Noise. You can see the noisy images here:

We plotted the MSE loss for different kernel sizes for each image:

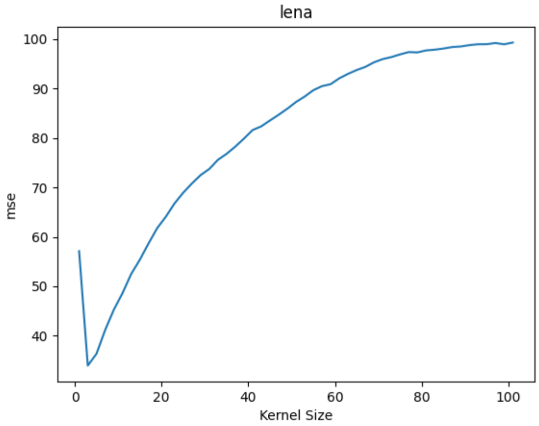


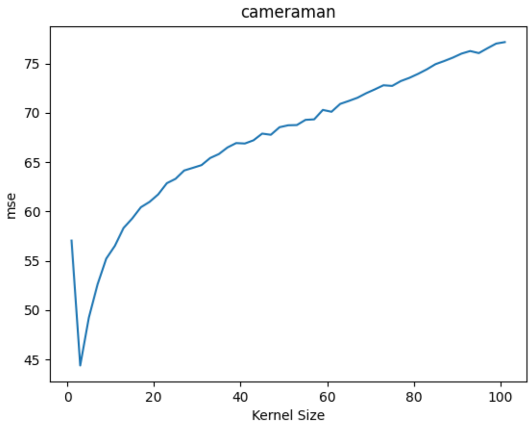
These plots are almost like variance=10, with a little anomaly at the beginning. So this must be because of the added noise. The baboon image's plot has not changed and that might be because the image itself is noisy-like and does not change much with noise.

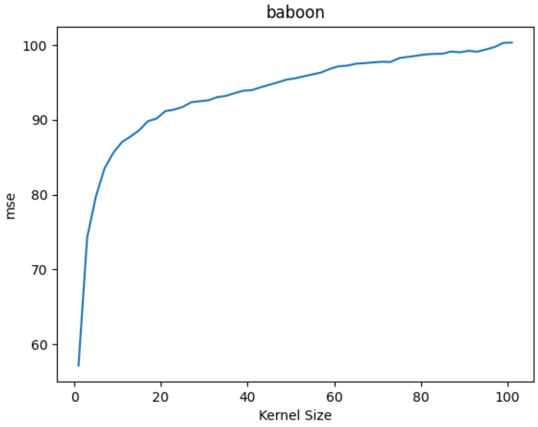
2.1.3 variance=100

Finally, we use $variance = 100$ for Gaussian Noise. You can see the noisy images here:

We plotted the MSE loss for different kernel sizes for each image:







We can see that the anomaly here is more significant than the last part(except baboon). So it most definitely is because of the noise.

2.2 B

This section is like part A, but uses PSNR as loss function which has this formula:

$$\mathrm{PSNR} = 10\log_{10}\left({}^{L^2}/_{\mathrm{MSE}}\right)$$

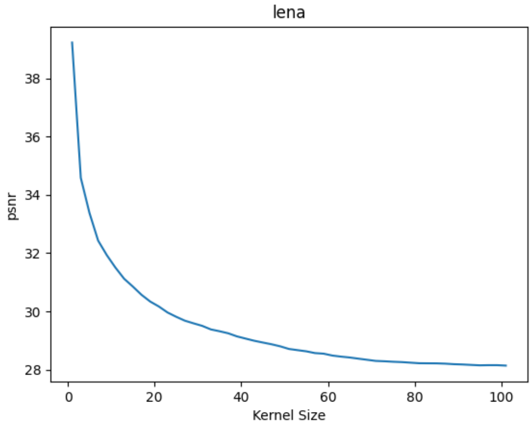
Because $log10$ is increasing and it inputs L^2/MSE, so it should mean that:

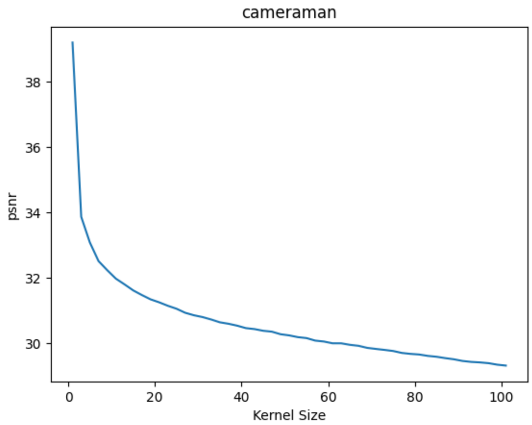
$$PSNR \quad \alpha \quad \frac{1}{MSE}$$

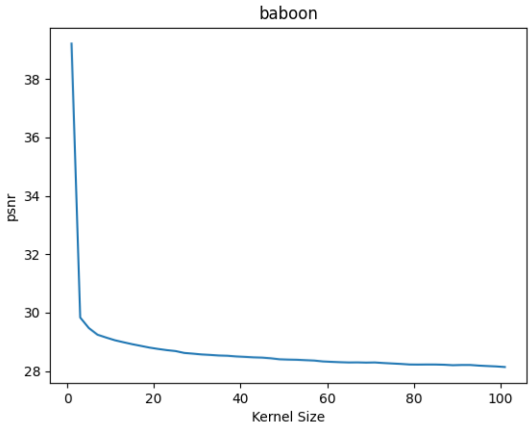
Therefore, although the MSE plot of part A looks like $\sqrt{x}$, PSNR looks like $\frac{1}{\sqrt{x}}$. All those things we said in part A are true here, but the plots are reversed.

2.2.1 variance=10

We plotted the PSNR loss for different kernel sizes for each image:

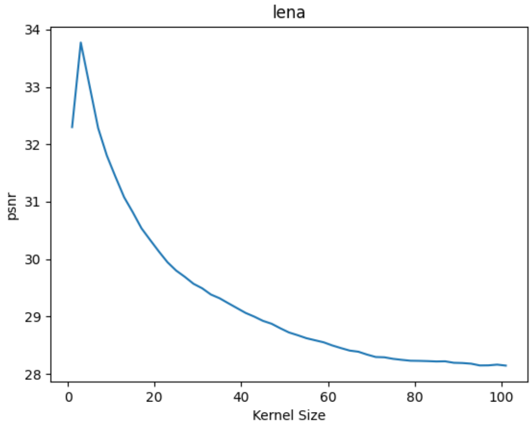


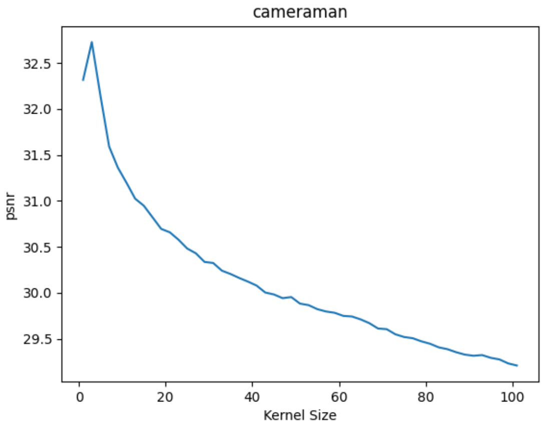


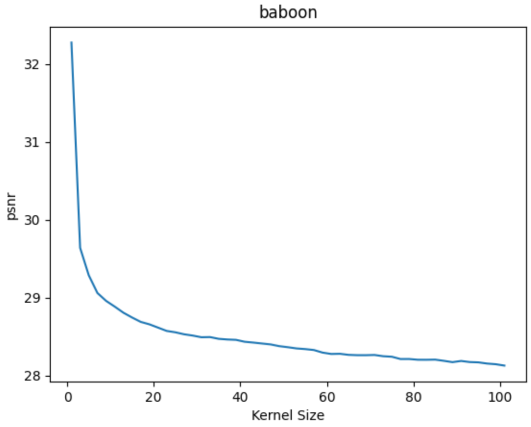


2.2.2 variance=50

We plotted the PSNR loss for different kernel sizes for each image:

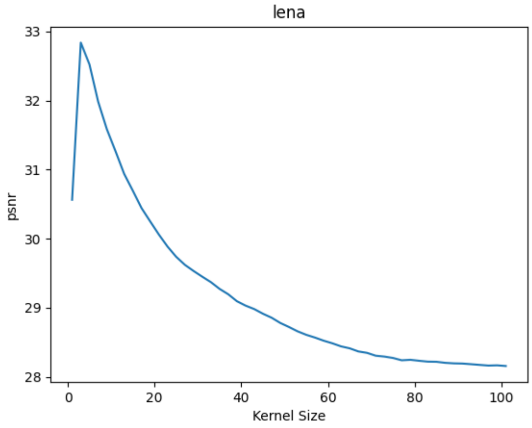


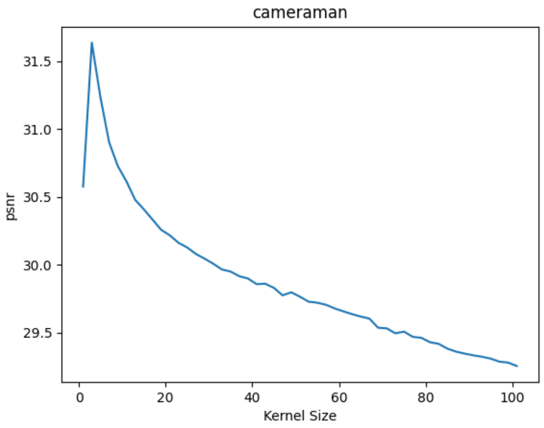


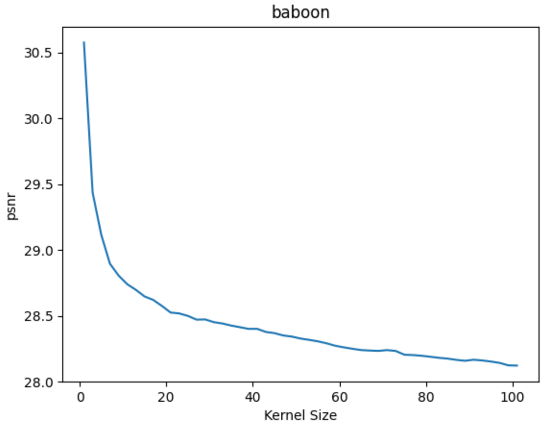


2.2.3 variance=100

We plotted the PSNR loss for different kernel sizes for each image:







As we saw in part A, here, too, the anomaly at the beginning grows more significant as the noise increases.

References

[1] https://scikit-learn.org/

[2] https://stackoverflow.com/

[3] https://towardsdatascience.com/