In The name of God

University Tehran

Engineering Facility

Electrical and Computer Engineering

Digital Signal Processing

CA4

Ashkan Jafari Fesharaki

810197483

Fall 99

Part 1: Application of filter design in audio processing

In this part, we are going to plot the audio signal.

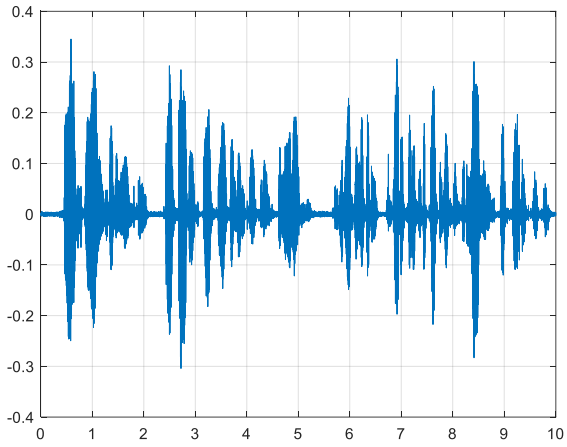

Figure 1

Figure 1 shows the audio signal of the main sound.

After that, we need to add a random noise to the signal with the magnitude of 0.1.

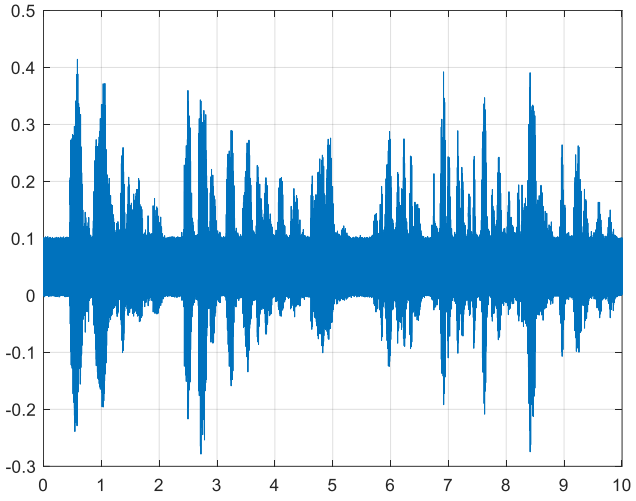

Figure 2

Figure 2 shows the audio signal after adding the additional noise that we assumed before.

We will use the *fdatool* in order to design a filter that will get rid of the noise from the past part.

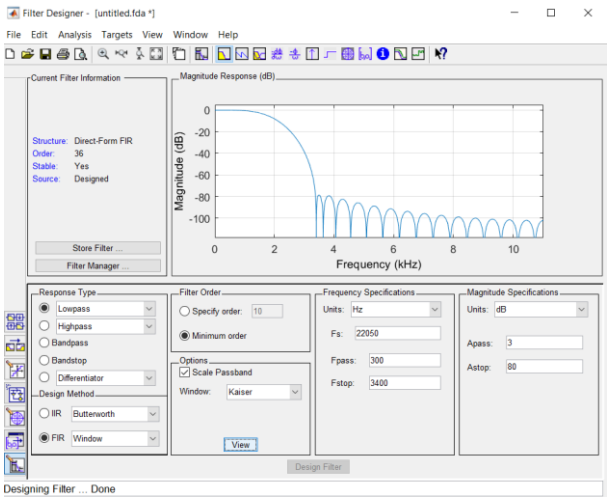

Figure 3

Figure 3 shows the setting that we assumed for the task.

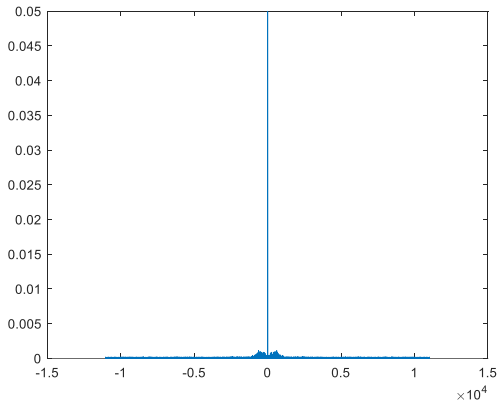

Figure 4

Figure 4 shows the signal in the frequency domain.

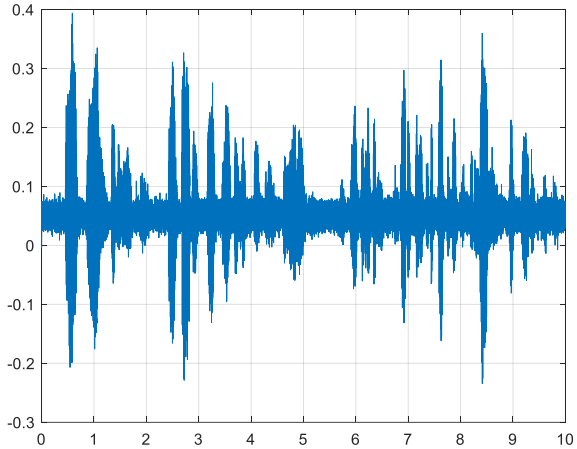

Figure 5

Figure 5 shows the result after applying the filter. As we can see the magnitude of the noise has reduced.
As the result by using the take in the given and using, the fdatool in Matlab we are able to get rid of the noise.

Part 2: Application of filter design in image processing

1) In this part, we are going to design a function that will show the result after applying a filter with some particular properties.

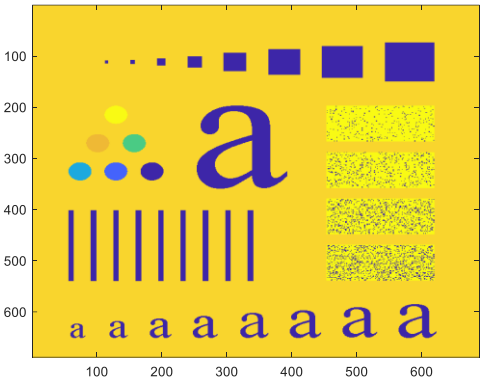

Figure 6

Figure 6 shows the main image.

All the following images show the result after the applying the filter and the result in the frequency domain.

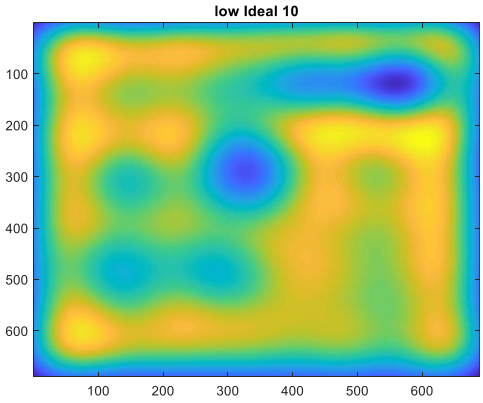

Figure 7

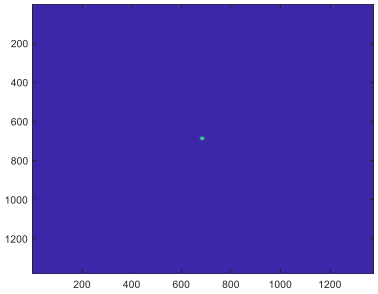

Figure 8

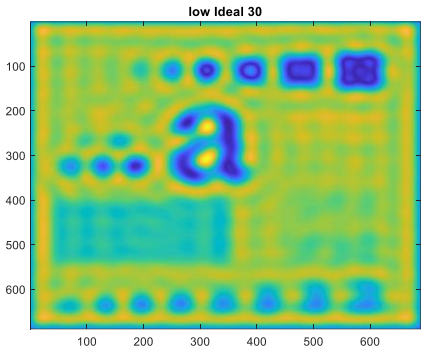

Figure 9

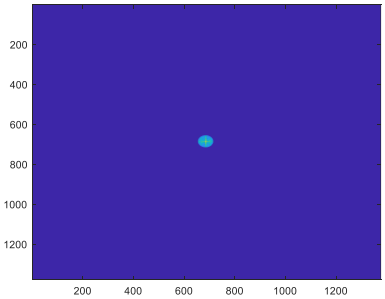

Figure 10

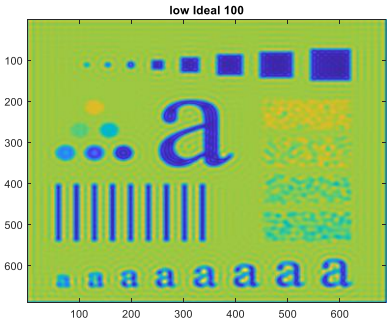

Figure 11

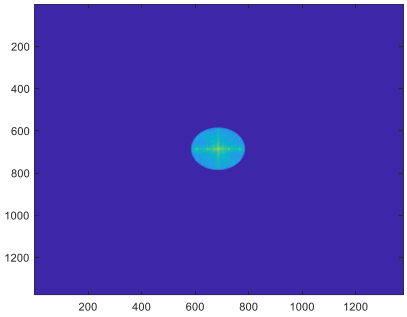

Figure 12

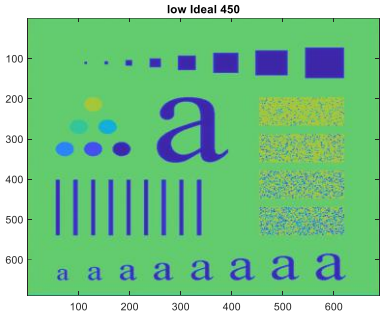

Figure 13

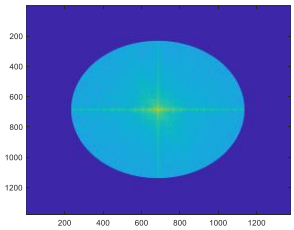

Figure 14

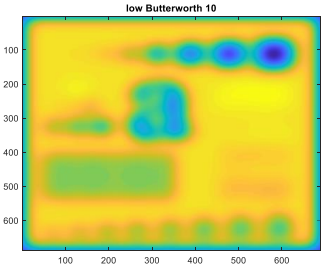

Figure 15

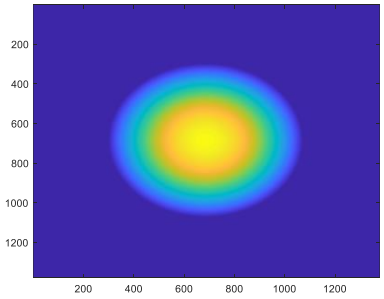

Figure 16

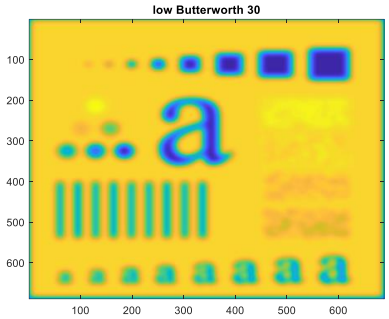

Figure 17

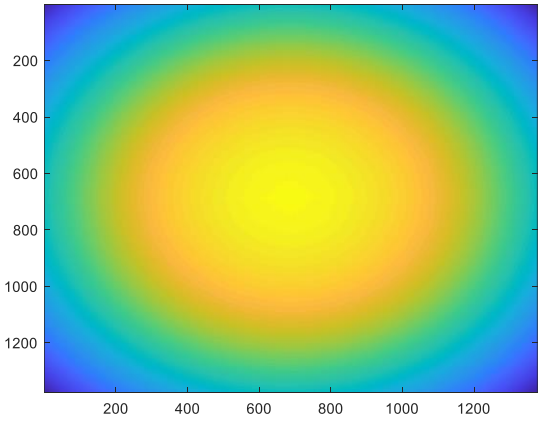

Figure 18

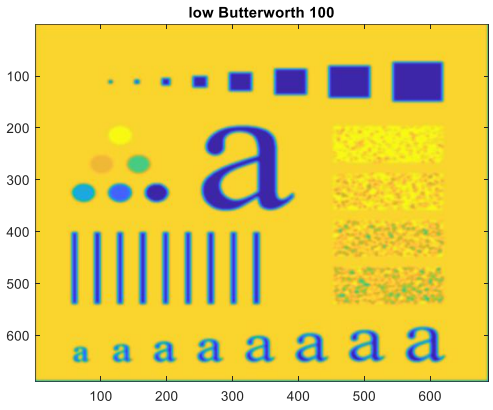

Figure 19

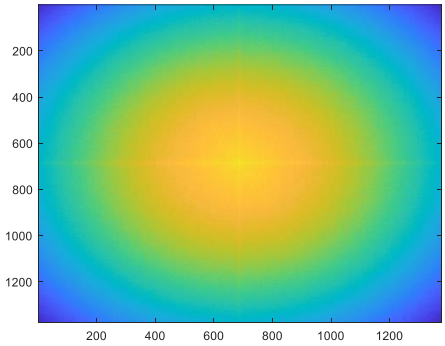

Figure 20

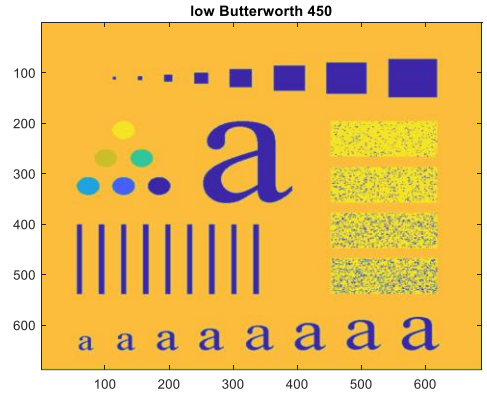

Figure 21

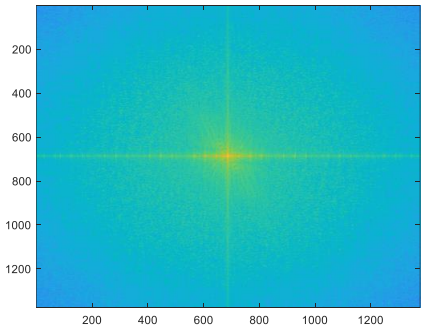

Figure 22

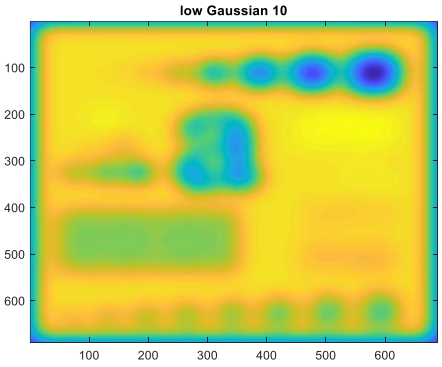

Figure 23

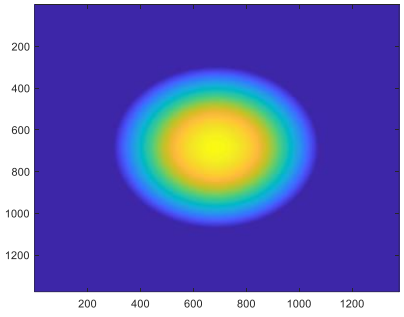

Figure 24

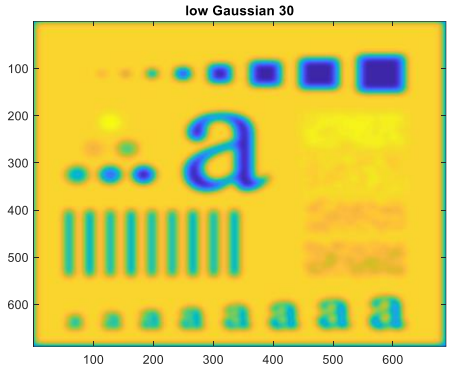

Figure 25

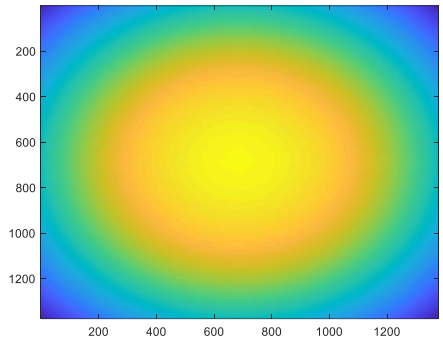

Figure 26

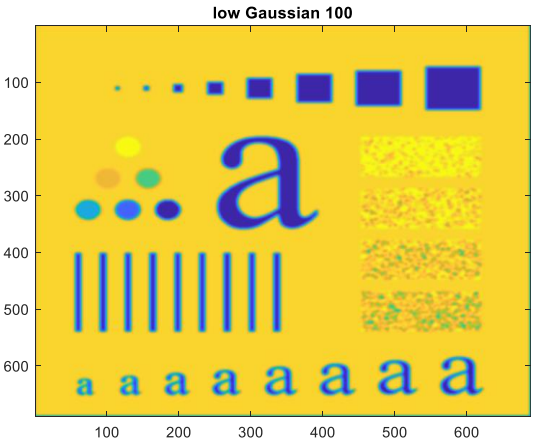

Figure 27

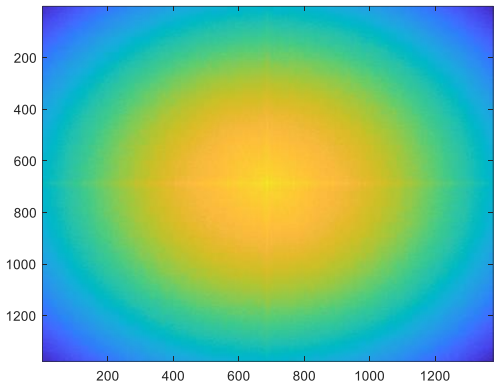

Figure 28

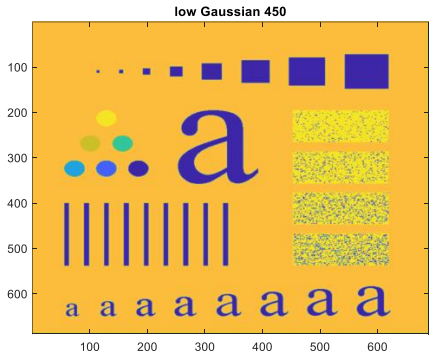

Figure 29

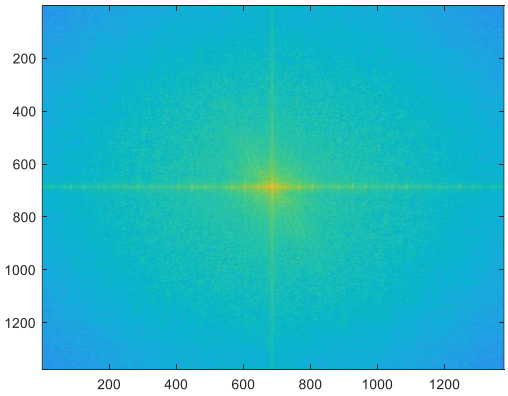

Figure 30

2) All the previous images are for the low pass filter, now we are going to plot the result for the high-pass filters.

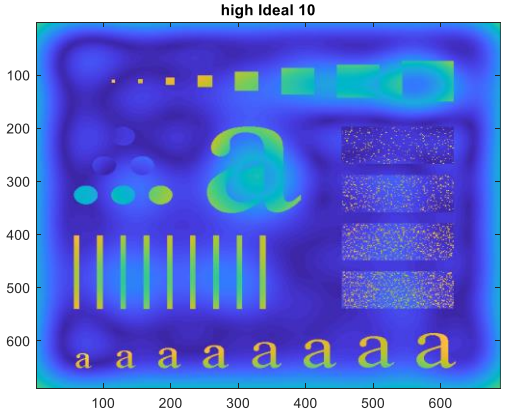

Figure 31

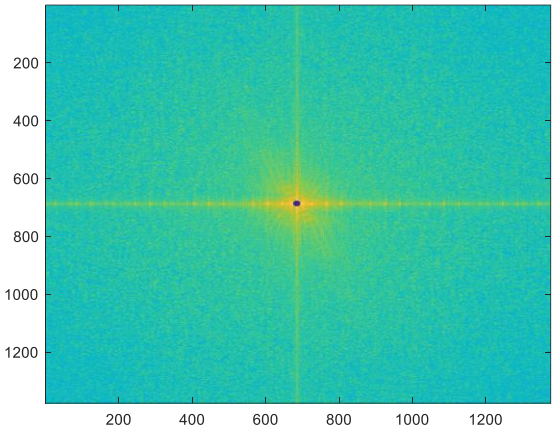

Figure 32

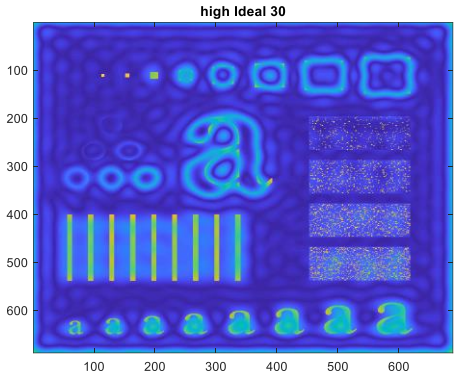

Figure 33

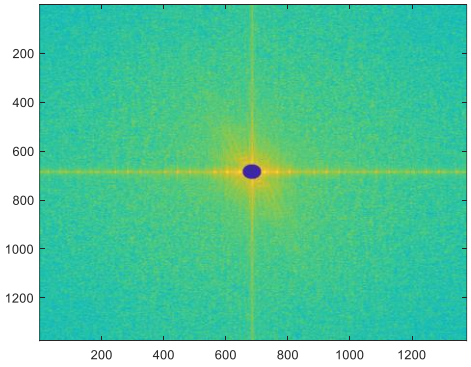

Figure 34

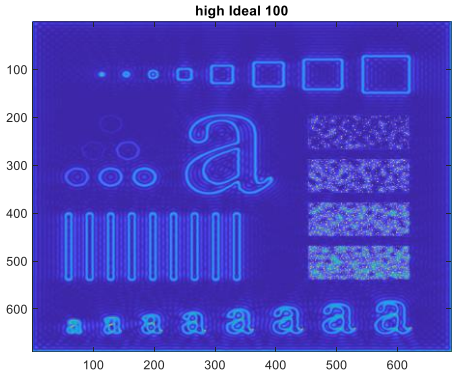

Figure 35

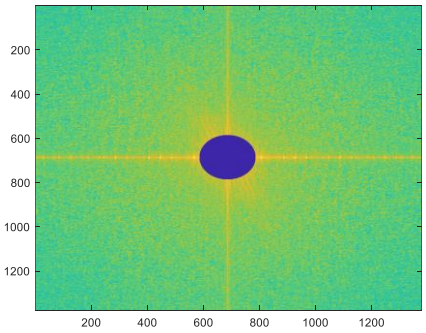

Figure 36

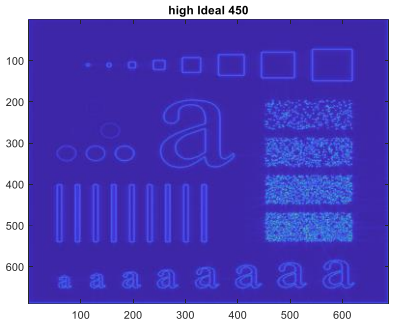

Figure 37

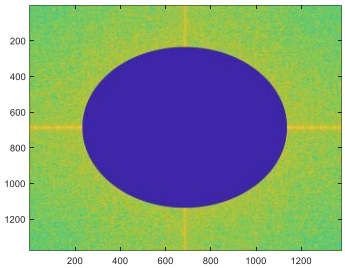

Figure 38

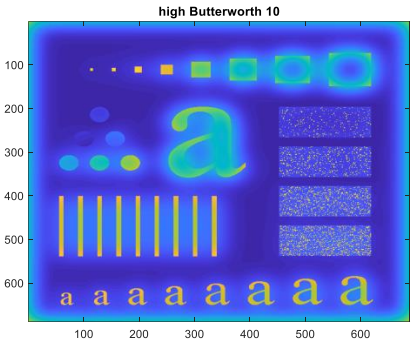

Figure 39

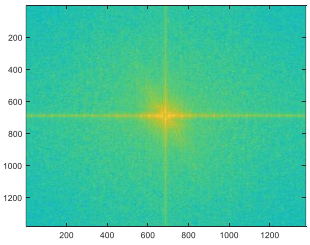

Figure 40

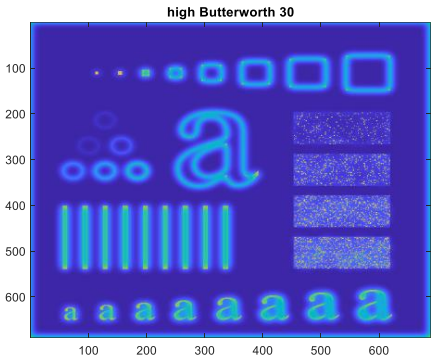

Figure 41

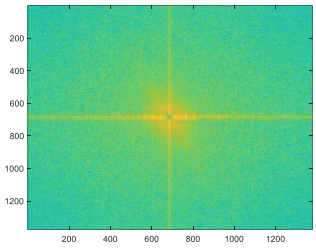

Figure 42

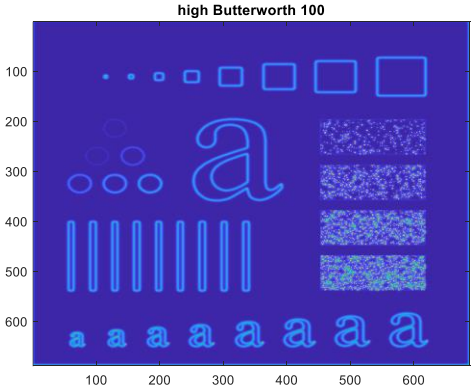

Figure 43

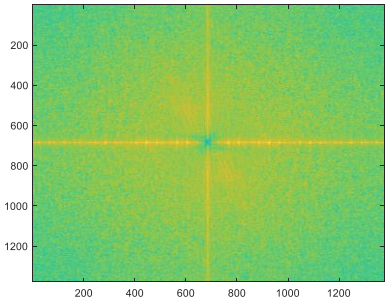

Figure 44

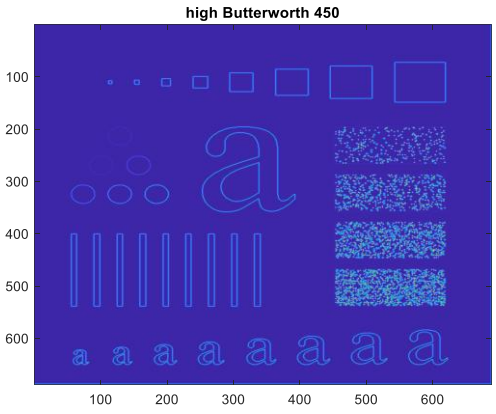

Figure 45

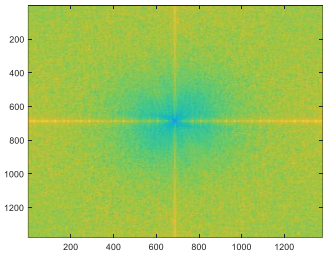

Figure 46

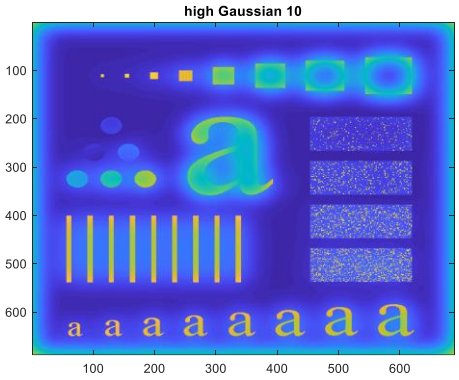

Figure 47

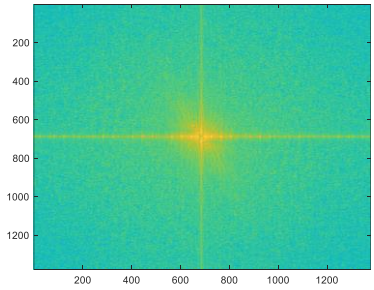

Figure 48

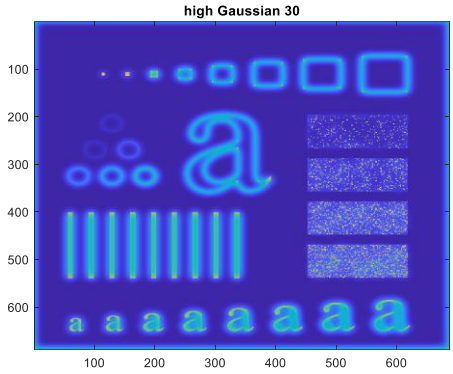

Figure 49

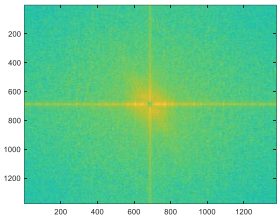

Figure 50

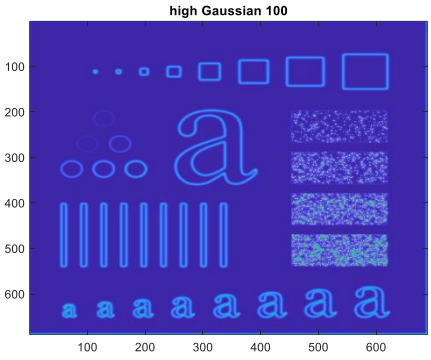

Figure 51

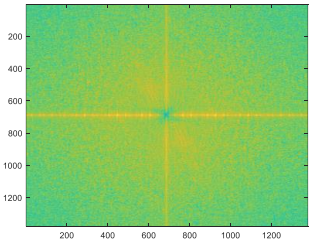

Figure 52

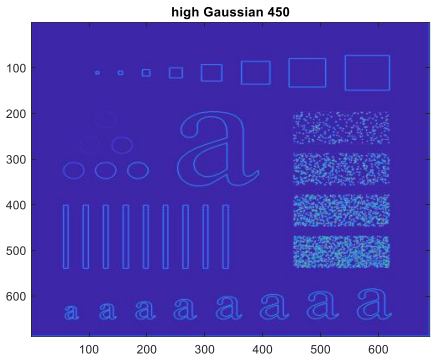

Figure 53

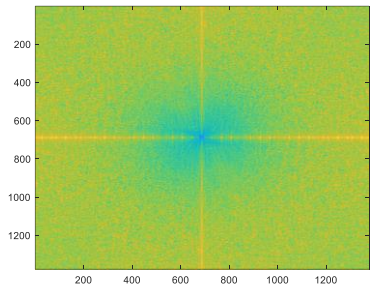

Figure 54

We can see the result for the high pass filter we have the high frequency in our image.

3) The reason that we use fftshift is that it will change the locations of the frequency domain sections in a particular position that we can understand it more easily. For example in the figures mentioned before all DC point is in the middle, so we can use the filter much easier compared to the time when we don't use the fftshift. Implementing and using a filter when we have the result of fftshift is much easier.
4) In the part we are going to see what does the zero-padding do to the image, so we will use a black and white image mentioned the question and see the result after applying the filter.

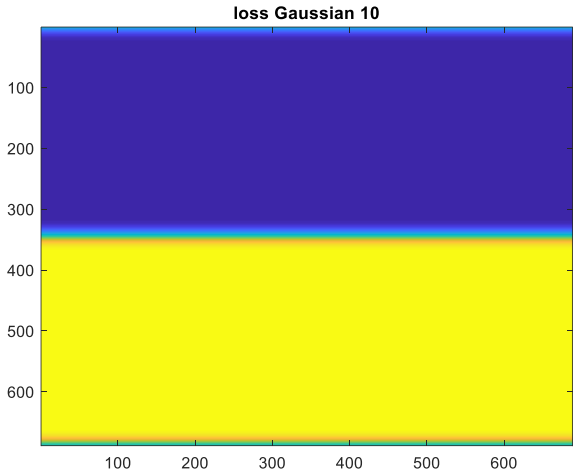

Figure 55

Figure 55 shows the image after a low pass filter when we did not used zero padding.

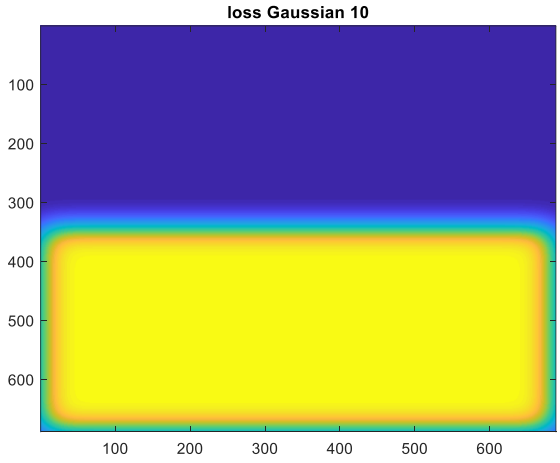

Figure 56

Figure 56 shows the image after applying the low pass filter when we used the zero padding.

As you can see when we used the zero padding, the image has some soft surface near the edge of the image, since when we zero pad the image it will have big changes near the edge of the black parts and therefore the filter will try to smooth that part.

Filtering using MATLAB algorithm and kernel matrix

1) In this question, we are going to see the effect that the filter we are going to us have on the image.

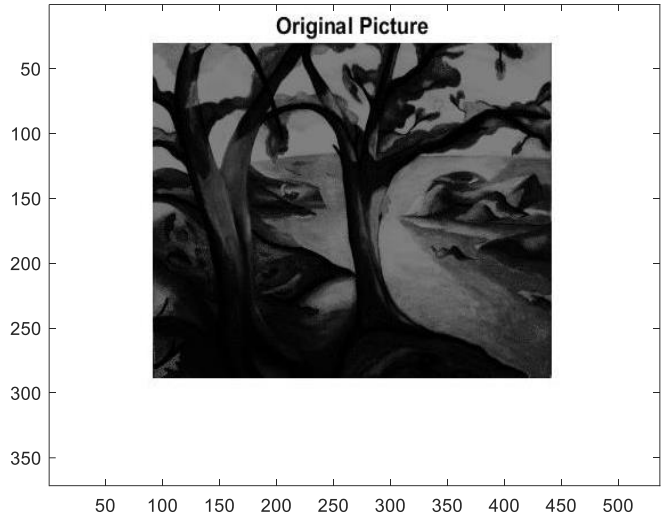

Figure 57

Figure 57 shows the original image that we are going to work with.

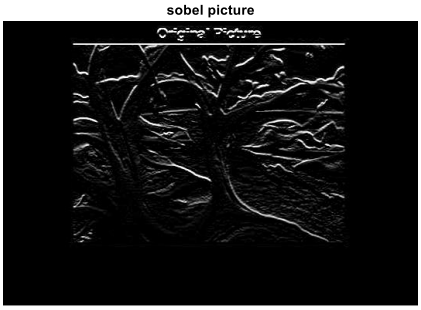

Figure 58

We can see in figure 58 the Sobel filter will detect the edges of the image.

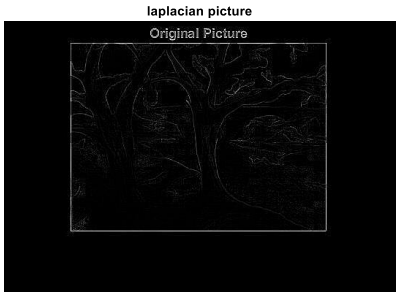

Figure 59

Figure 59 shows the image after applying the laplacian filter witch will detect the edge of the image as well.

Figure 60

Figure 60 shows the image after using the Gaussian filter will try to smoothen the image, as we can see the result is much smoother.

Figure 61

Figure 61 shows the image after using the disk filter as you can see the image will become smoother.

2) In this part, we are going to use the given matrix. The result will look like the following images. The result are in order.

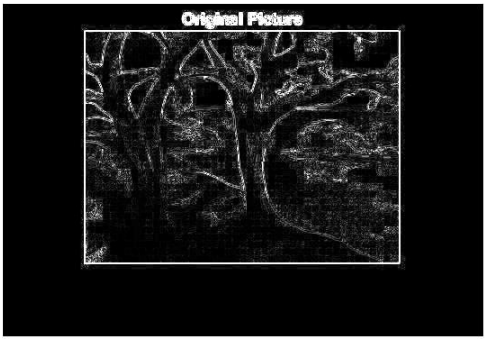

Figure 62

Figure 63

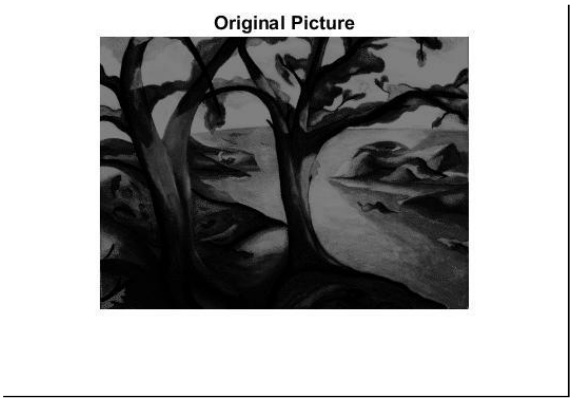

Figure 64

Figure 65

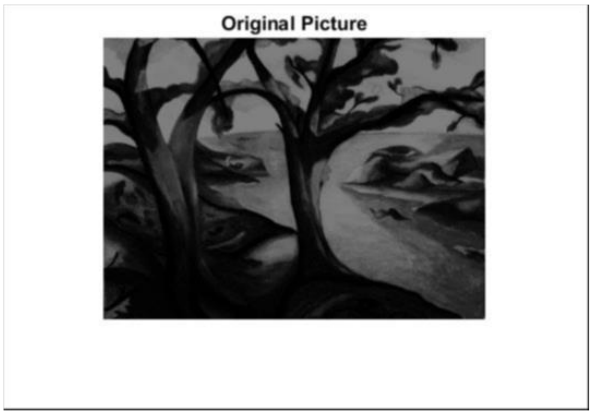

Figure 66

The following images show the result after using Conv2 in Matlab to make the new image.