TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

Fall 2019 - CMPE 362 Digital Image Processing

Assignment 3

Frequency Domain Processing

by

Kaan Üstündağ

16171099958

Part 1: Image Smoothing Using Frequency Domain Filters

In this part of the assignment, I used my *lpfilter.m* implementation for Ideal, Butterworth and Gaussian low pass filters. I used two different cutoff frequencies which are 0.05 and 0.25 values. Original version of the input image that I used in this section is shown in figure 1. The steps of what I have done for each filter are, starting with adding a padding to the input image due to its size, after that I create lowpass filter and apply it to the Fourier transformed image, after that I converted the result to the spatial domain and observed the result for each different filter implementation. All the results and details for each low pass filter will be mentioned below.

Figure 1: "panda.png" Input image of part 1

Before going into details for each low pass filter, I would like to state Fourier spectrum of the input image in figure 2, since in each filter we will observe filter applied Fourier spectrums and compare them.

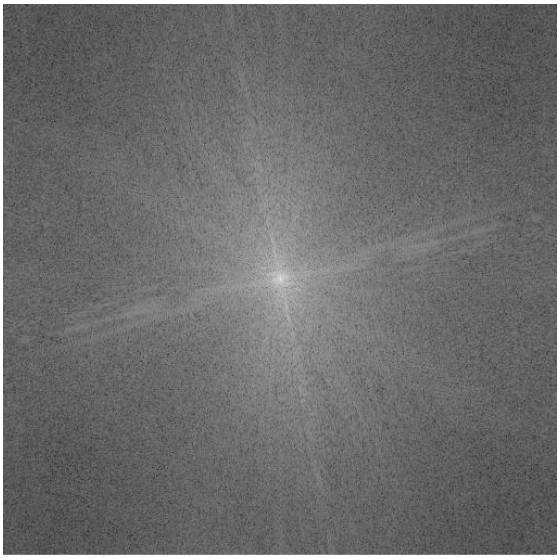

Figure 2: Fourier spectrum of input image

- **Ideal Low Pass Filter:**

So, as I stated before I applied two different cutoff frequencies in my experiment, which are 0.05 and 0.25. The magnitude spectrum result for 0.05 cutoff frequency and its spatial domain result is shown in figure 3. Fourier spectrum result of ideal low pass filter with 0.25 cutoff frequency and the result of the filter is shown in figure 4.

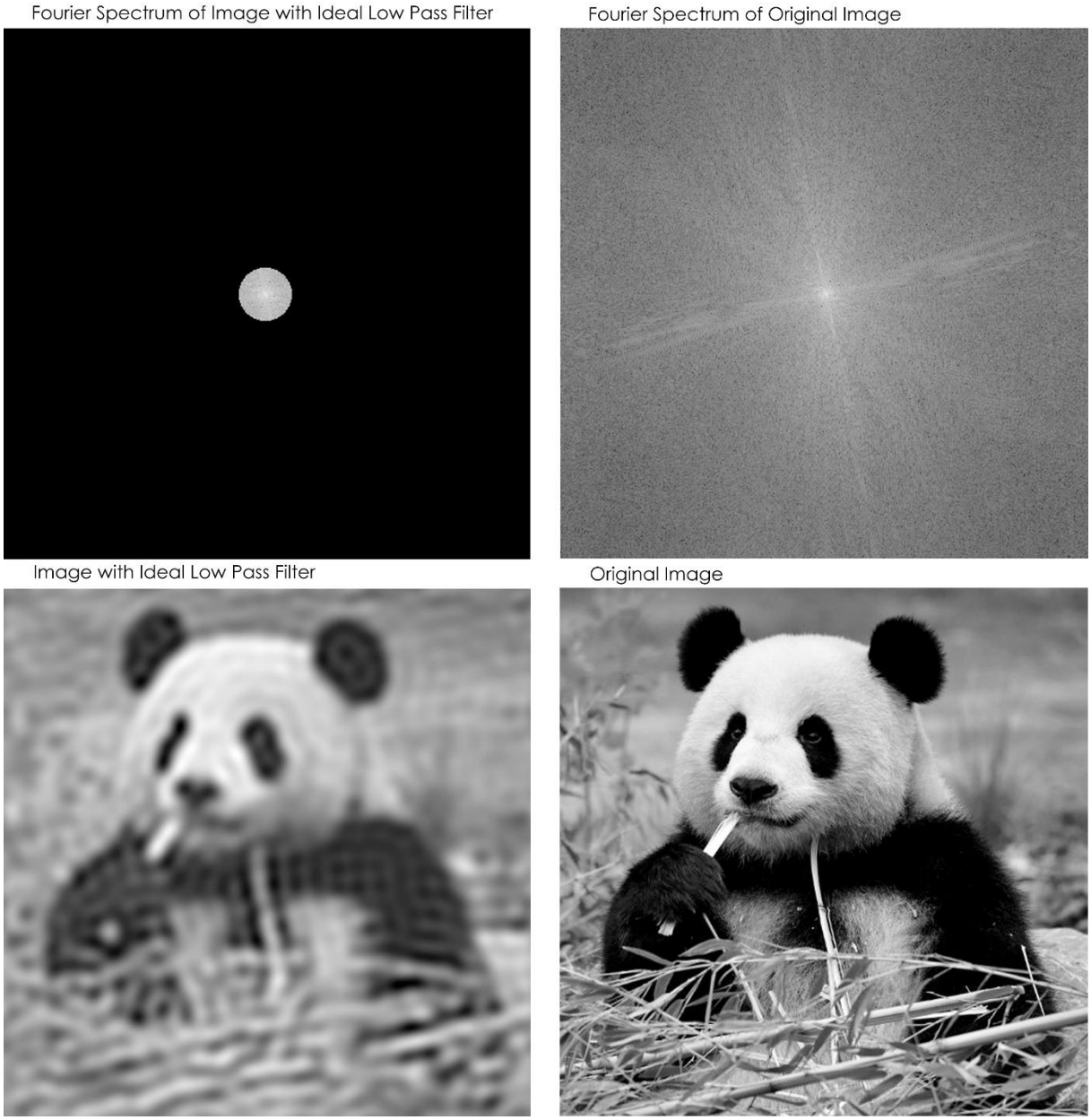

Figure 3: Results for Ideal low pass filter with 0.05

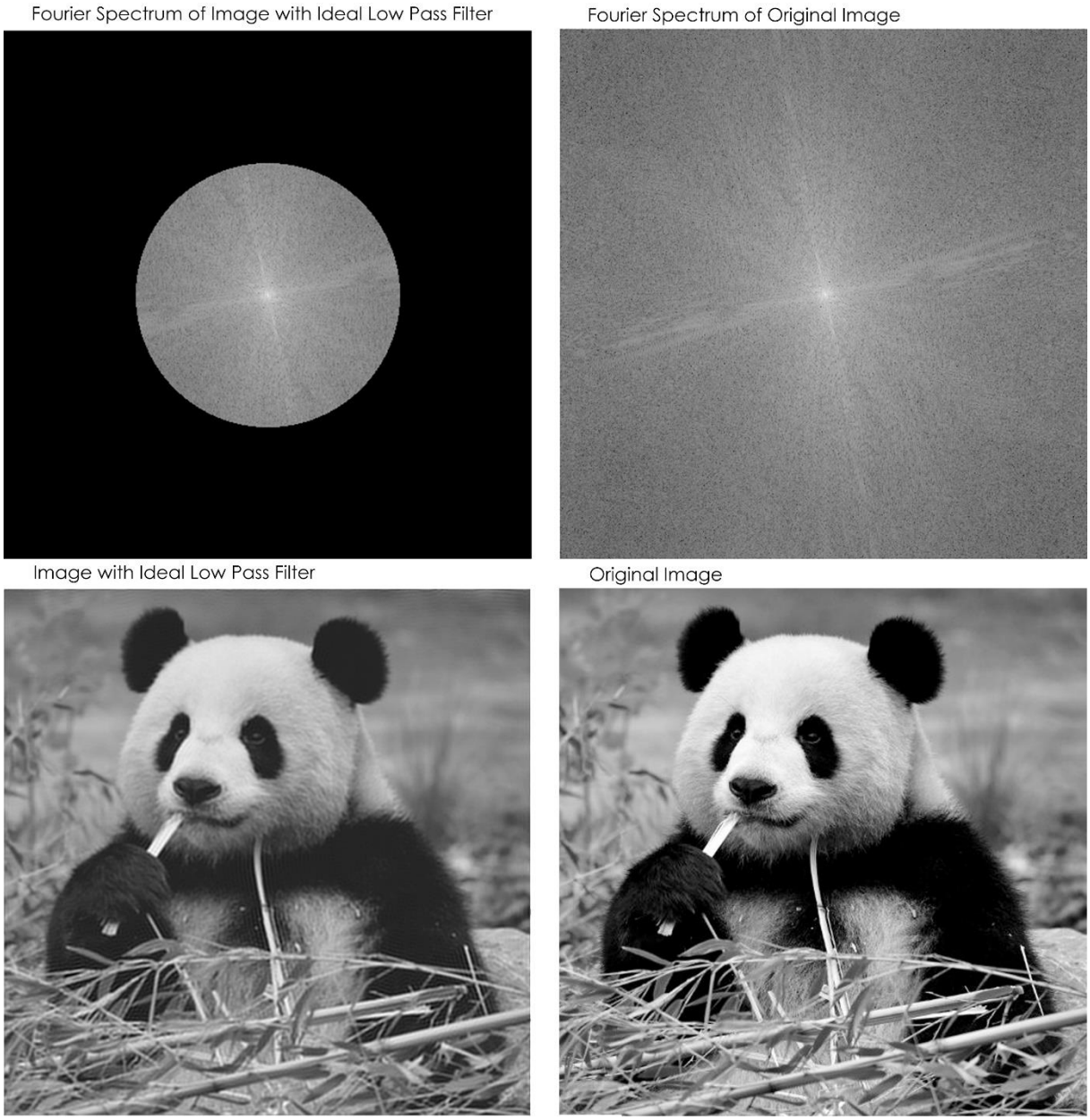

Figure 4: Results for Ideal low pass filter with 0.25

Thus, when we increase the cutoff frequency, we observe that the impact of the filter is decreasing. We can observe from the results that the ideal lowpass filter creates a spectrum with a sharp bordered circle at the center of the image, and the cutoff value is like the radius of this circle.

- **Butterworth Low Pass Filter:**

Again, I applied the same cutoff frequency values with Butterworth low pass filter and observed the following results for the input image.

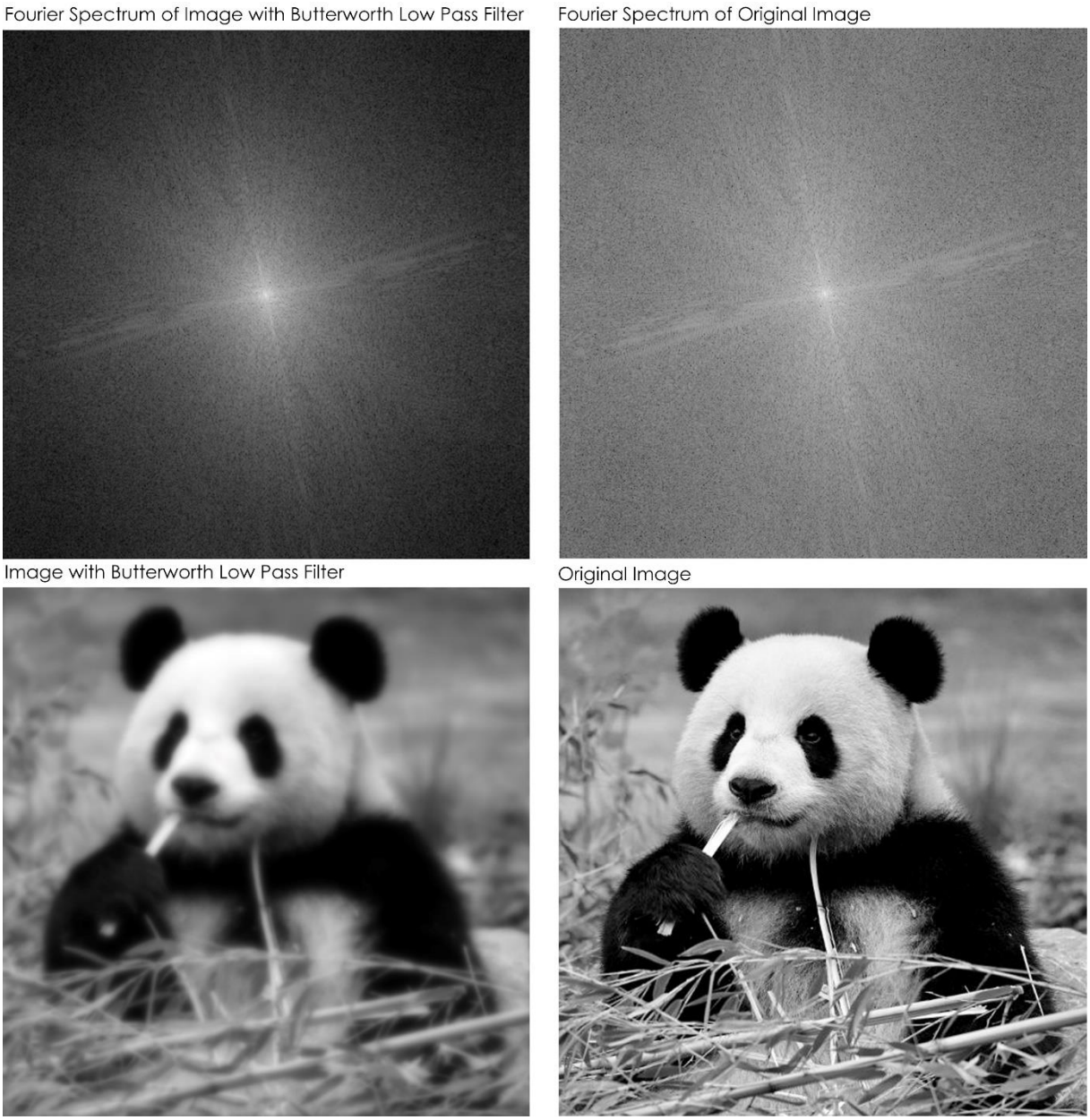

Figure 5: Results for butterworth low pass filter with 0.05

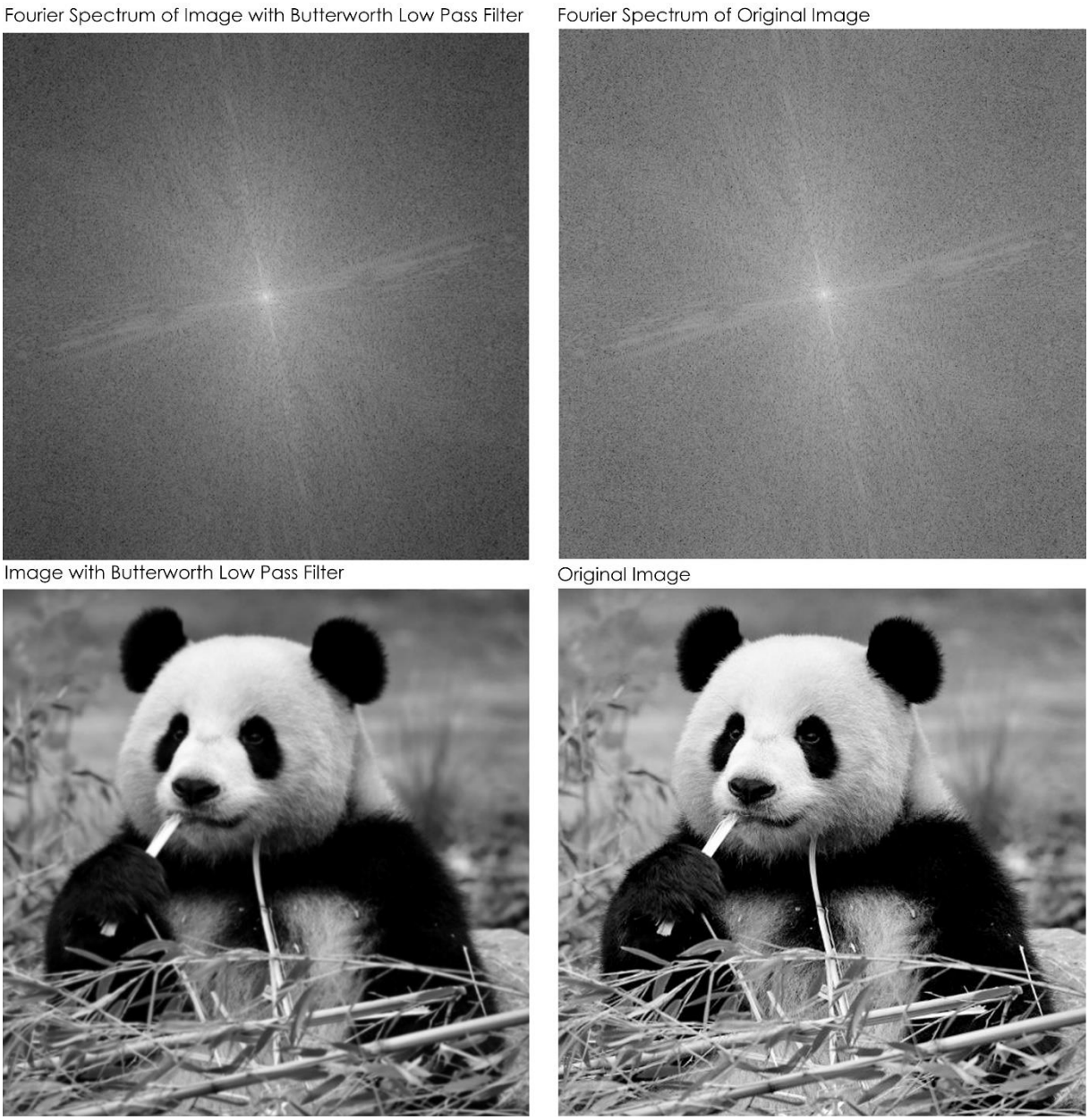

Figure 6: Results for butterworth low pass filter with 0.25

Hence, we observe that with the increase of the cutoff frequency, impact of the filter is again decreasing, and the blur effect is also decreasing.

- **Gaussian Low Pass Filter:**

I applied same cutoff frequency values for Gaussian low pass filter and observed the following results for the filter within spatial domain.

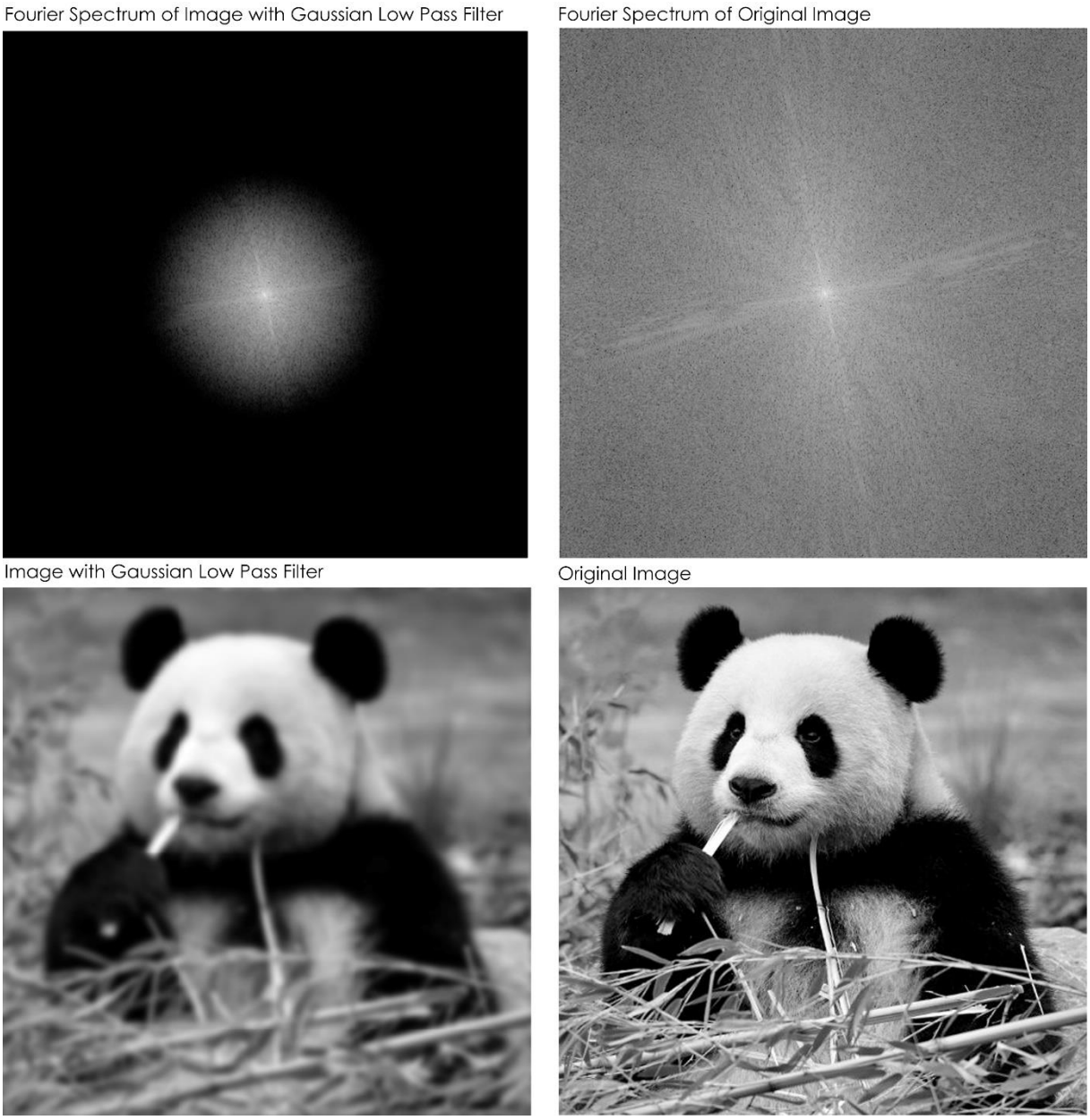

Figure 7: Results for Gaussian low pass filter with 0.05

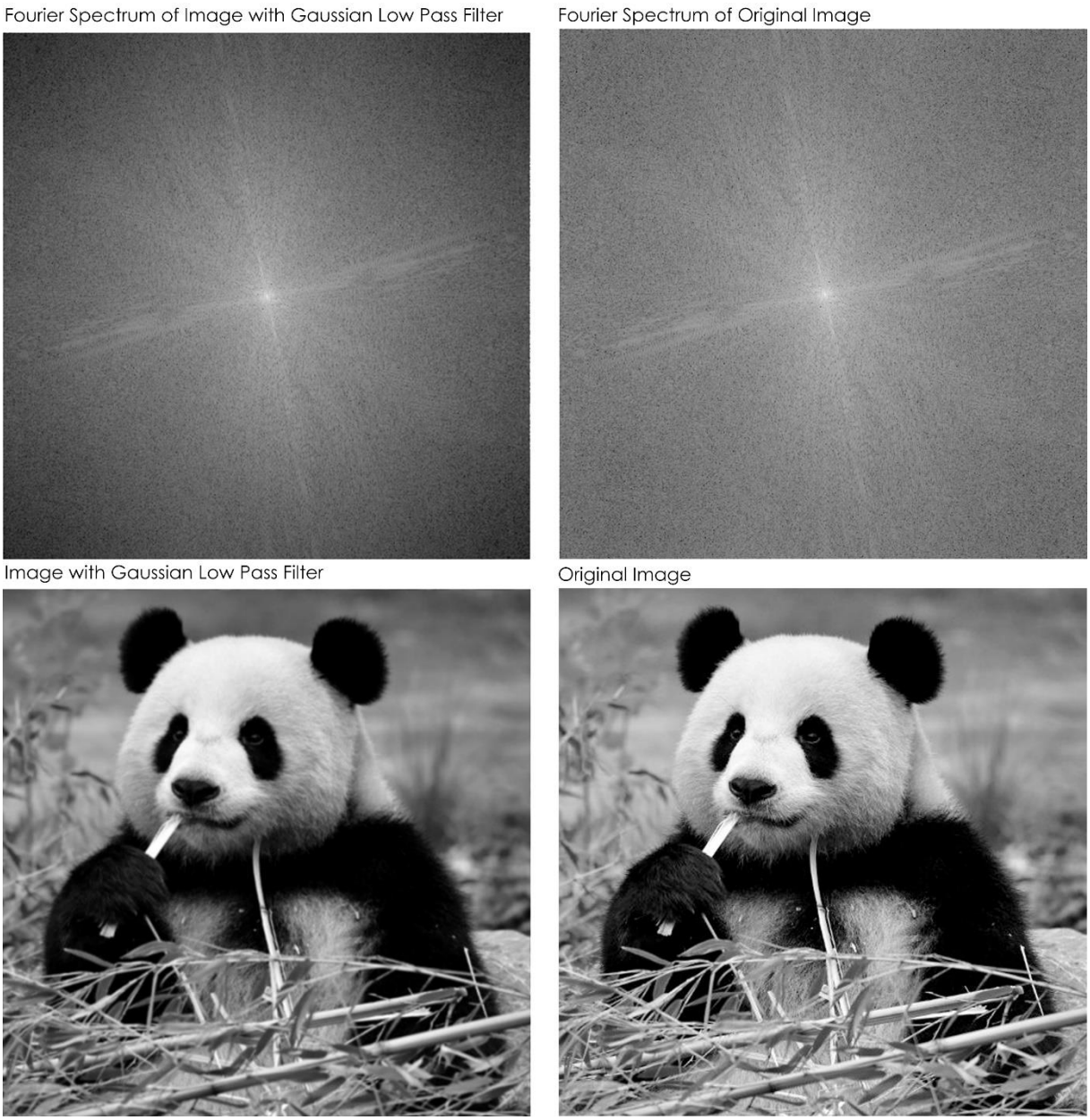

Figure 8: Results for Gaussian low pass filter with 0.25

Hence, we observe that with the increase of the cutoff frequency, impact of the filter is again decreasing, and the blur effect is also decreasing. It is very similar to the Butterworth low pass filter results.

Part 2: Image Sharpening Using Frequency Domain Filters

In this part, I have used the panda.png and for the filtering process I have used *hpfilter.m* implementation for Ideal, Butterworth and Gaussian high pass filtering. Again, in this part I used the same cutoff frequency values for the filtering process. The steps of what I have done for each filter are, starting with padding the input image due to its size, after that I create high pass filter and apply it to the Fourier transformed image, after that I converted the result to the spatial domain and observed the result for each different filter implementation. All the results and details for each high pass filter will be mentioned below.

In my experiment I have used 0.05 and 0.25 cutoff frequency values for all the high pass filters. The results of the high pass filter were just like a reverse (negative) version of the low pass values, which actually makes sense since we simply subtract the low pass filter result from 1 to achieve high pass filter. Thus, all the logic of the high

pass filter is just negative version of low pass filter process, I will just put the results for each high pass filters without detailed explanations.

- **Ideal High Pass Filter:**

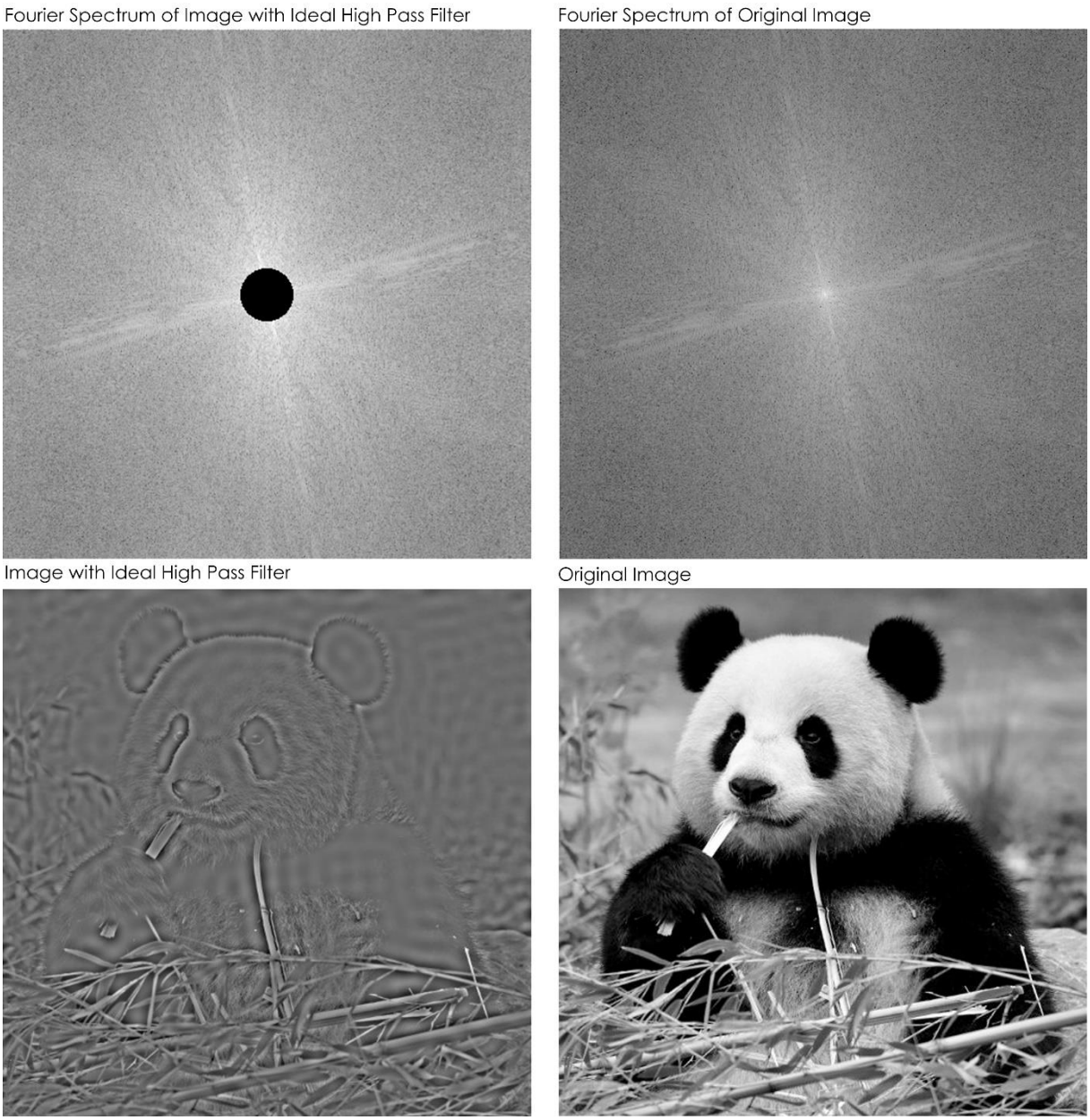

Figure 9: Results for Ideal high pass filter with 0.05

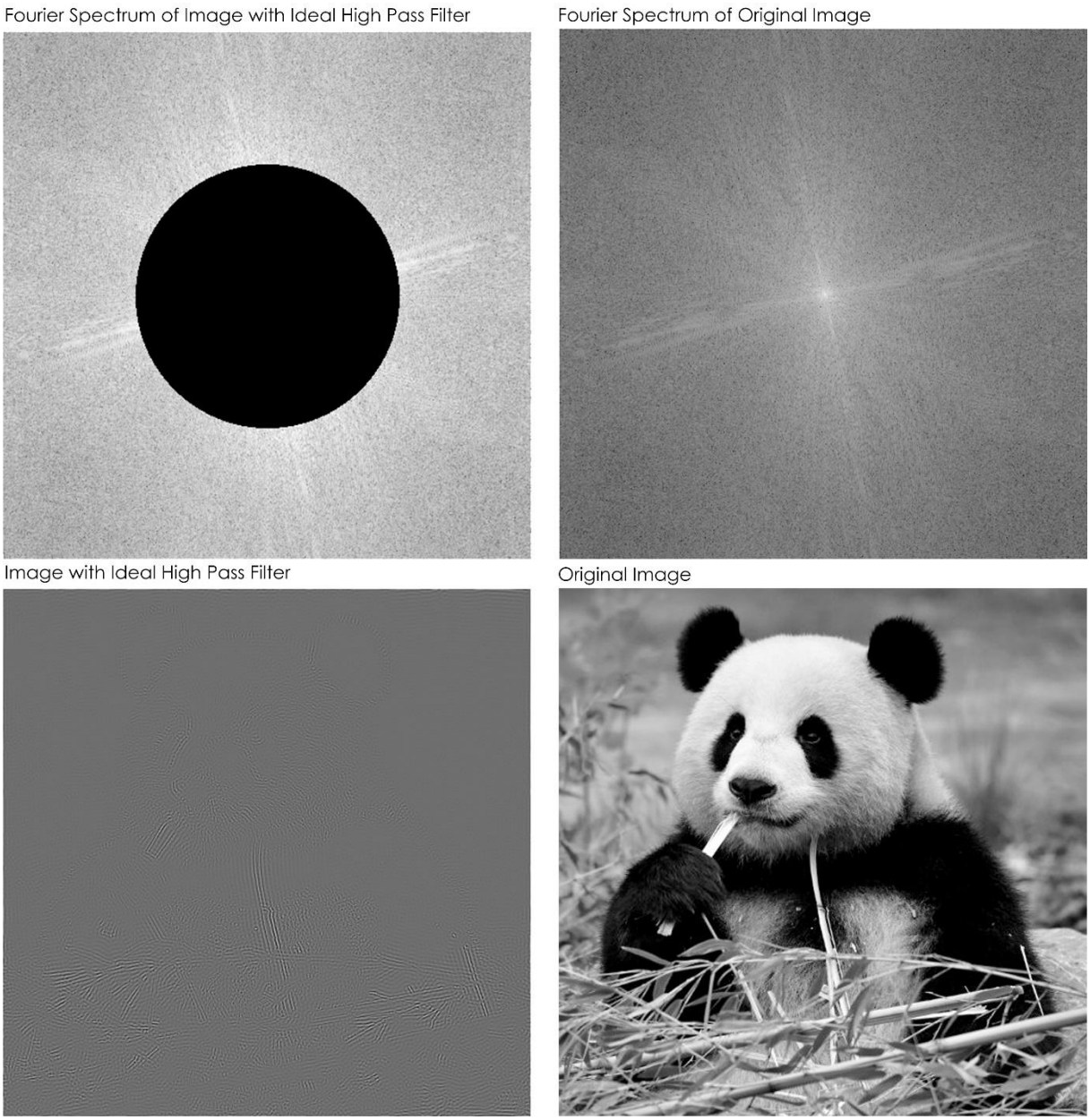

Figure 10: Results for Ideal high pass filter with 0.25

- **Butterworth High Pass Filter:**

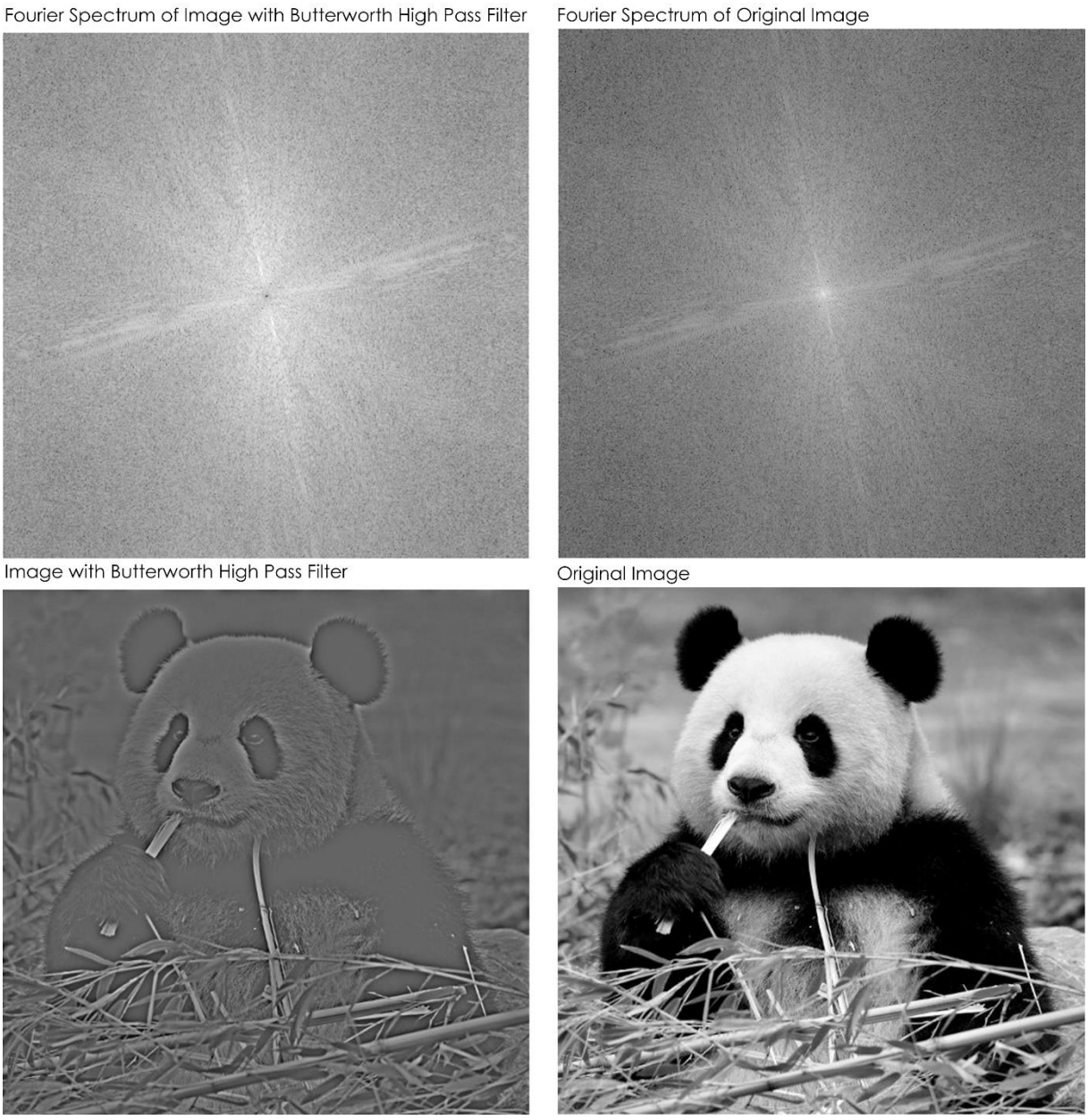

Figure 11: Results for Butterworth high pass filter with 0.05

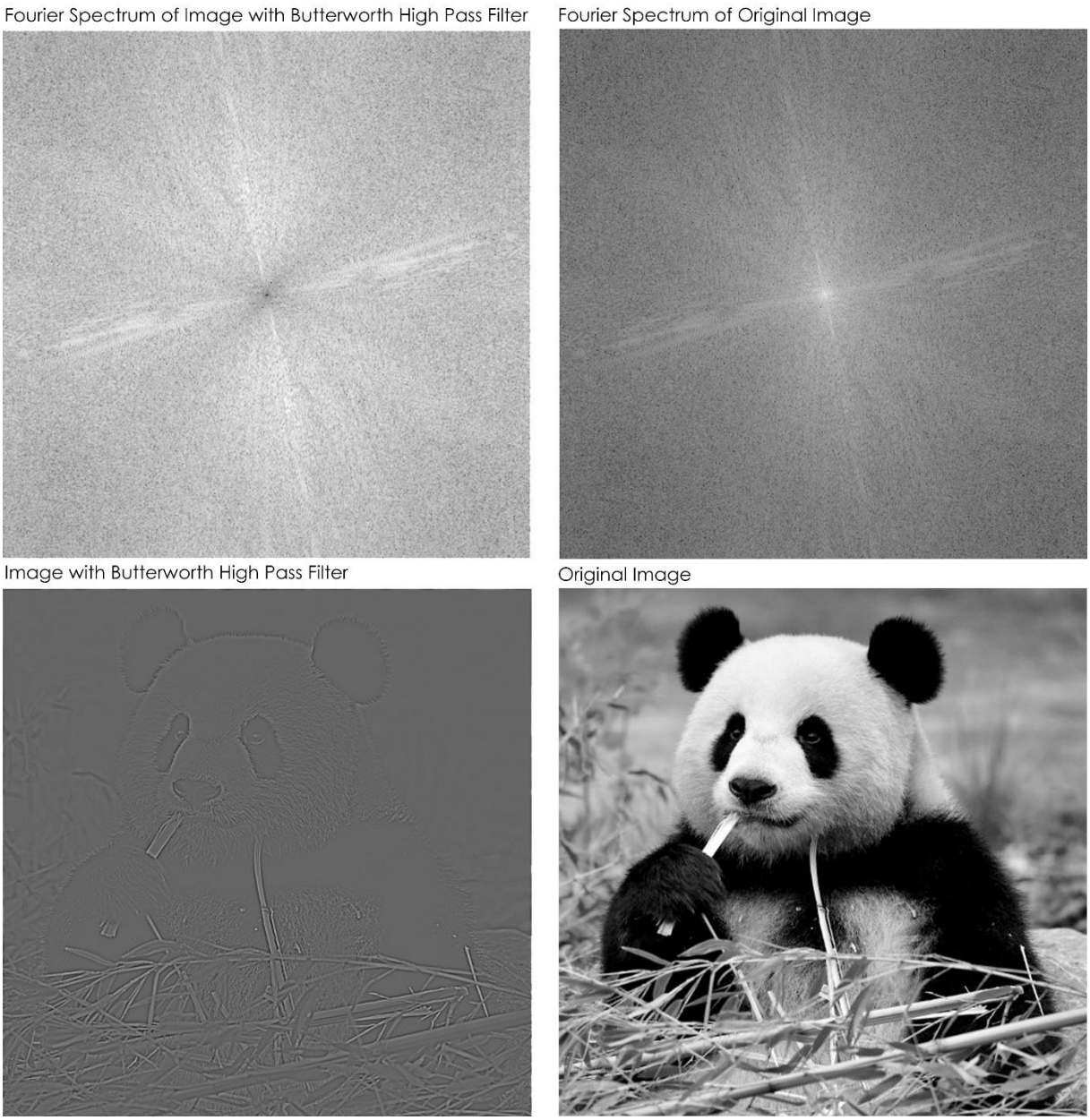

Figure 12: Results for Butterworthhigh pass filter with 0.25

- **Gaussian High Pass Filter:**

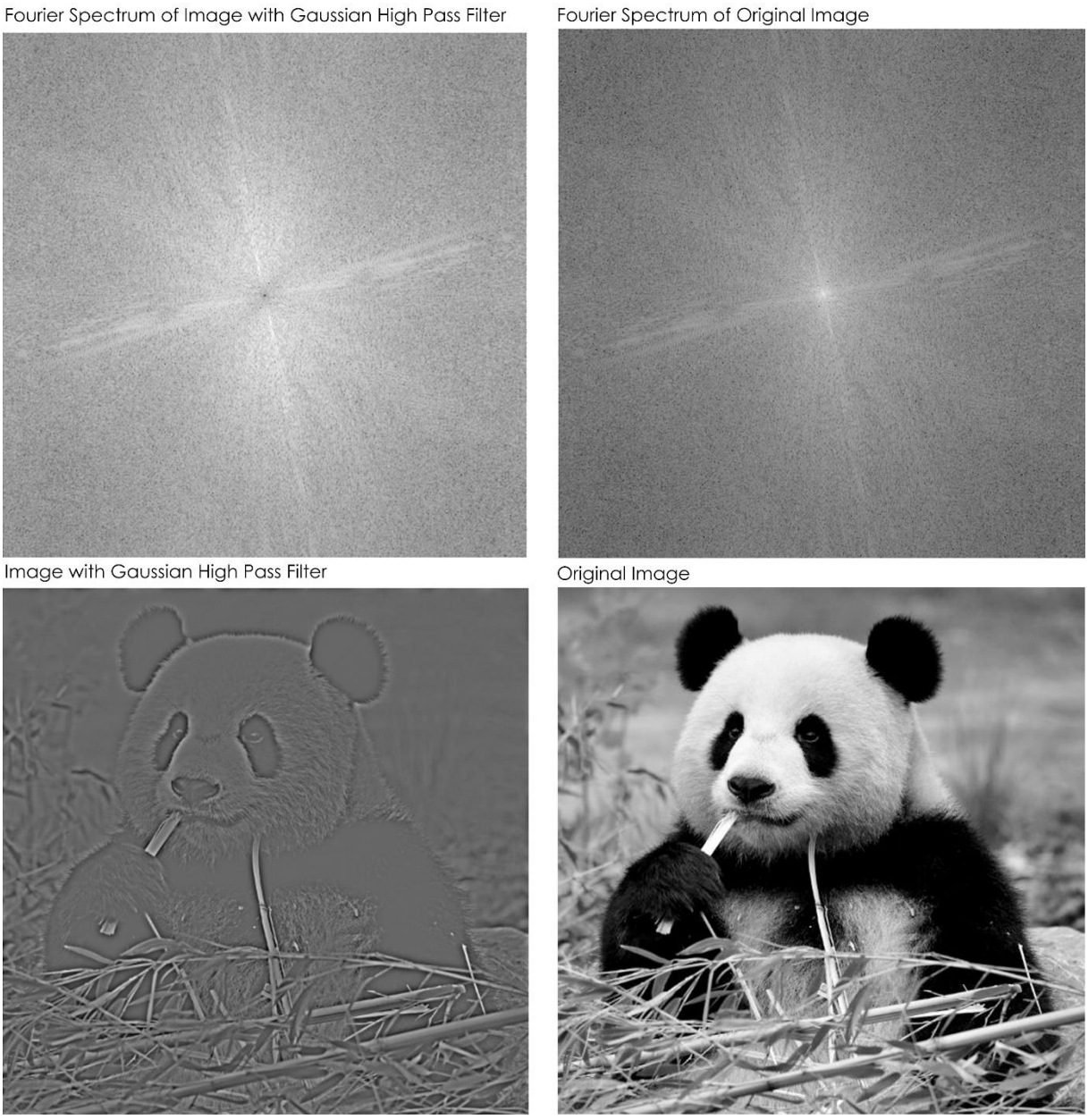

Figure 13: Results for Gaussian high pass filter with 0.05

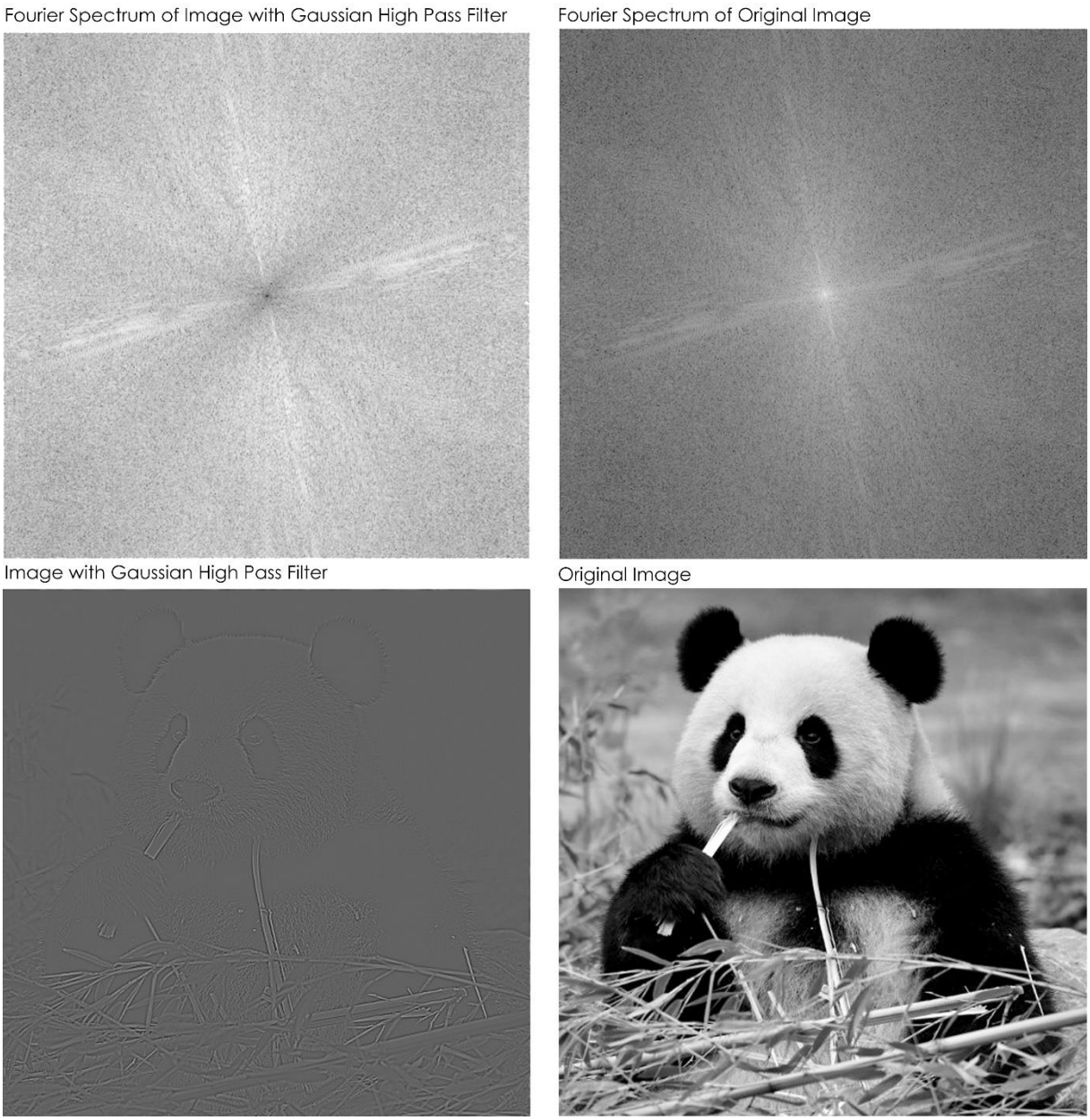

Figure 14: Results for Gaussian high pass filter with 0.25

Part 3: Selective Filtering - Notch Filter

In this part, I have used 3 different input images and applied my notch filter implementation that contains a manual process for drawing a black circle or square on corrupting noises within the input images. Other than that, I applied the notch filter to each image and observed a better version of them, ball.png had a great result since its noises were more obvious, other two input images were again had better results than the original versions, however these two images had less obvious noises within their Fourier transforms. The transform of the all input images and their Fourier spectrum changes are shown below.

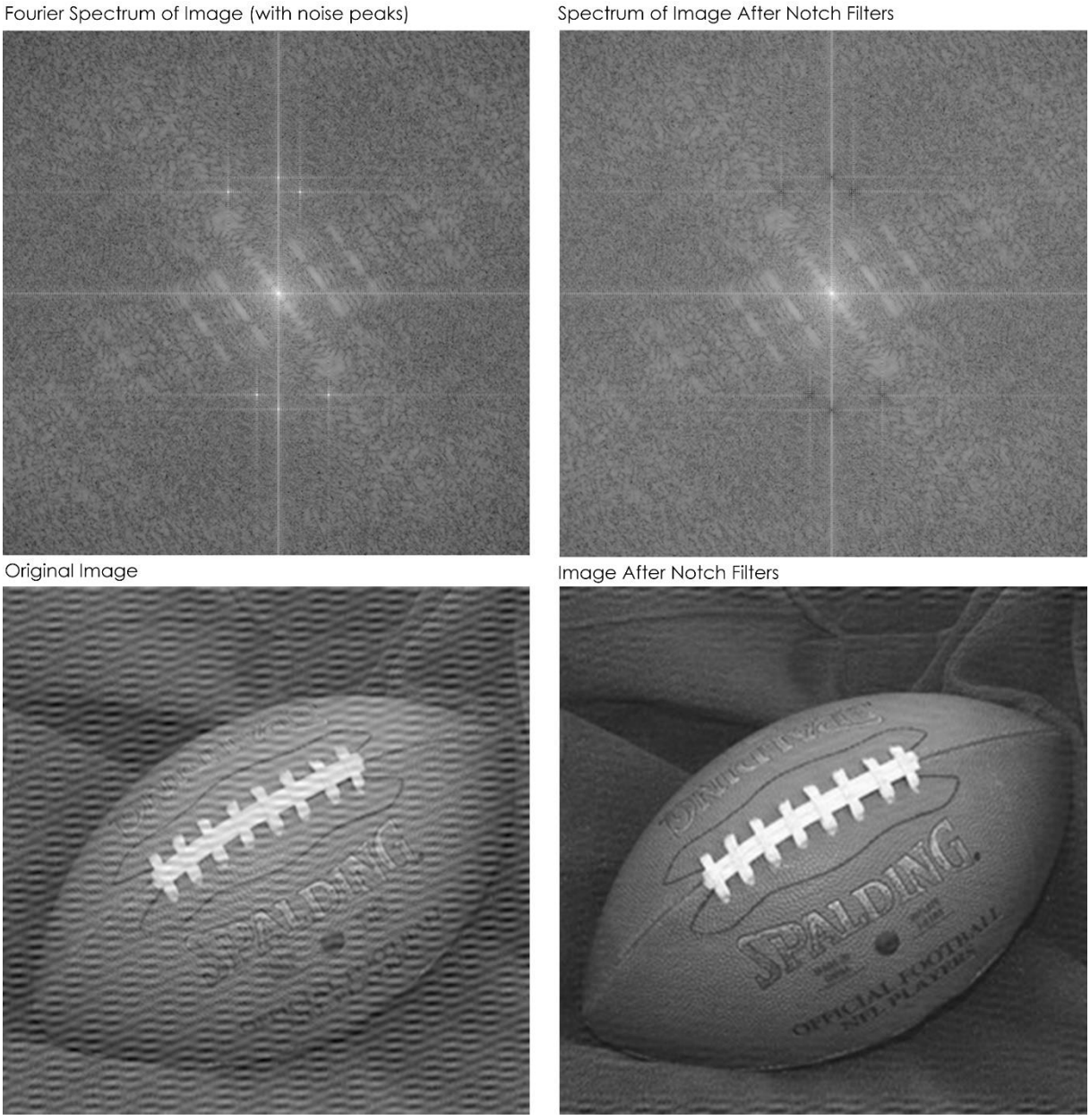

Figure 15: Notch Filters applied to the ball.png

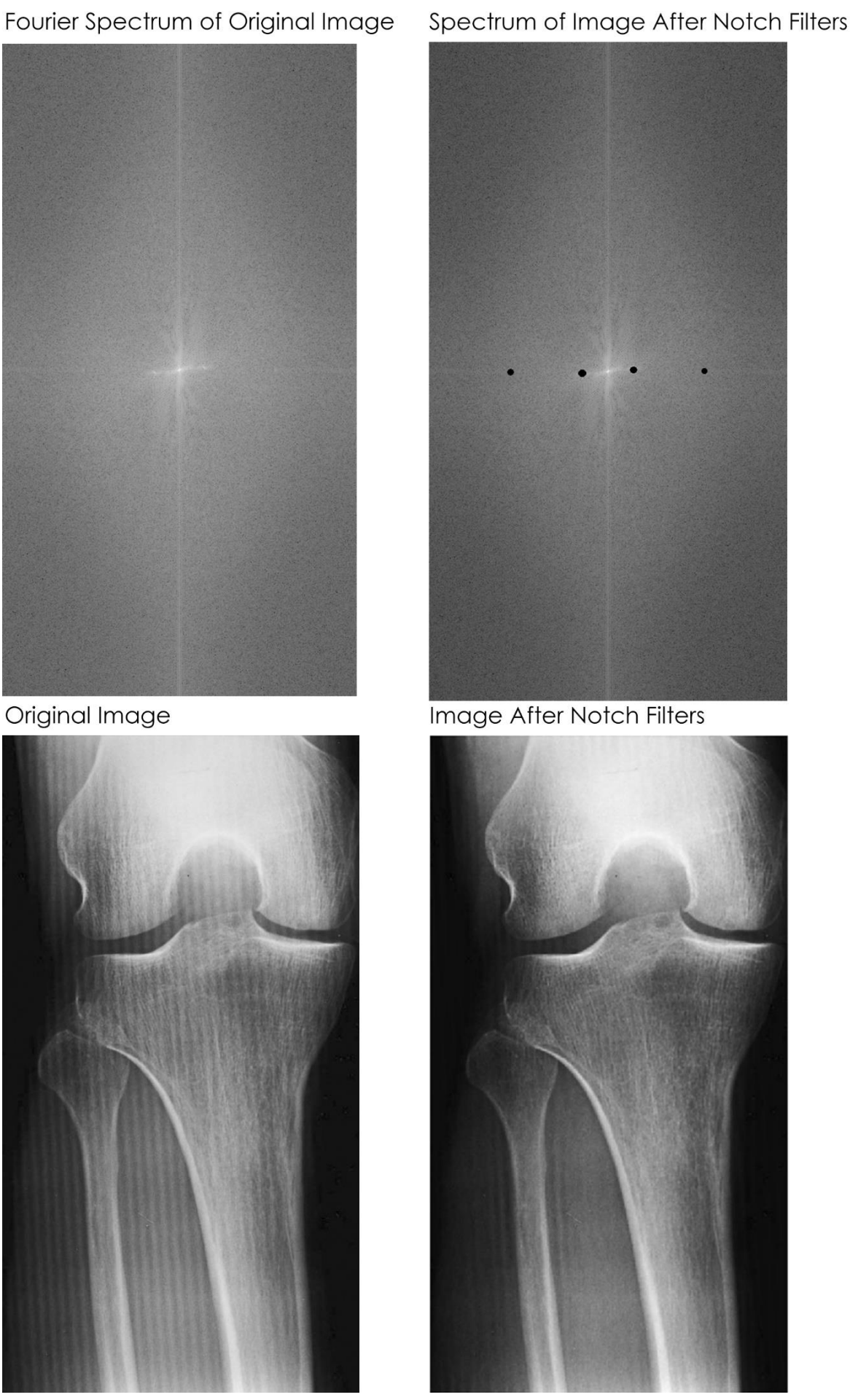

Figure 16: Notch Filters applied to the bone.png

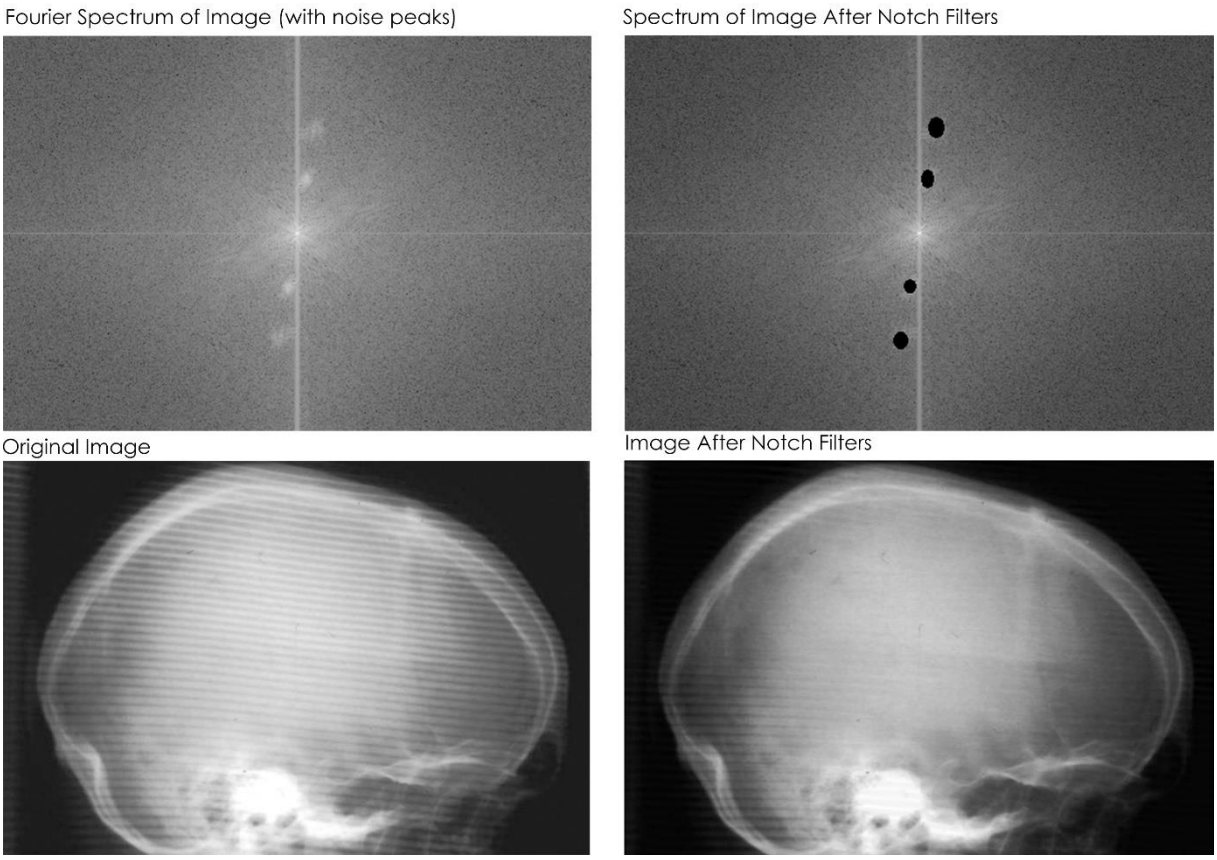

Figure 17: Notch Filters applied to the skull.png

In figure 16 and 17, I drew the black circles by myself to show the noises that barely observable, and in my *part3.m* you need to draw circles or squares to the noise positions to apply the notch filter. (Just like in the example of our slides.)

All in all, in part 3 experiment, I observed that with notch filters that has been applied to the direct locations of the Moiré patterns, the result images are way clearer, and it feels like there is almost no noise within the image. Especially, in ball.png input image, the difference between the original version and the notch filtered versions are so obvious to observe.