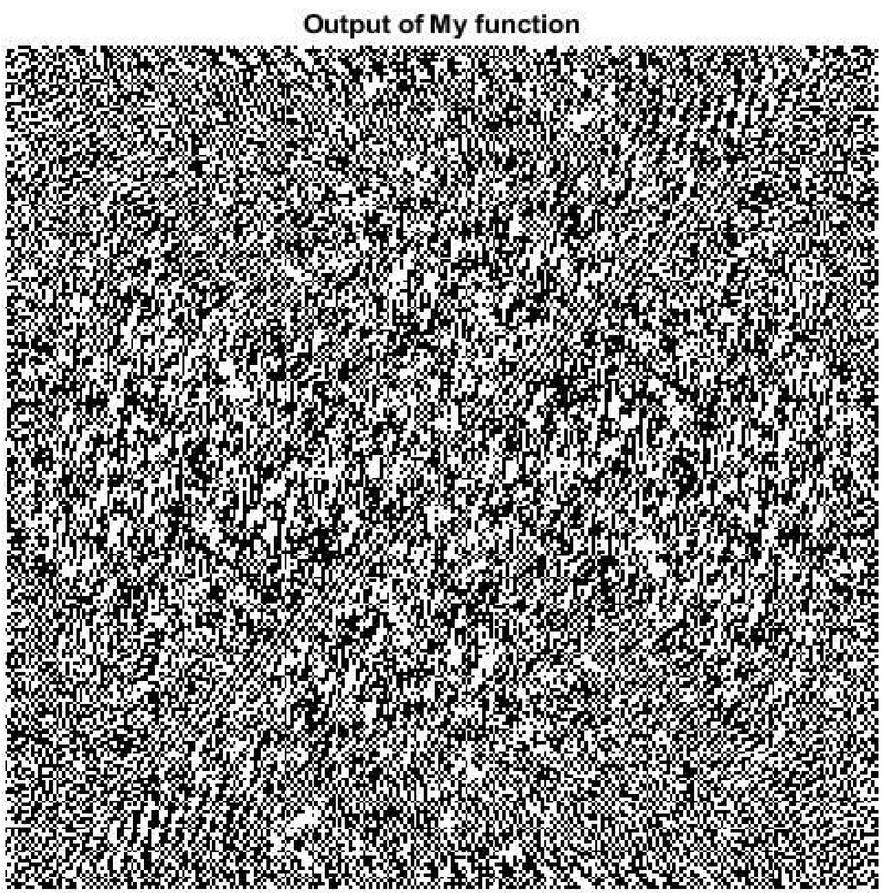
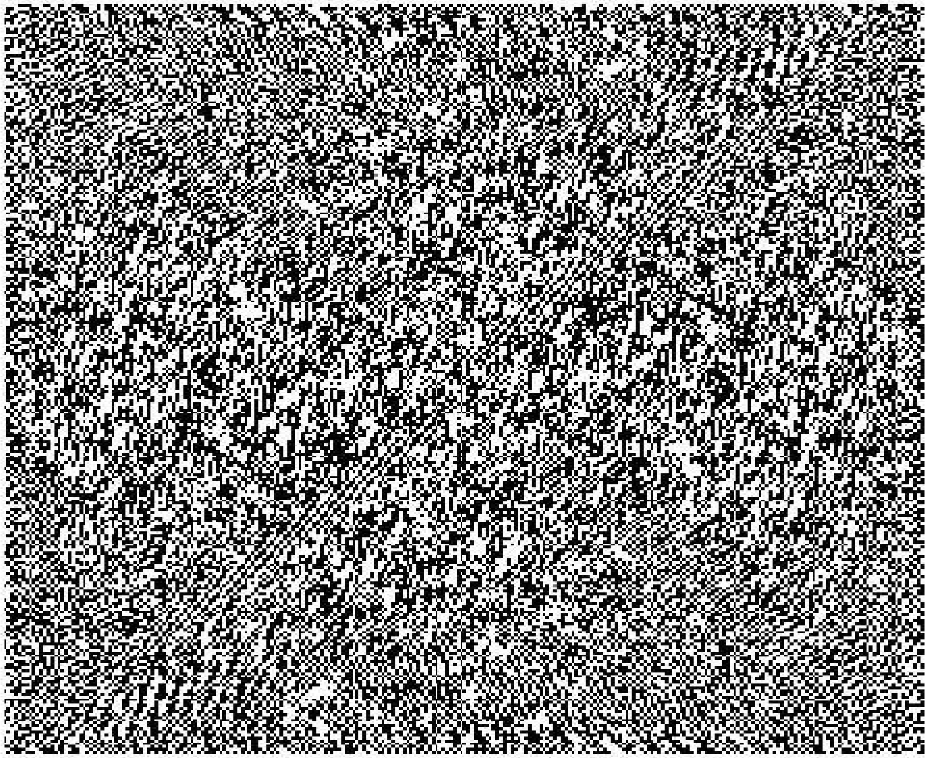
# Sayak Kundu
# Roll No: 20161035
# Assignment 2

## Question 1. 2D Fast Fourier Transform (Recursive Formulation).

1. ***Cameraman.tif:***

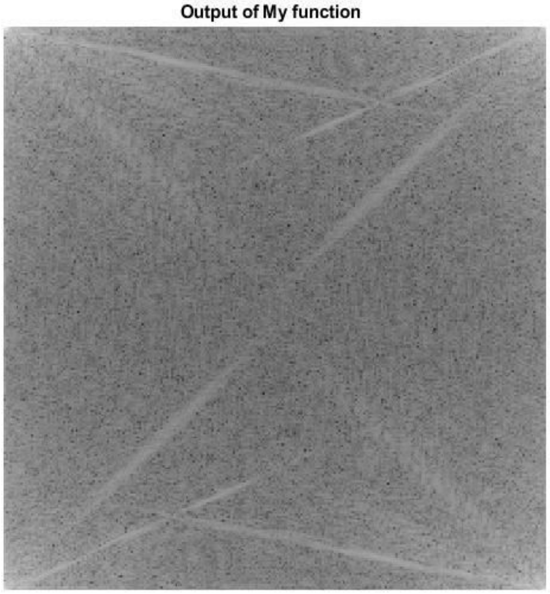


Output of Builtin function

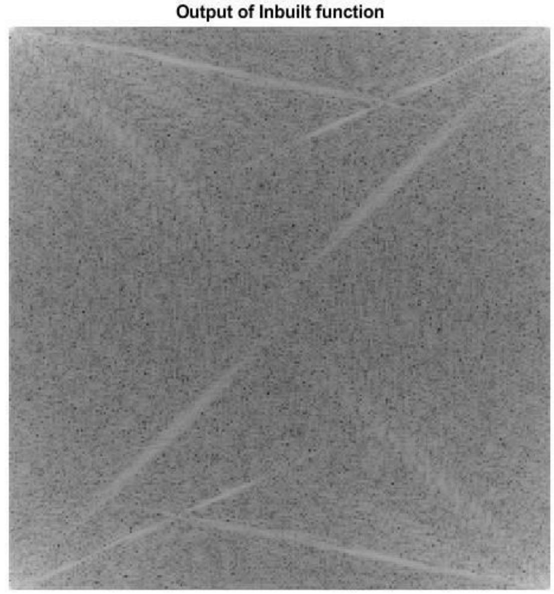


Log Scale of Image

Output of My function

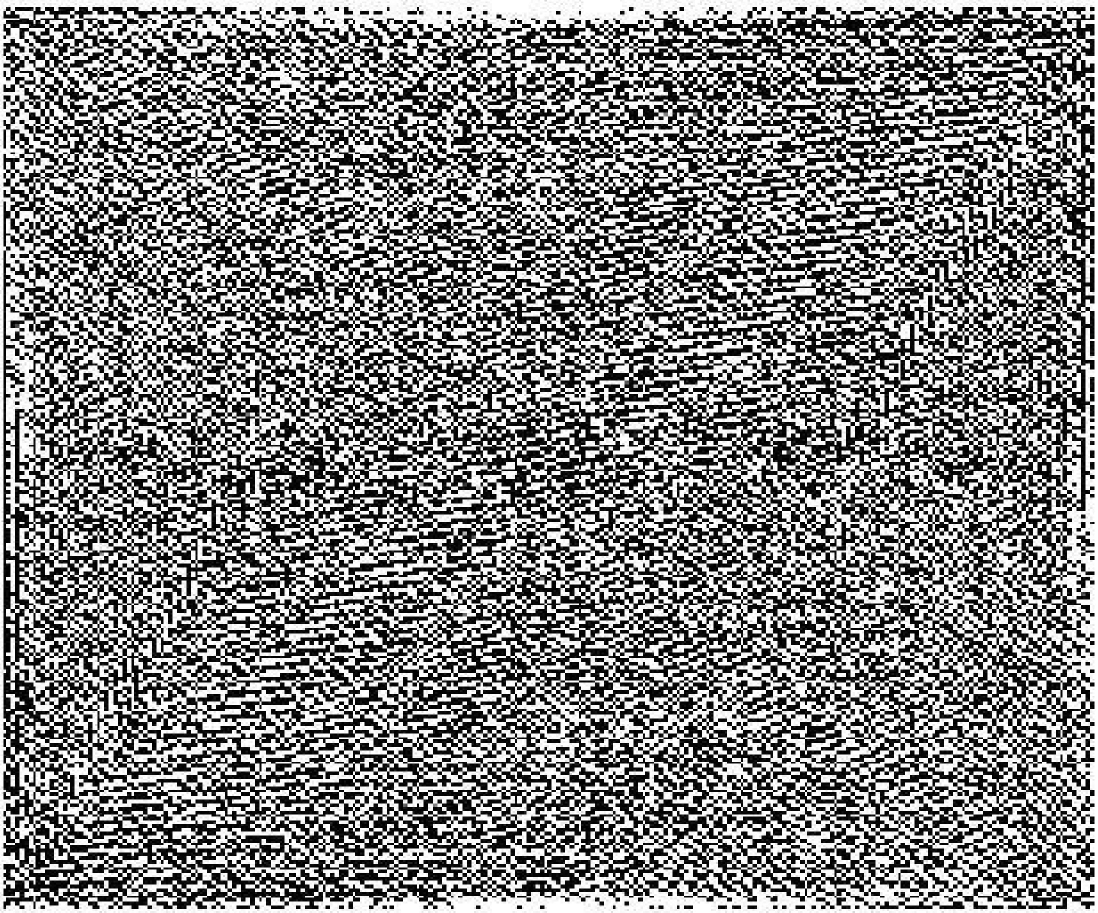


Output of Inbuilt function

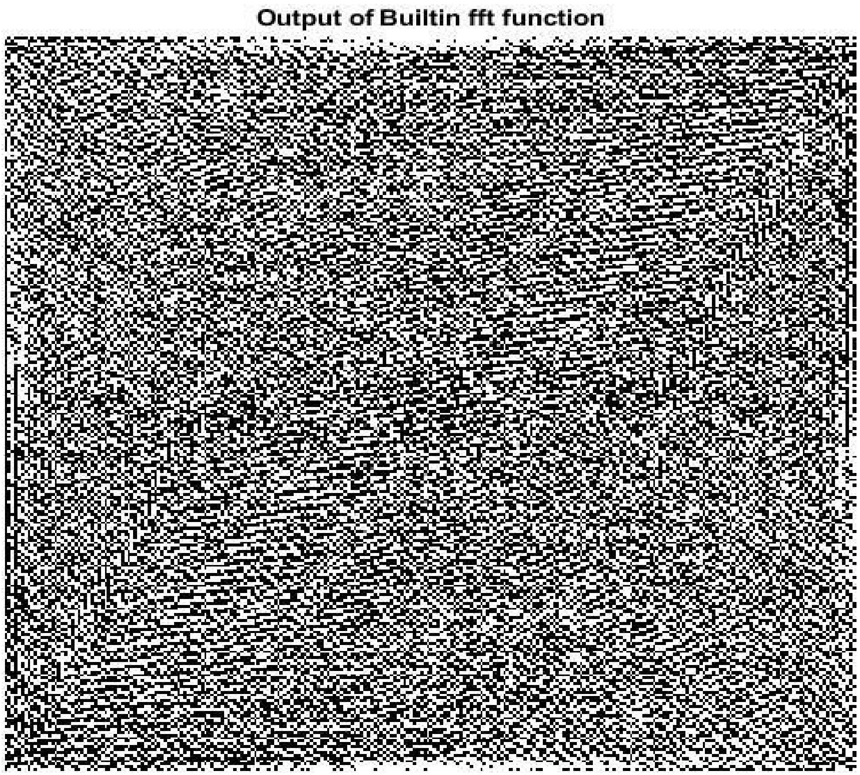


2. ***Peppers.png***

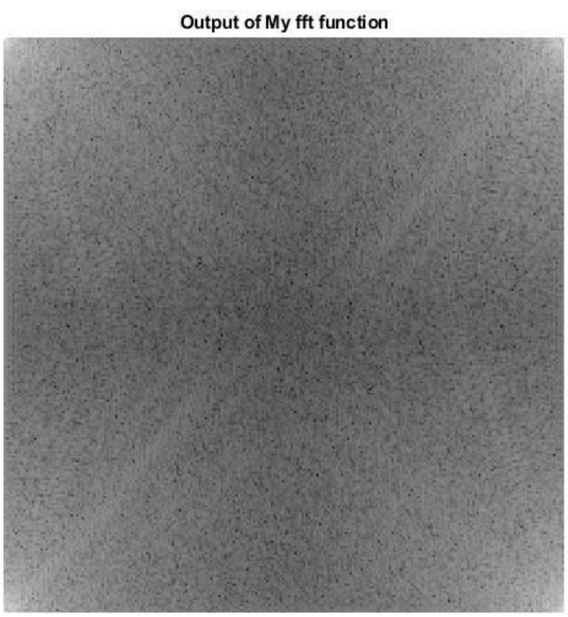
Output of My fft function

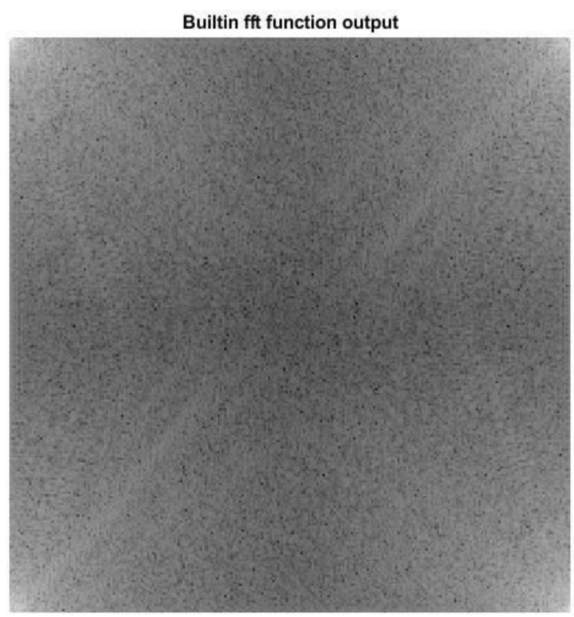


Peppers.png

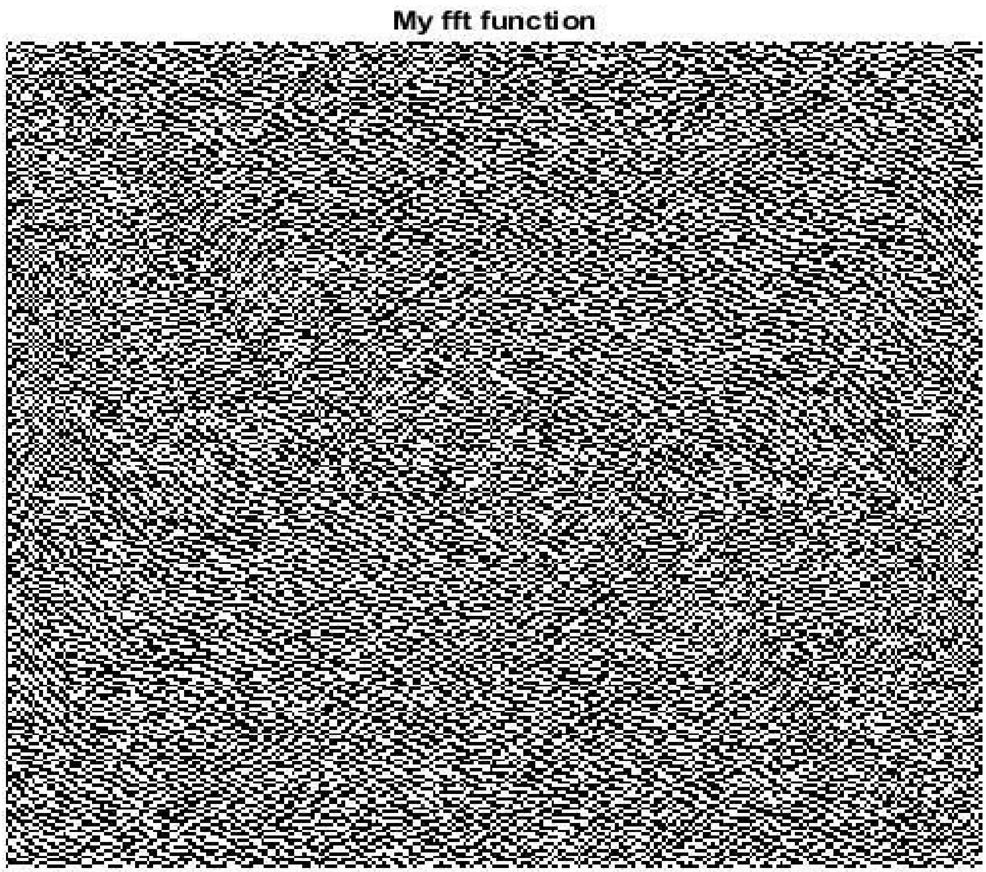



Log Scale Image

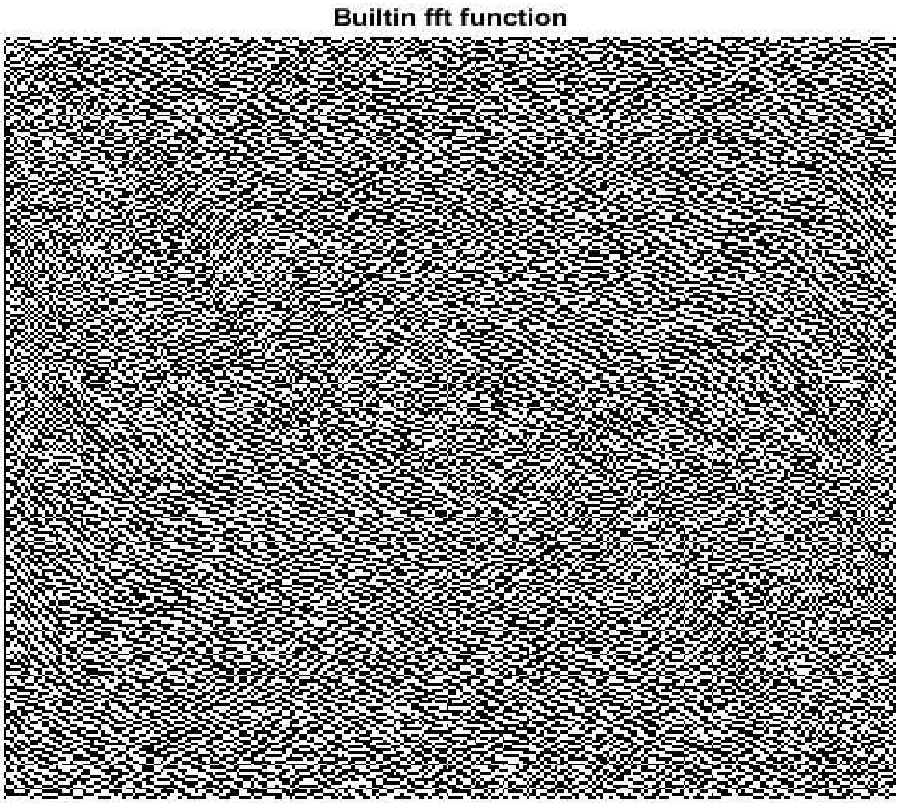


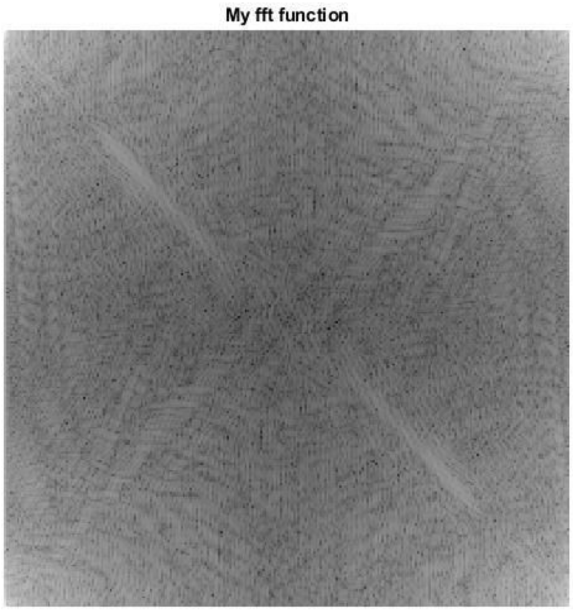
3. ***Car.jpg***

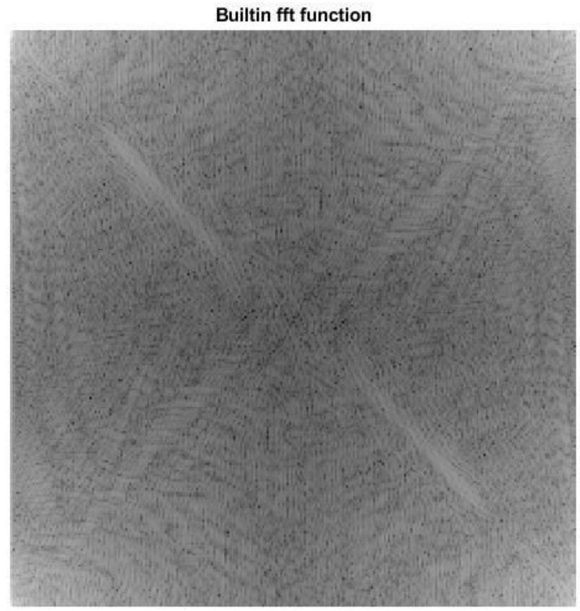


Car.jpg




Log scale of Image



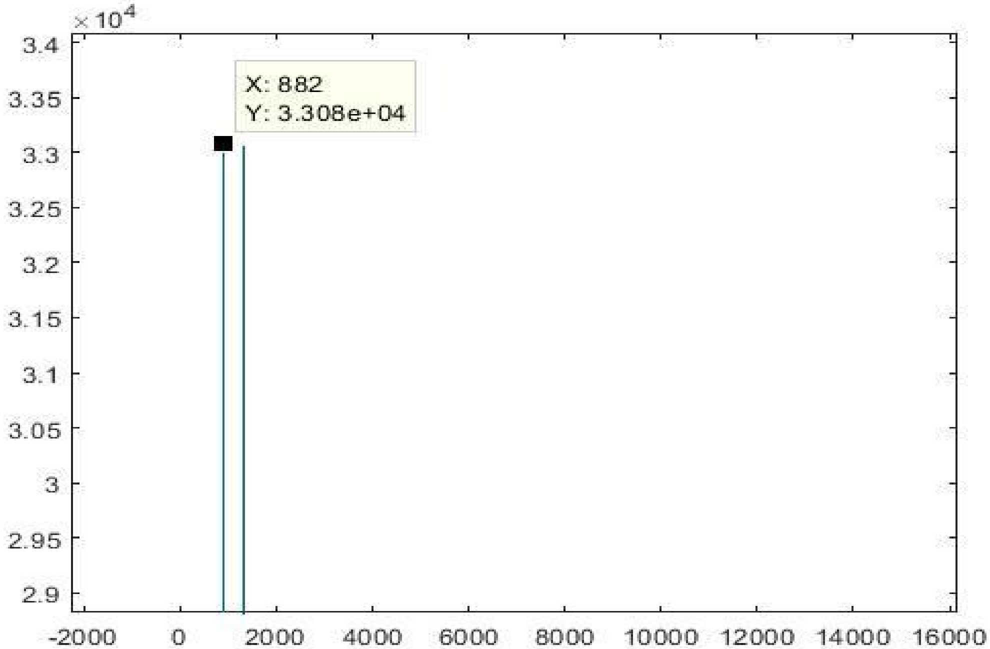



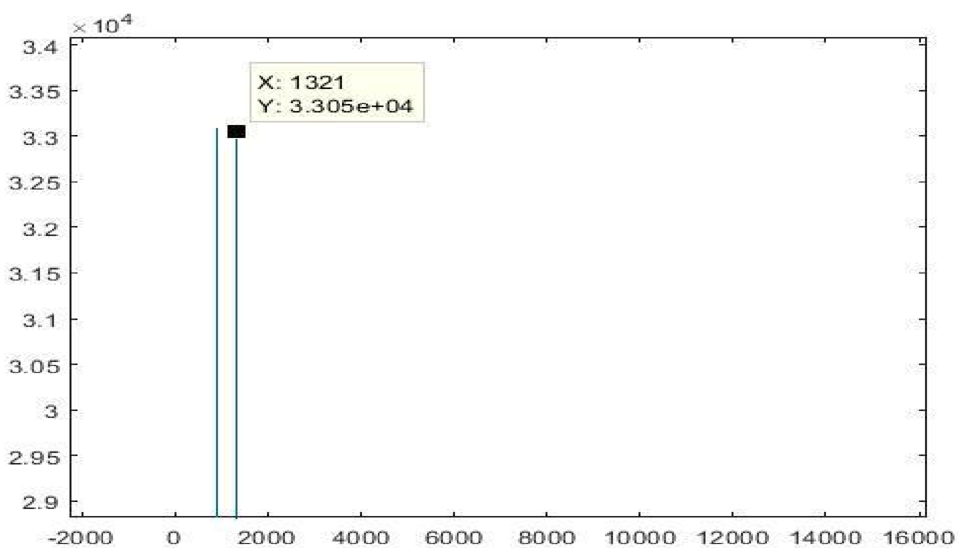

## Question 2. Standard two tone telephones

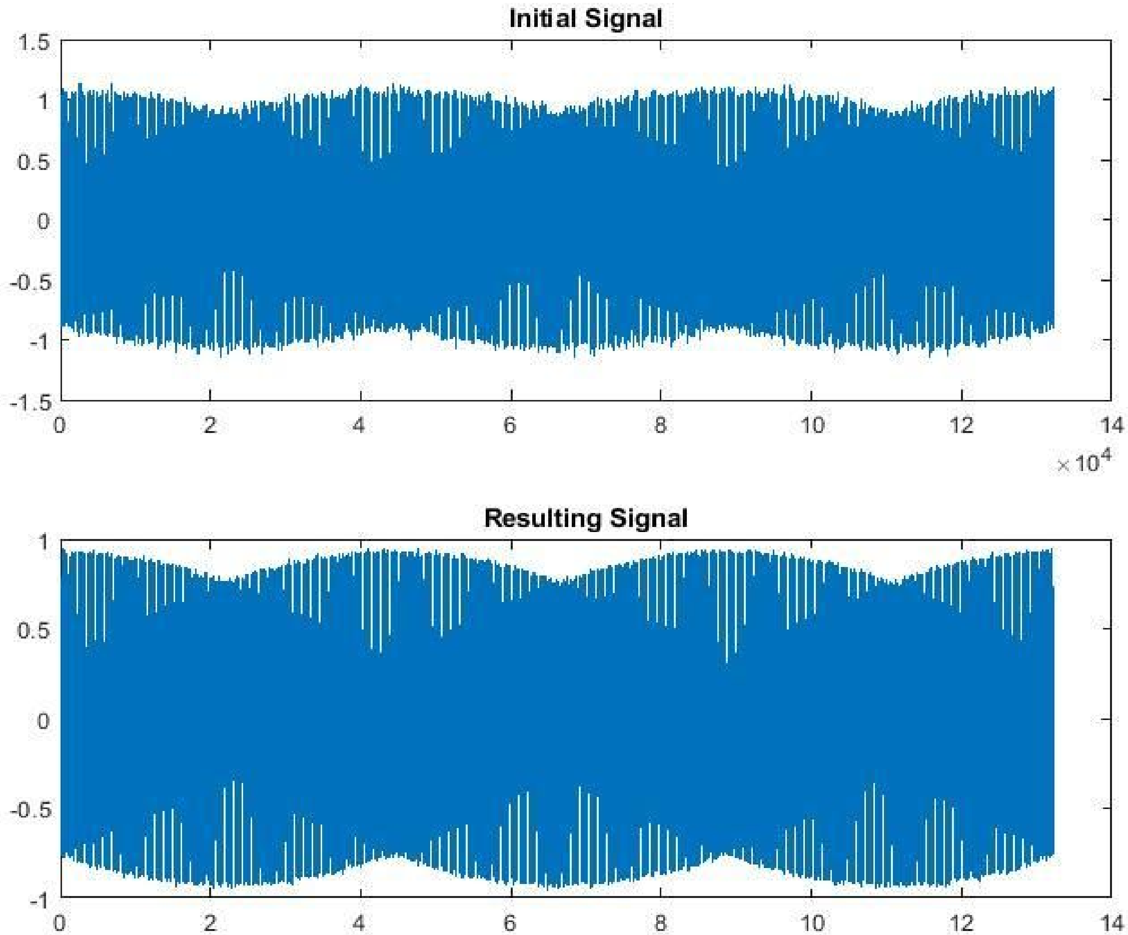
First i load the data using load and store the samples in x and frequency in fs variable.
Then i applied the moving average filter to reduce the intensity of the high frequency noise if it contains.
Then i have applied the Gaussian filter of gaussian window size 24 and Savitzky-Golay smoothing filter to remove the noise. Savitzky-Golay smoothing filter is considered as a good filter for smoothening of audio files.
Then i save the resultant file with name as ‘filtered.wav’ and original signal as ‘original.wav’ using audiowrite function.

At last i plot the signal and corresponding frequency using plot and fft function.

**Two peaks in real part of signal and there is repetition of it.**

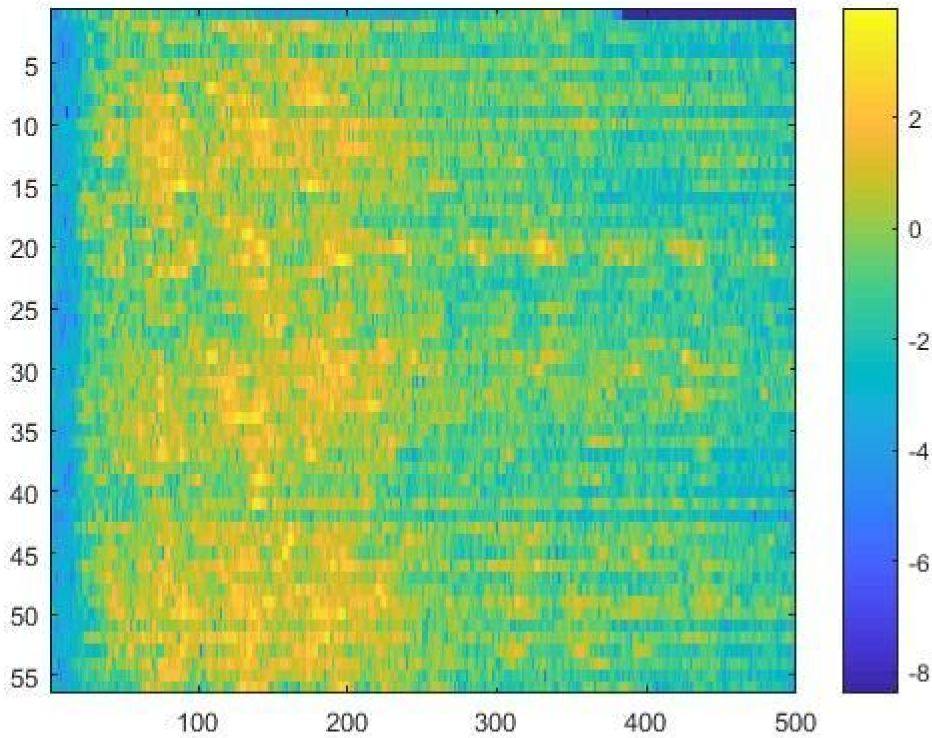

×10^4
X: 882
Y: 3.308e+04
3.4
3.35
3.3
3.25
3.2
3.15
3.1
3.05
3
2.95
2.9
-2000
0
2000
4000
6000
8000
10000
12000
14000
16000


×10^4
X: 1321
Y: 3.305e+04
3.4
3.35
3.3
3.25
3.2
3.15
3.1
3.05
3
2.95
2.9
-2000
0
2000
4000
6000
8000
10000
12000
14000
16000

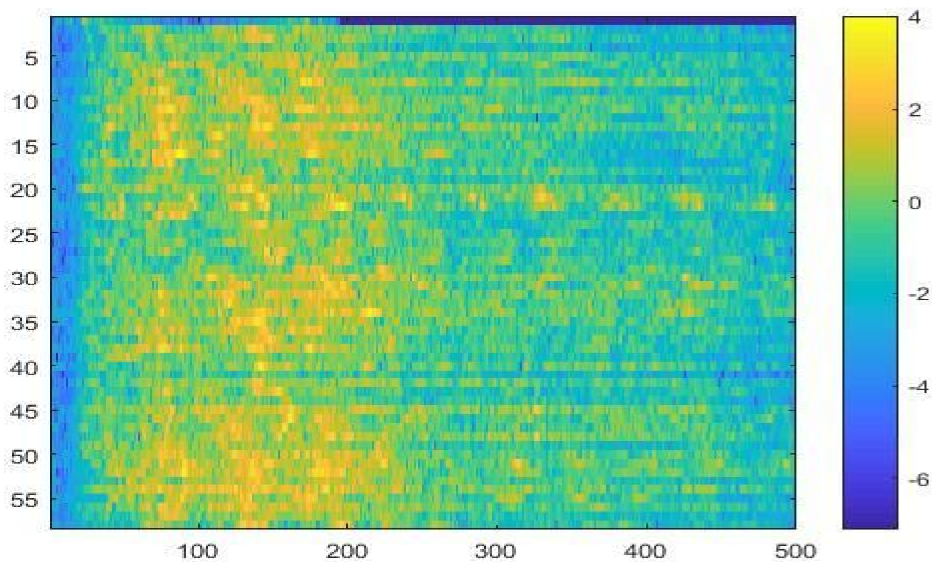



## Question 3. Compute the spectrogram of a given audio file

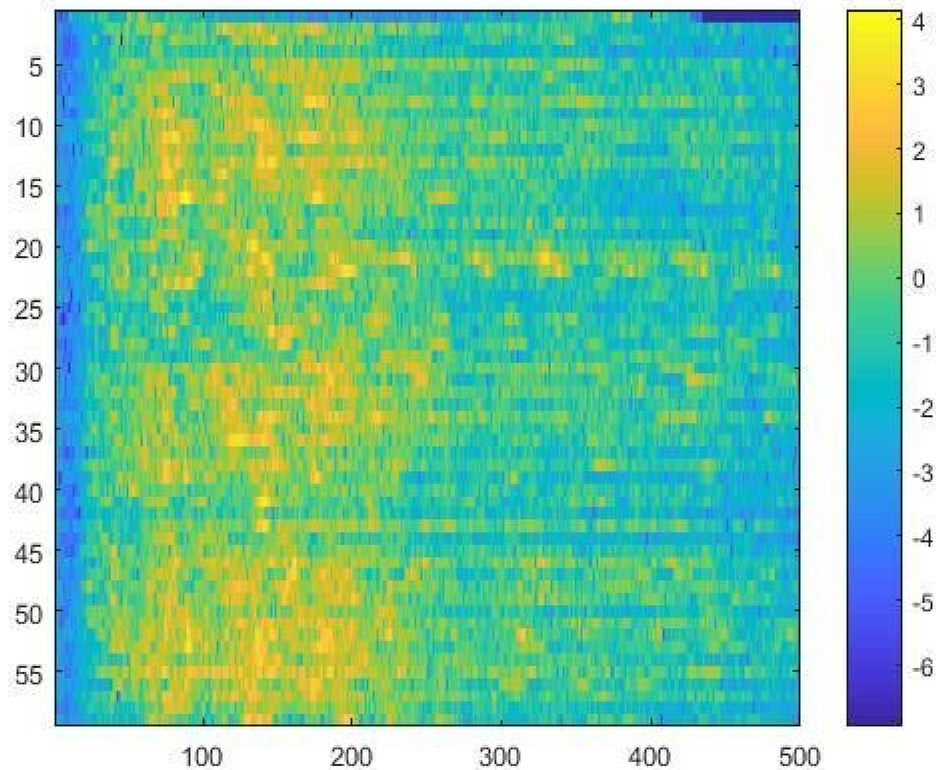
Window length : 1000, Stride length : 50




Window length : 1000, Stride length : 80

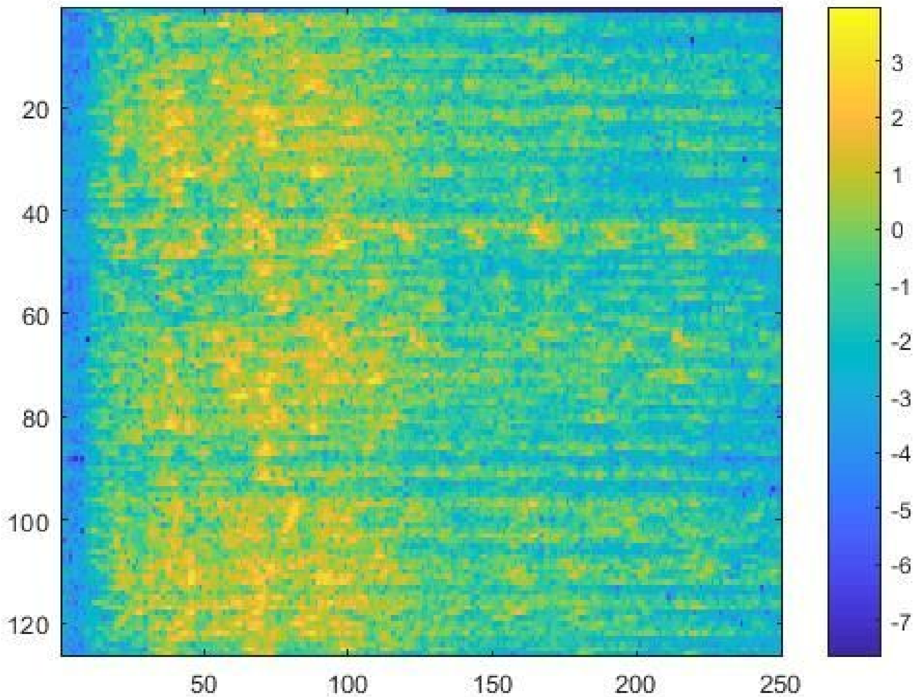



Window length : 1000, Stride length : 100

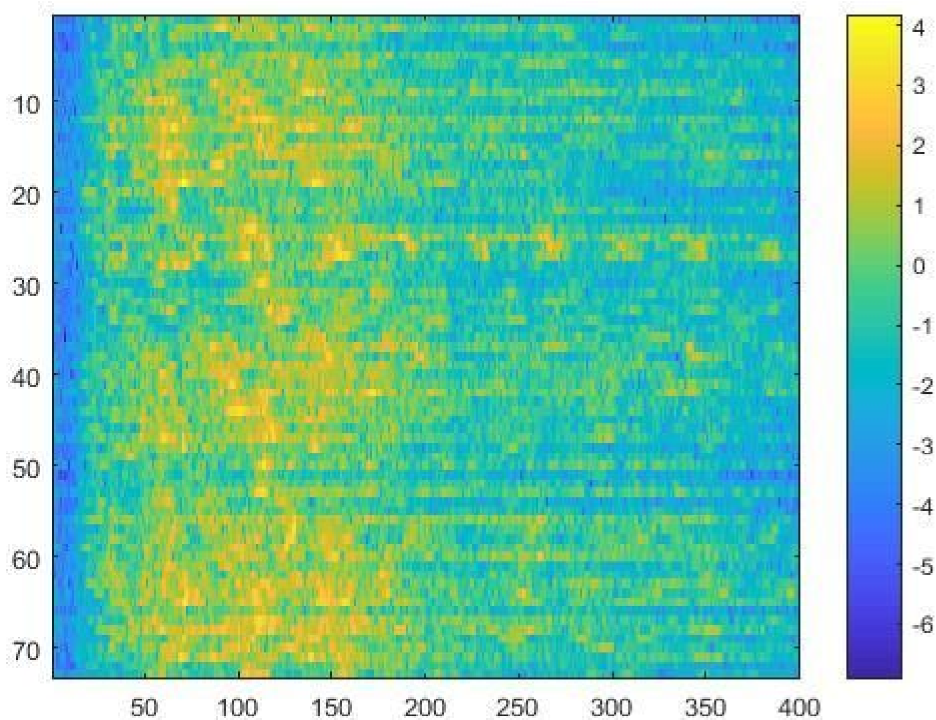



**Increasing the stride length causes loss in data captured i.e there is some frequency miss in spectrogram.Resolution of spectrogram also decreases. This is due to frequency-time representation.If we have varying frequencies then we have to keep time fix for that moment same as of heisenberg uncertainty principle.**

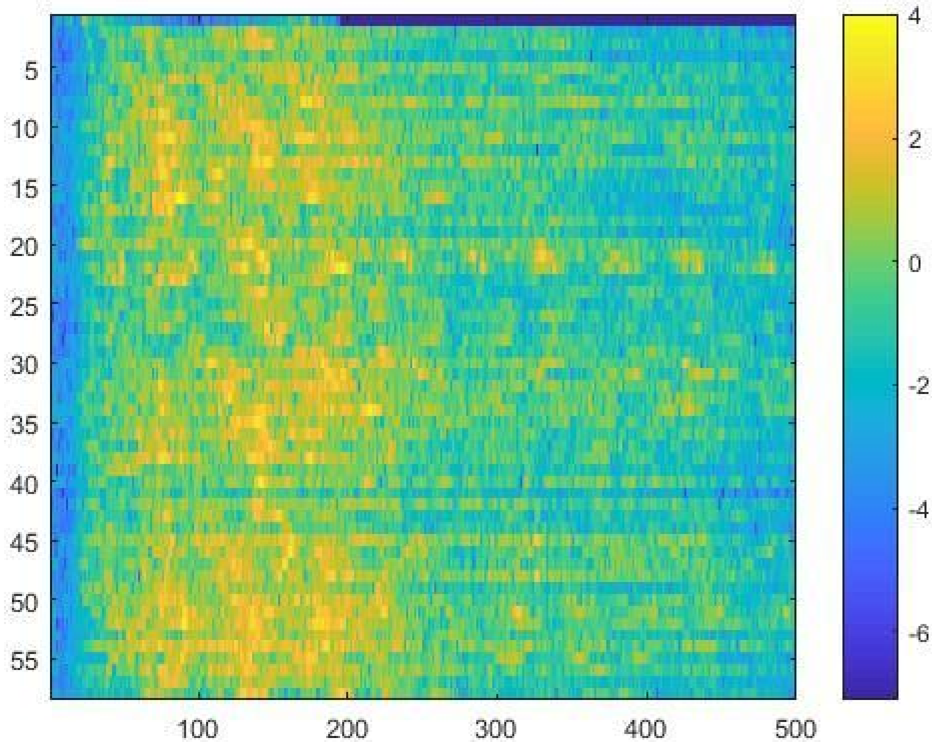
Window length : 500, Stride length : 80




Window length : 800, Stride length : 80




Window length : 1000, Stride length : 80

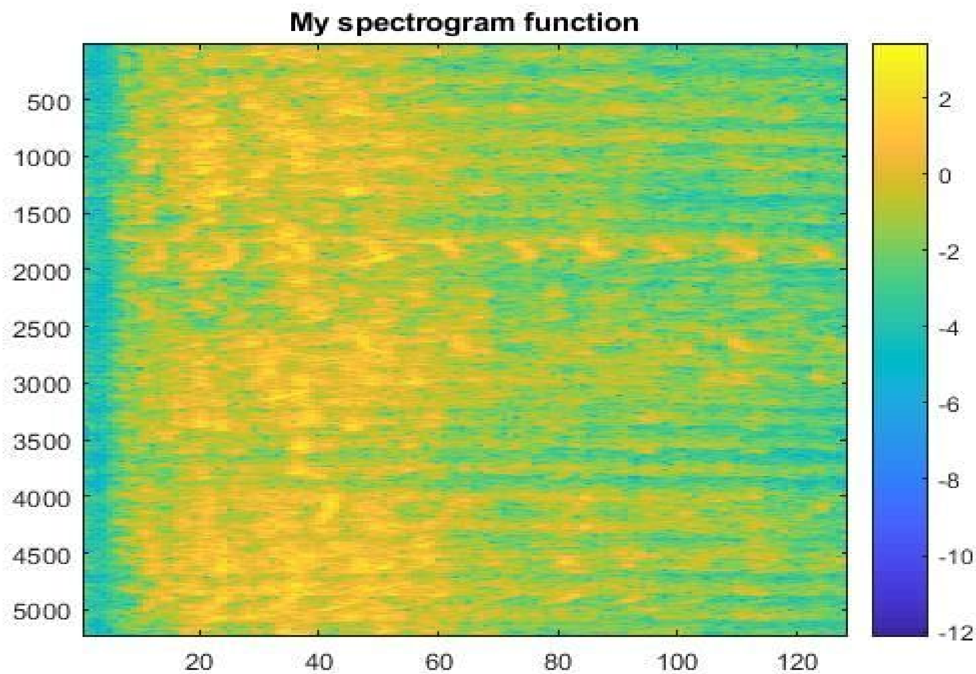


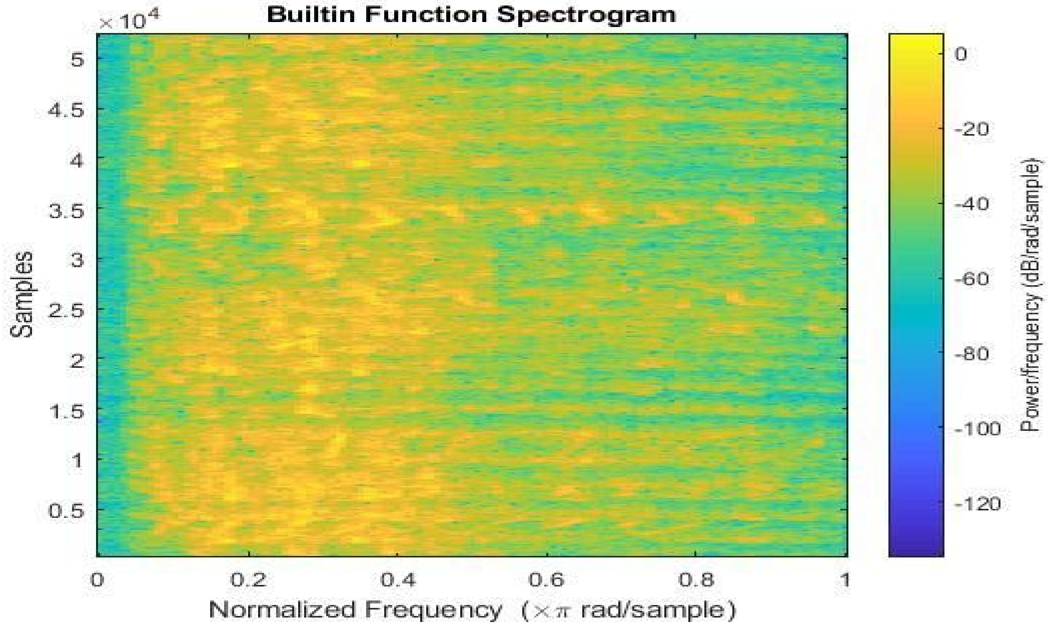

**Increasing the window length causes increase(enhancement) in data captured i.e there is more accurate plotting of frequencies in spectrogram. Resolution of spectrogram also increases. This is due to increase in overlapping of window with the given signal at a time.Also the gap between two window also decrease which result in better resolution.**

## Comparison between my result and inbuilt spectrogram

Window length : 256, Stride length : 10

**File name : laughter**

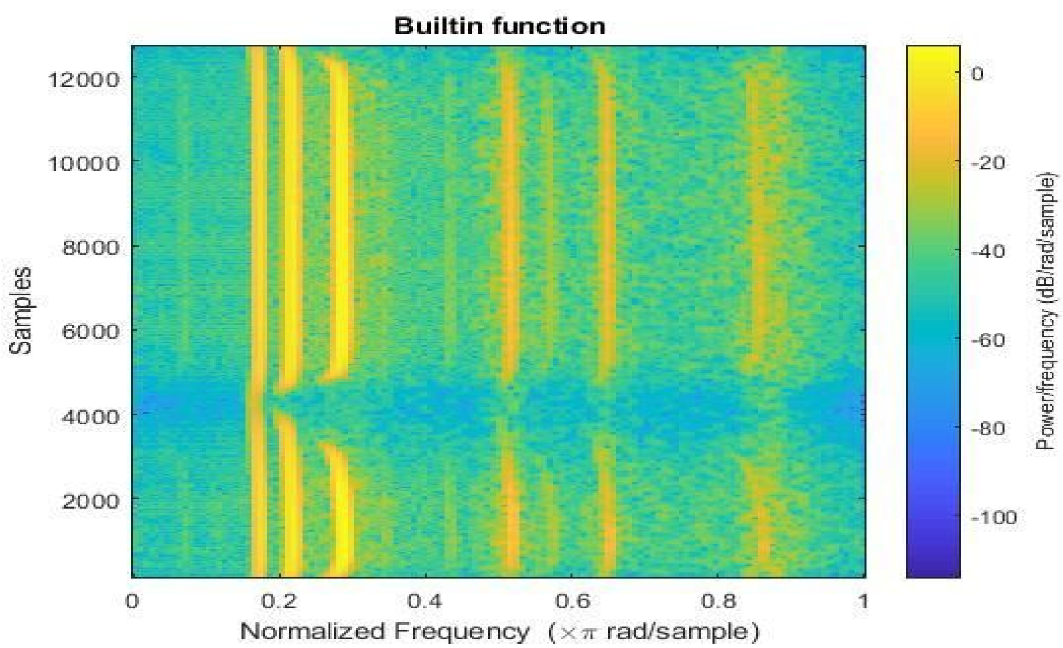


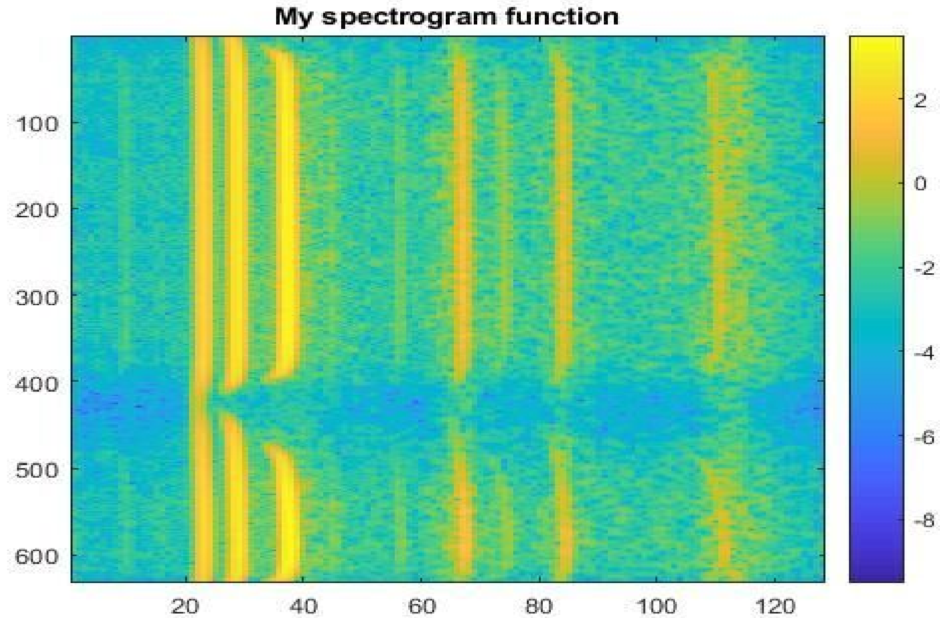



**Comparison between my result and inbuilt spectrogram**

Window length : 256, Stride length : 20

**File name : train**

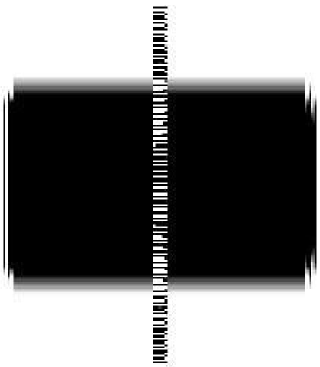


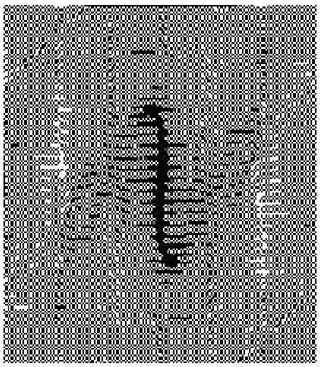


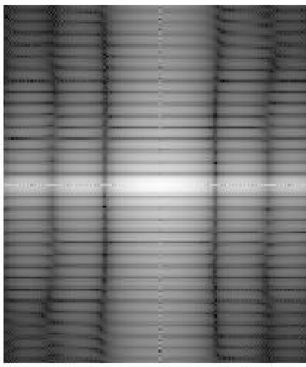

## Question 4. Observations on doing FFT

1. Img1a.png

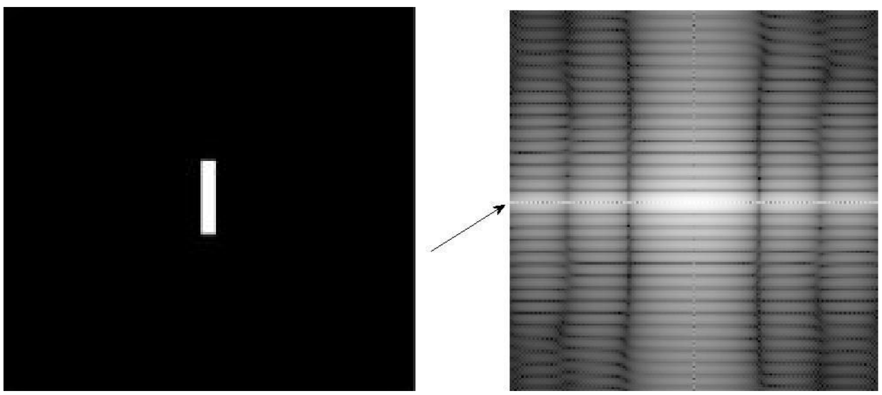
| FFT | Log scale log(abs(signal+1)) | FFT2 | Log Scale FFT2 |
|---|---|---|---|





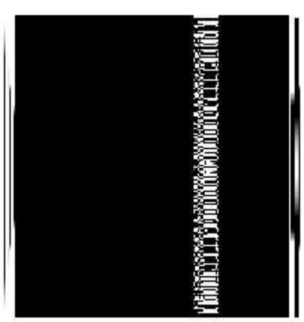
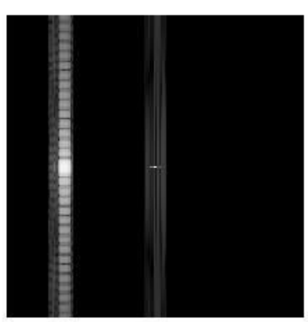
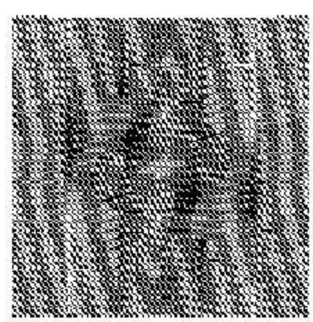
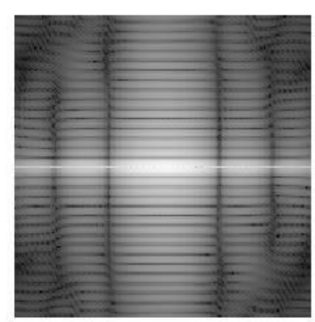


***The resulting image has a very particular pattern, with a lot of harmonic frequencies.We can  see that the rectangle appeared to be rotated 90 degrees. That is incorrect, what we are seeing is that big features become small and small features become big. As such the smaller dimension of the rectangle became larger and the larger dimension became smaller.***

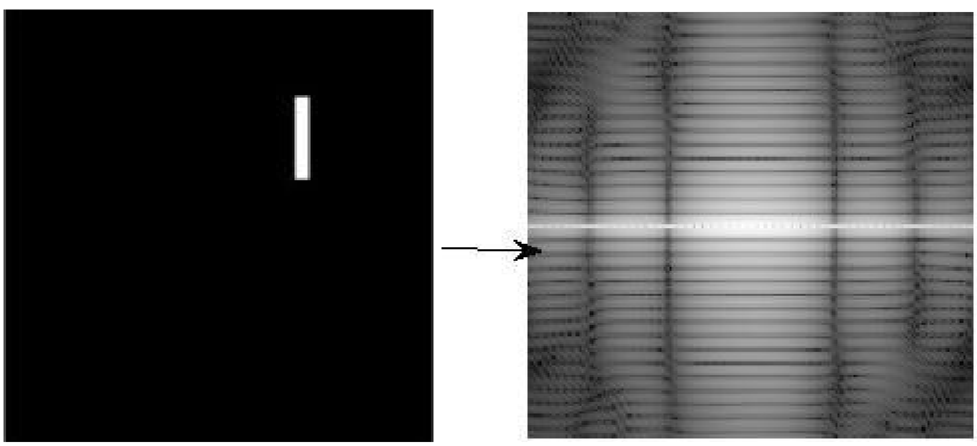


2. Img1b.png

| FFT | Log scale<br>log(abs(signal+1)) | FFT2 | Log Scale FFT2 |
|---|---|---|---|

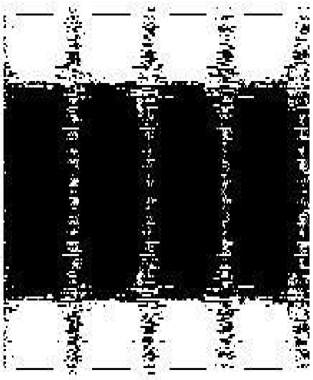
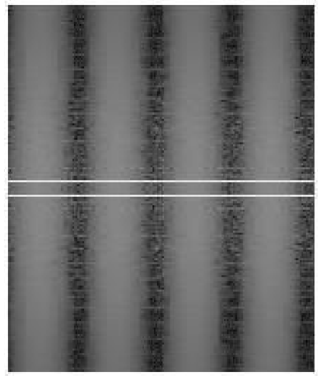
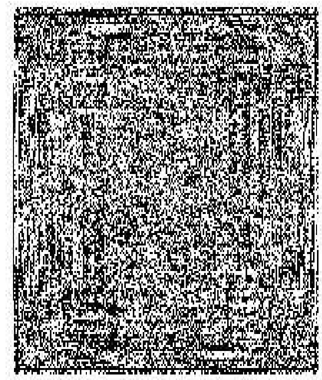
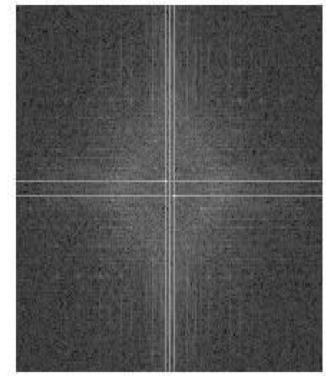





The frequency pattern did not move. That is because all the positioning information is contained in the phase image. The frequency pattern (magnitude or its spectrum does not change because it moved). This position separation, is one of the key features of the Fourier Transform that makes it so very important. It will allow you to search for specific image pattern within a larger image, regardless of the location of the object that produced that fourier spectrum pattern.



3. Img2a.png

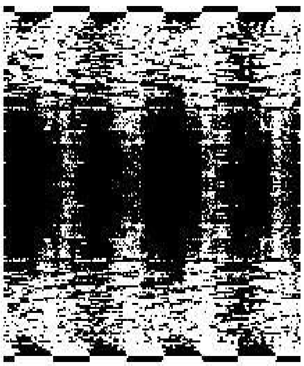
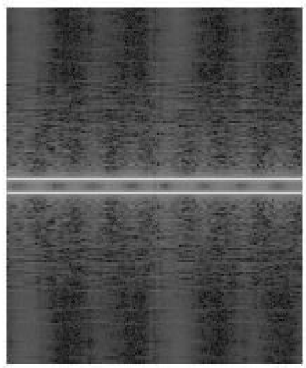
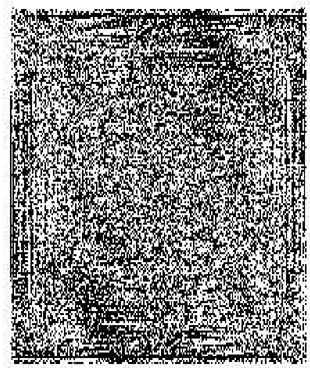
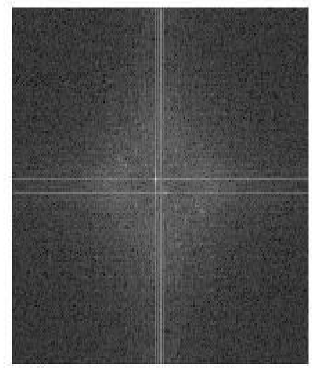
| FFT | Log scale log(abs(signal+1)) | FFT2 | Log Scale FFT2 |
|---|---|---|---|

**The given image is tilted sinusoidal with 8-cycles .In the spectrum image it has three dots.The center dot is as before the average DC value. The other two dots represent the perfect sine wave that the Fourier Operator found in the image. As the frequency across the width of the image is exactly 8 cycles, and as a result two frequency pixels are exactly 8 pixels away from the center DC value.**

4. **Img2a.png**

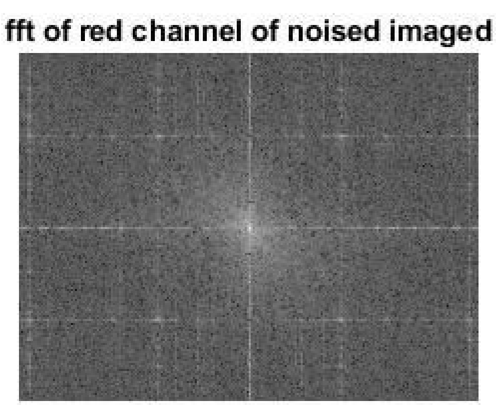
| FFT | Log scale log(abs(signal+1)) | FFT2 | Log Scale FFT2 |
|---|---|---|---|

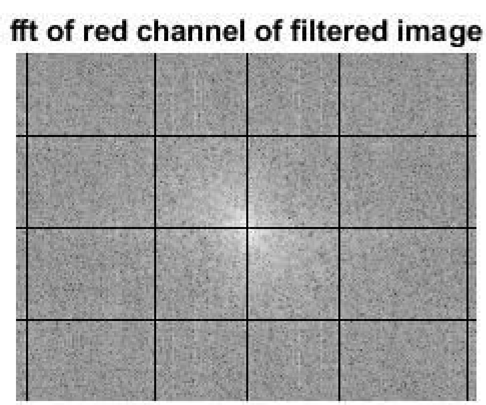





**The given image is tilted sinusoidal having some pixel value close to white or black .In the spectrum image it has five dots.The center dot is as before the average DC value. The other four dots represent the sine wave that the Fourier Operator found in the image. As the frequency across the width of the image is exactly 4 cycles, and as a result two frequency pixels are exactly 4 pixels away from the center DC value and other two are 8 distance**

away.

**Using rec filter :**



Filtered Image

**Question 5.**

In this question i have taken fft of all audio signals of digit 0 to 9 and then iterated through the input signal taking the window of size = that of the digit’s audio, thereafter i have compared the window with digits by taking the dot product of two and determining the digit press by taking maximum of all .
This is because in dot product we are multiplying each sample of the window with the corresponding sample of the digit’s audio and taking the sum.This way, we make sure that the digit which was pressed in that window can be accurately. We are taking the window size = that of digit’s audio because it is the time interval for which the particular key is pressed.

**Question 6**

**MESSAGE1: IF YOU ARE GOOD AT SOMETHING NEVER DO IT FOR FREE.**
**MESSAGE2: WHY SO SERIOUS**
**MESSAGE3: LET'S PUT A SMILE ON THAT FACE.**

In this question i have taken sub parts of fft of given message signal by dividing the fft signal into four parts. Then i have generated all the possible permutation of sub parts which is 24 and constructed the signal back by taking inverse fourier transformation.
I have stored all the corresponding signal in files numbered from 1 to 24 and provide saving dialog box so that you can save your message in any directory.