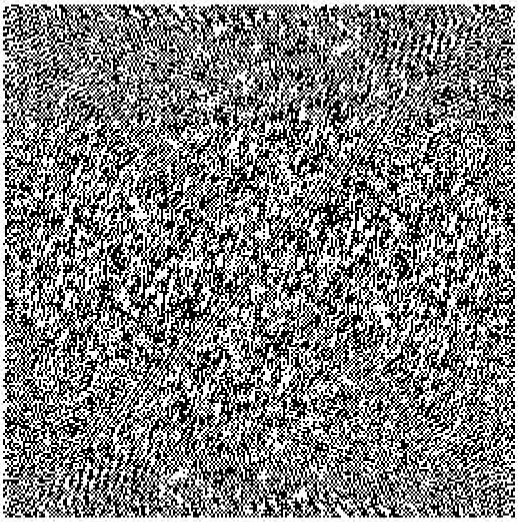
ASSIGNMENT 2

NAME - AKSHAT MAHESHWARI
ROLL NO. - 20161024

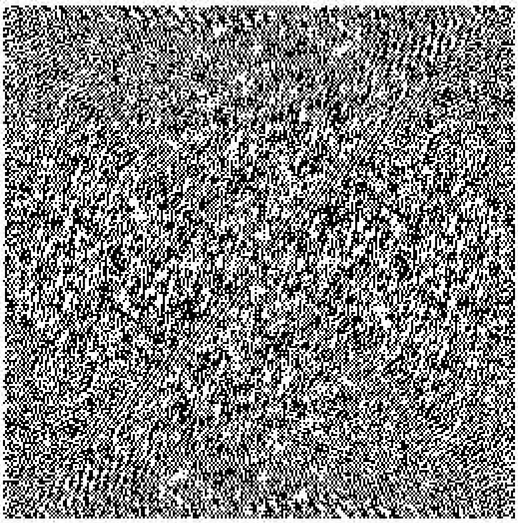
QUESTION - 1: Recursive 2D fft

1. Cameraman.tif
- Original Image

- My function for fft2

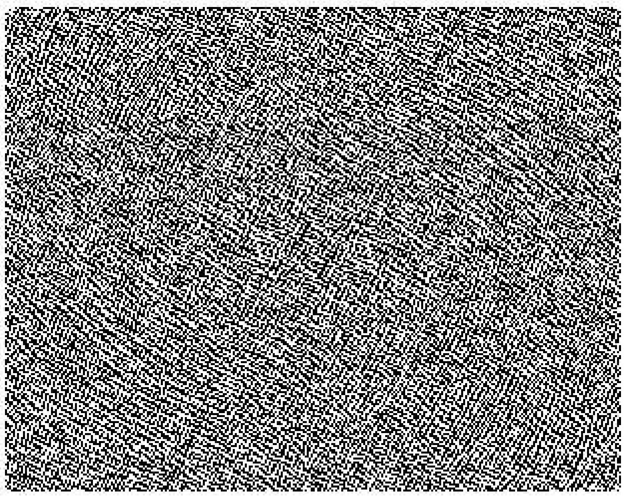


- Inbuilt function fft2

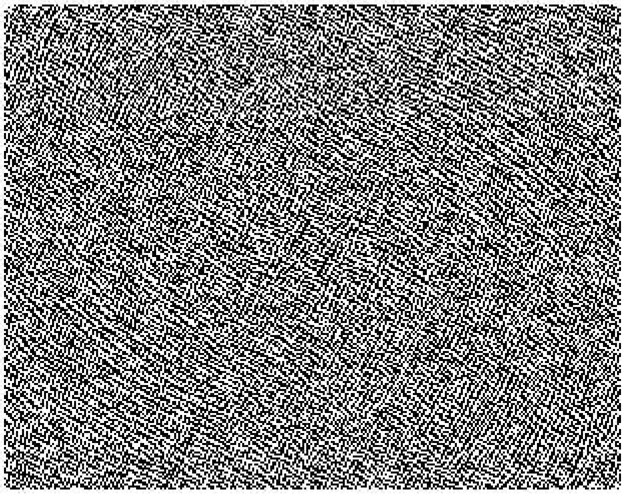


2. Coins.tif
   - Original Image

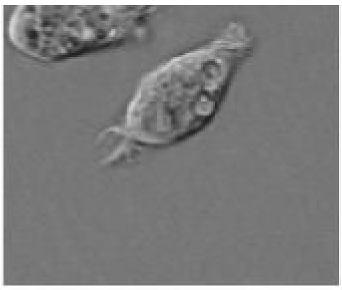
- My function for fft2



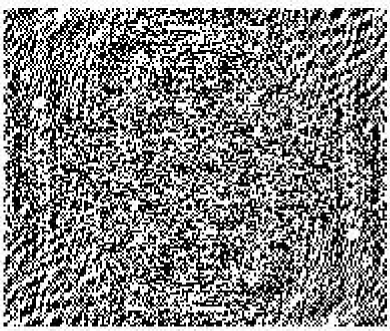
- Inbuilt function fft2



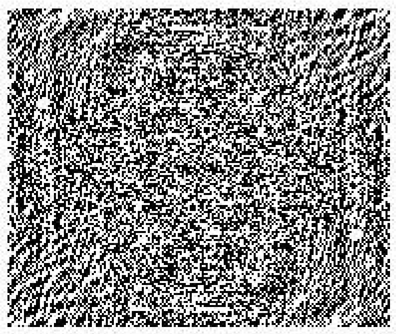
3. Cell.tif
   - Original Image



   - My function for fft2



- Inbuilt function fft2



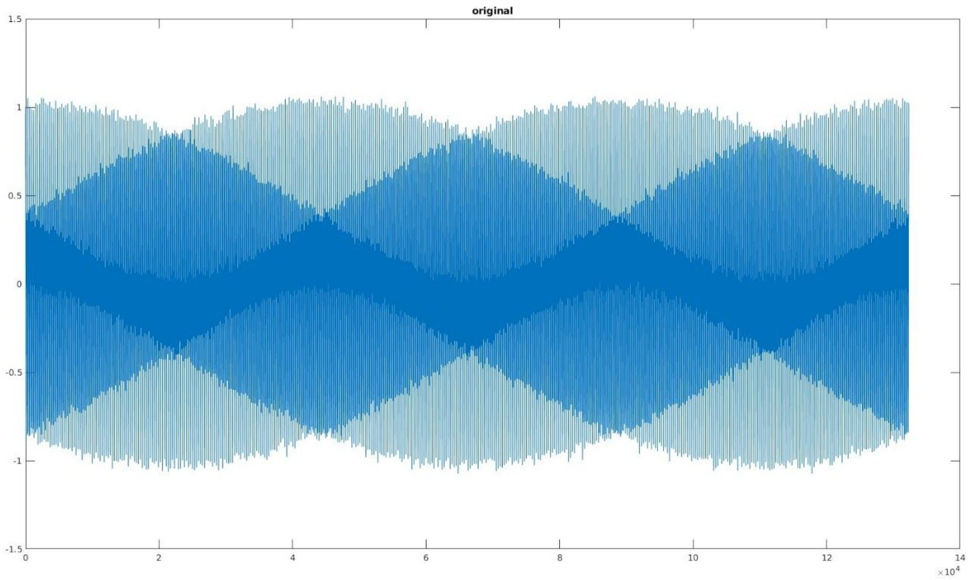
QUESTION - 2: Standard 2 Telephones

In this question, we are given audio signal containing noise. To remove the noise from the signal, fourier transform can be used. By obtaining the signal in the frequency domain, we can identify the noise signal present in certain range and zero it out. Then inverse fourier is taken to get back noise free signal.

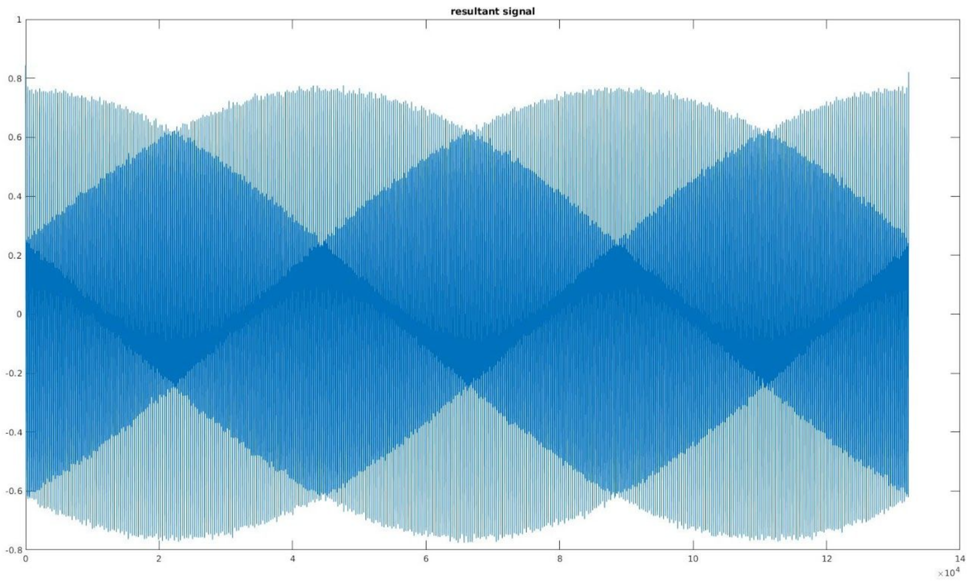
I have used another alternative method by using convolution along with averaging. At first I have taken the fourier transform. I have taken the average of the signal by considering value of previous unit , value of current unit, and value of next unit. Hamming window of size 40 is used to generate 40 random number and are normalized. Signal averaged above is convoluted with the normalized gaussian window and Savitzky-Golay filter is applied for smoothening the signal (I got it on the internet). The resulting signal is the noise free signal.

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

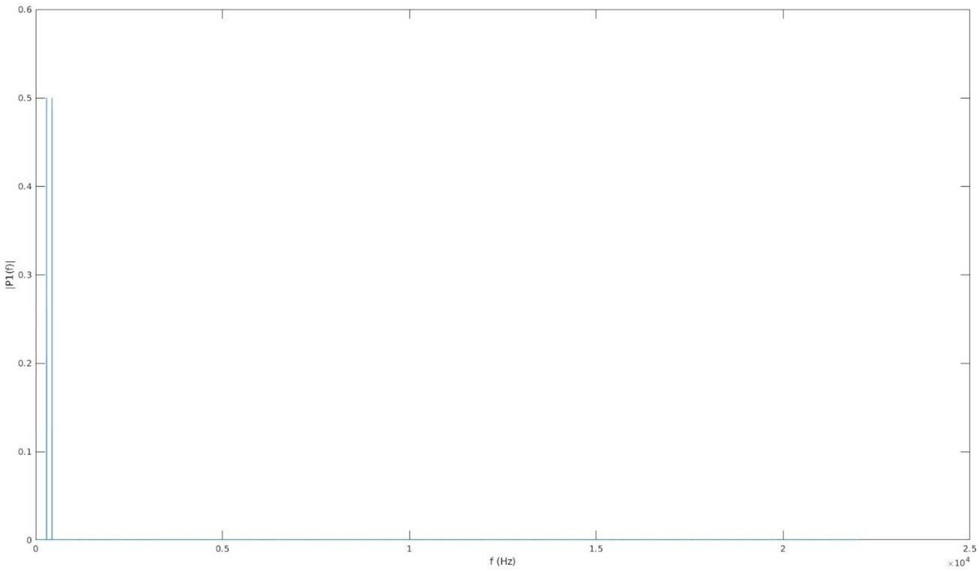
- Original Signal



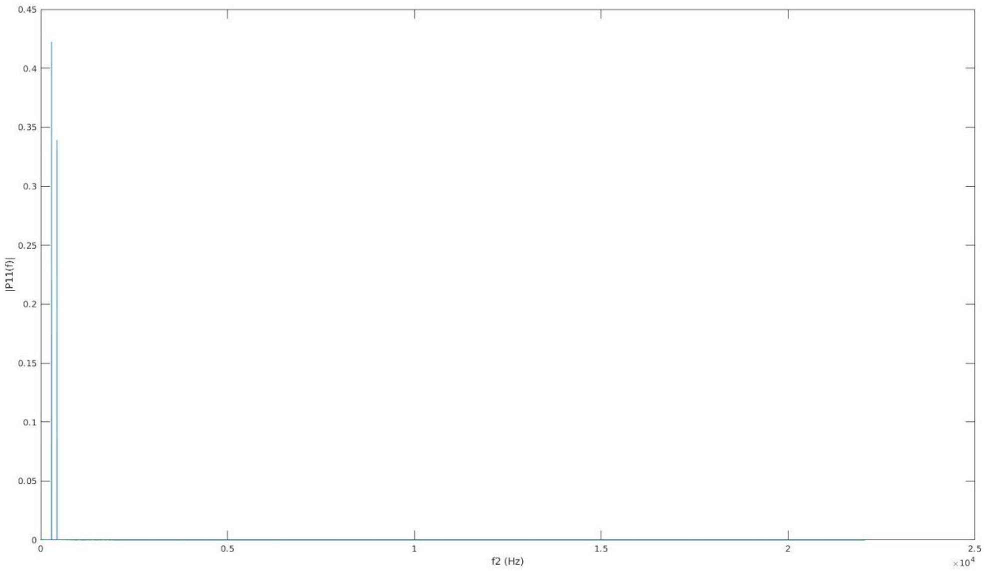
- Resultant Signal



- Original Frequency Plot



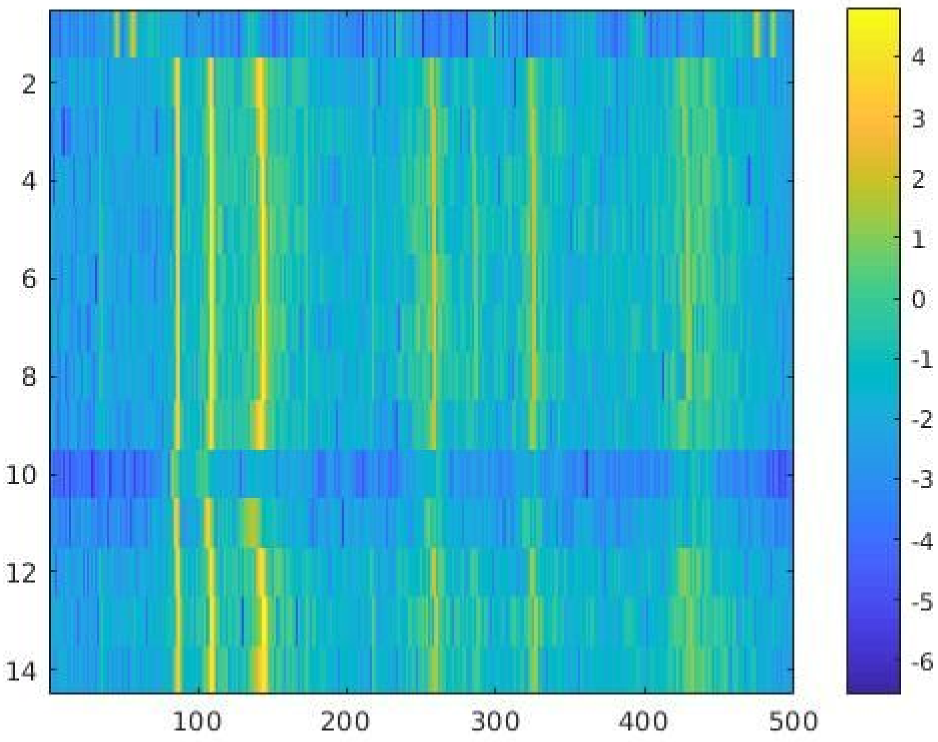
- Resultant Frequency Plot



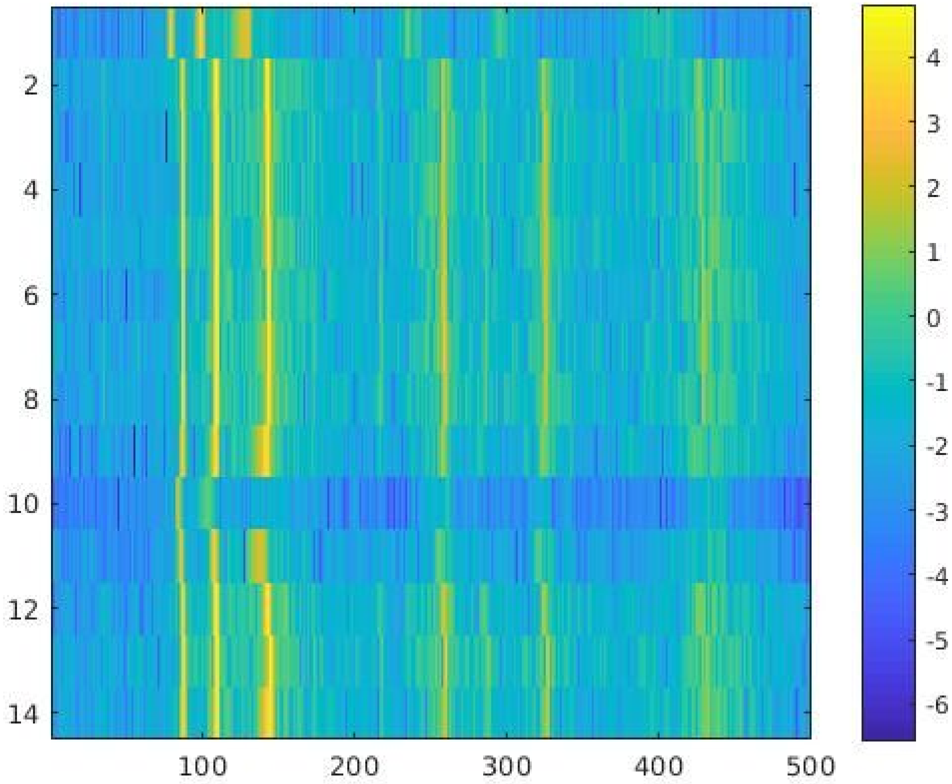
QUESTION - 3: Spectrogram of Audio File

Used Sound: TRAIN

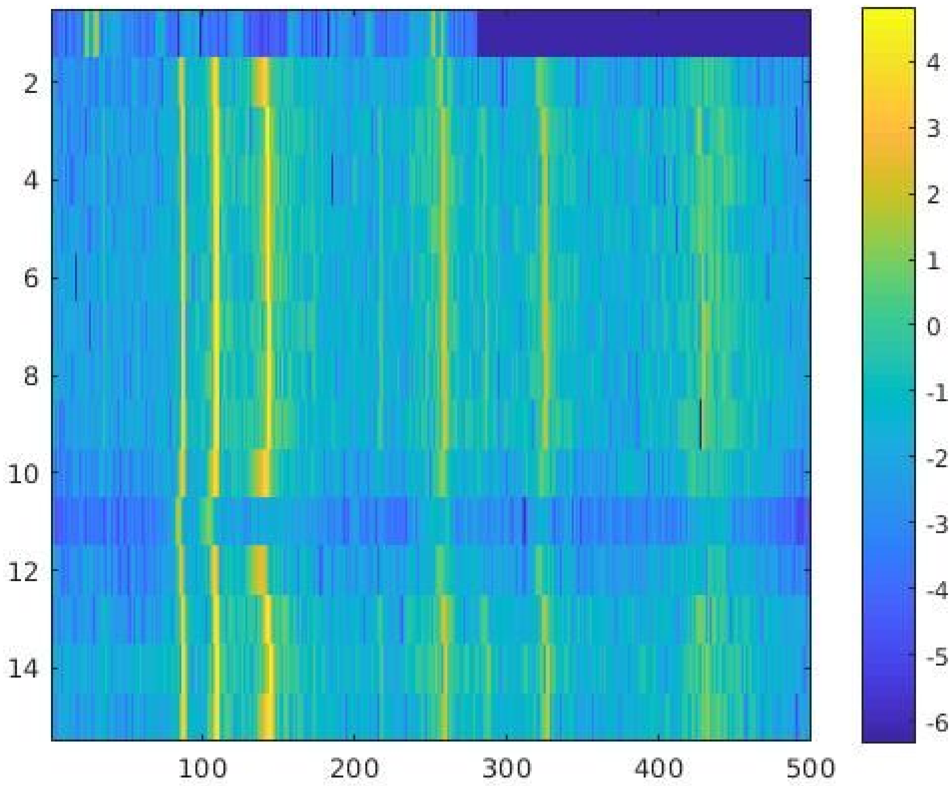
- WindowLen:1000, StrideLen:50



- WindowLen:1000, StrideLen:80

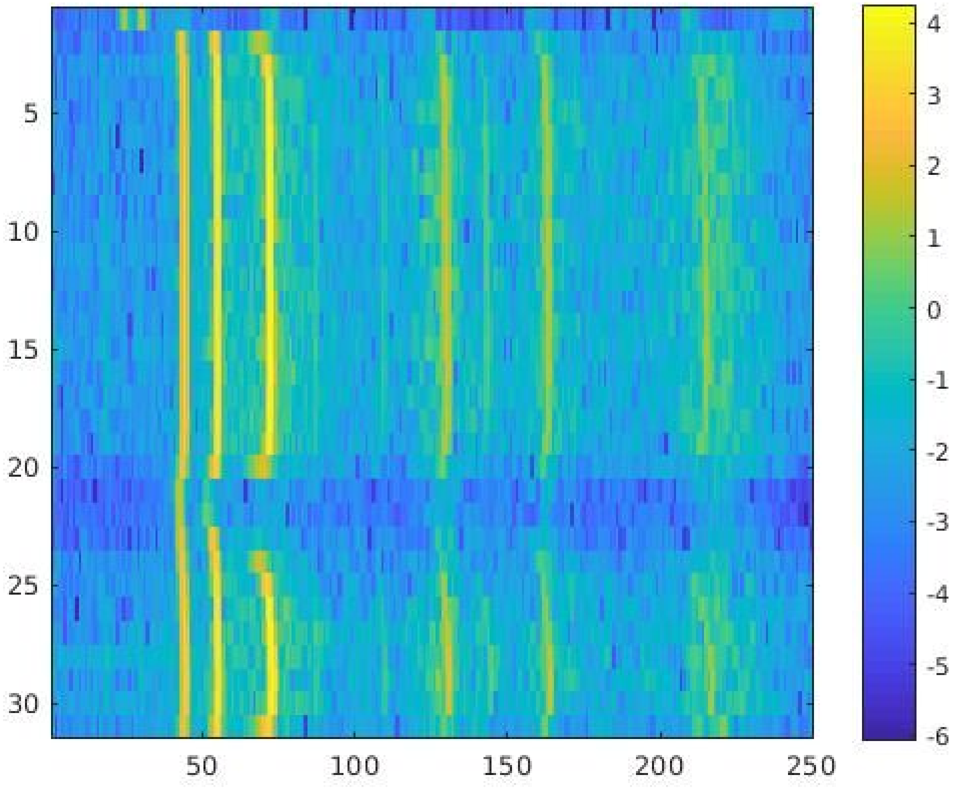


- WindowLen:1000, StrideLen:100

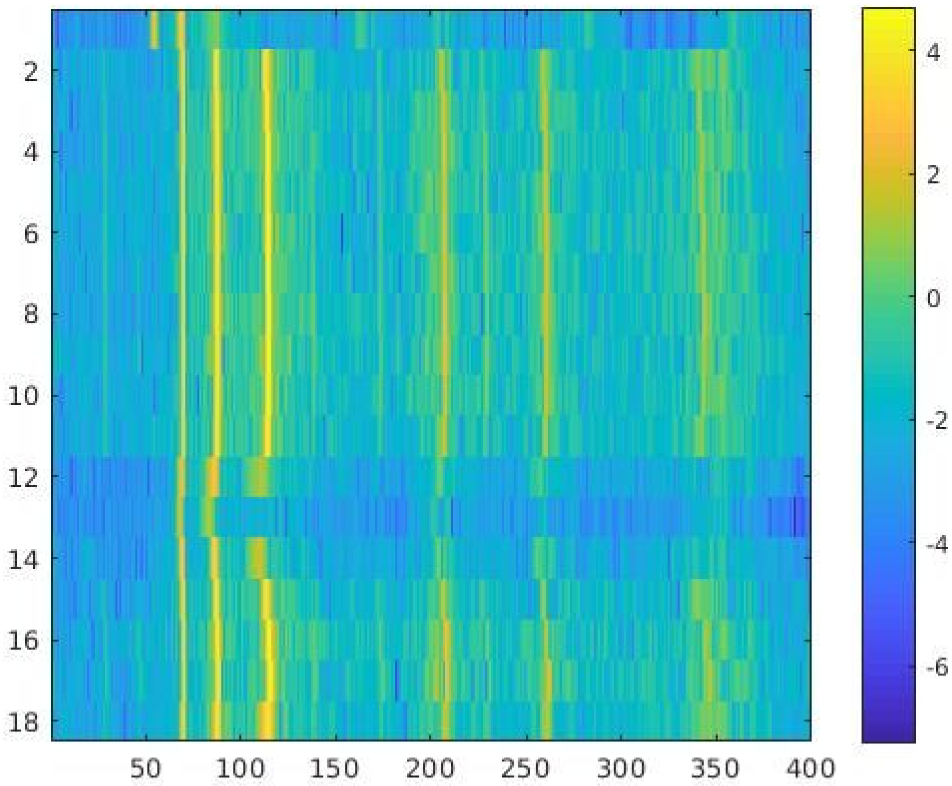


Increasing the stride length causes loss in data captured i.e there is some frequency miss in spectrogram. Resolution of spectrogram also decreases. This is due to frequency-time representation. If we have varying frequencies then we have to keep time fix for that moment same as of heisenberg uncertainty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

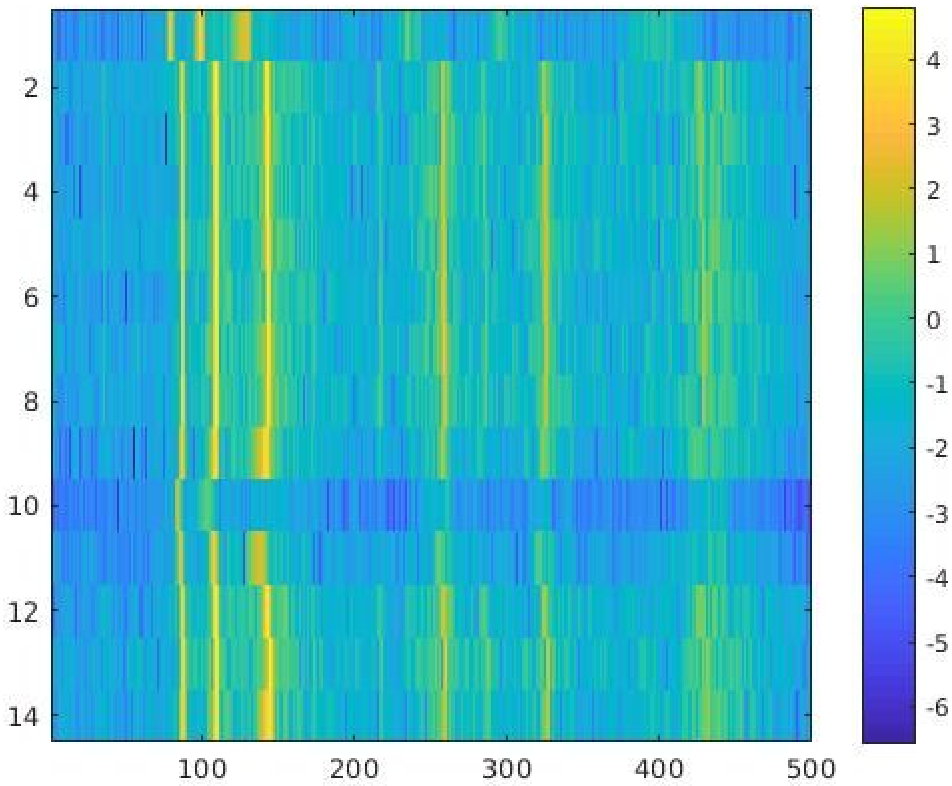
- WindowLen:500, StrideLen:80



- WindowLen:800, StrideLen:80



- WindowLen:1000, StrideLen:80

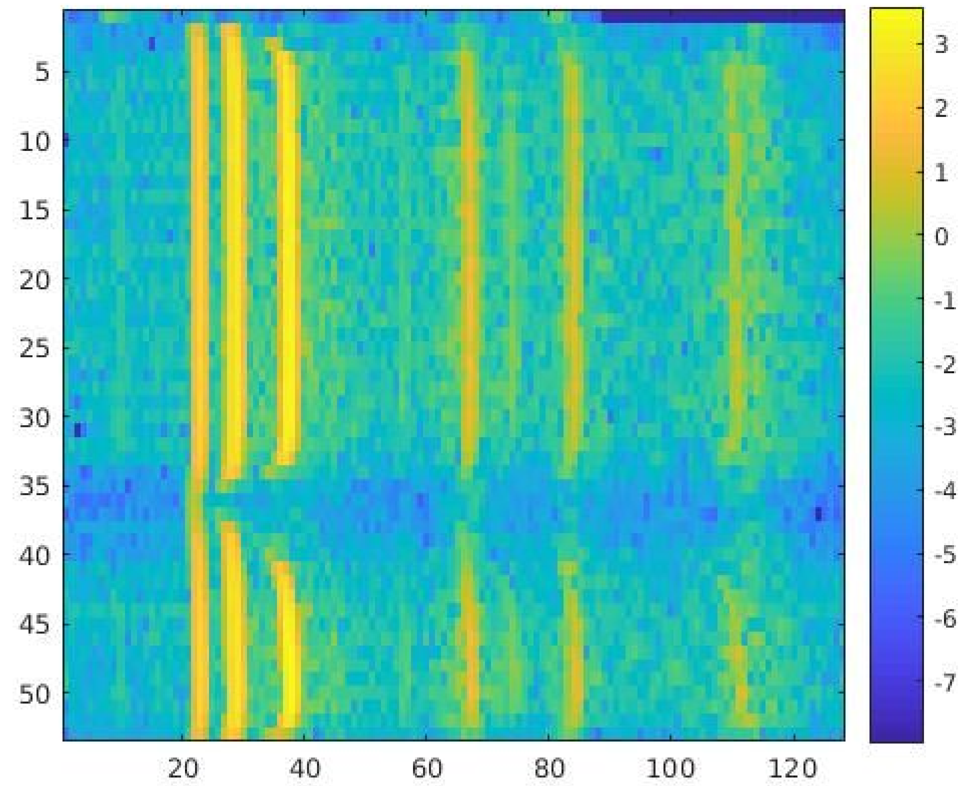


Increasing the window length causes increase(enhancement) in data captured i.e there is more accurate plotting of frequencies in spectrogram. Resolution of spectrogram also increases. This is due to increase in overlapping of window with the given signal at a time. Also the gap between two window also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

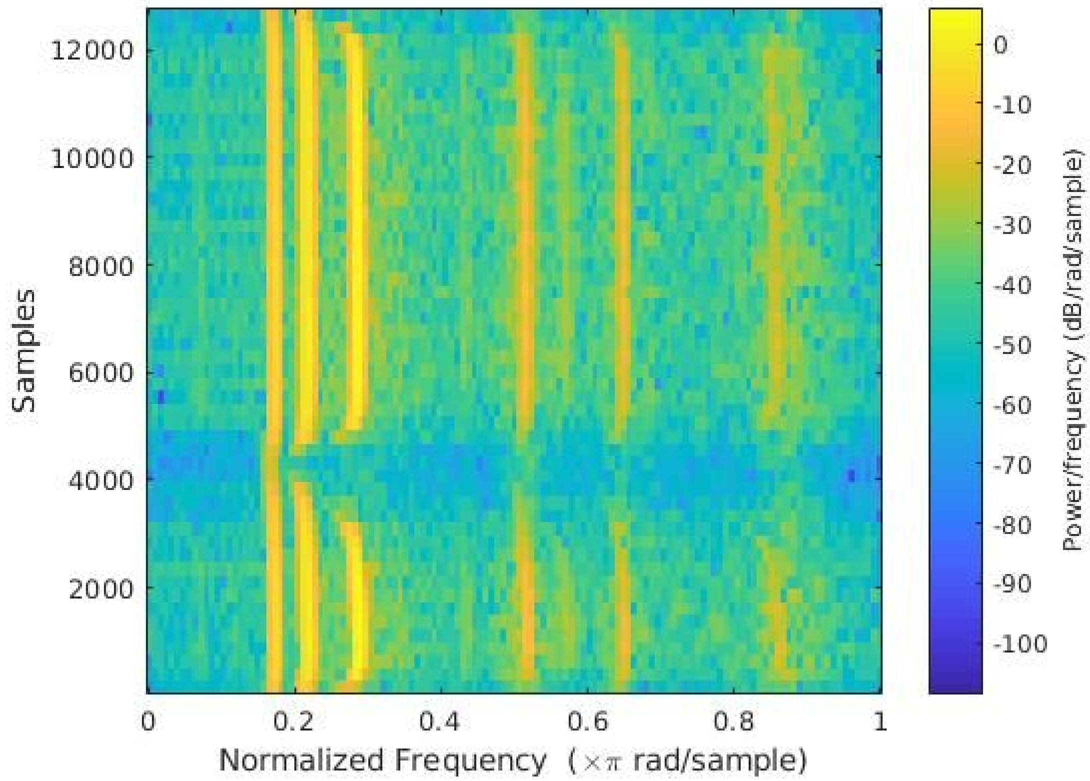
<u>Comparison Between my result and inbuilt</u>

1. WindowLen:256, StrideLen:10

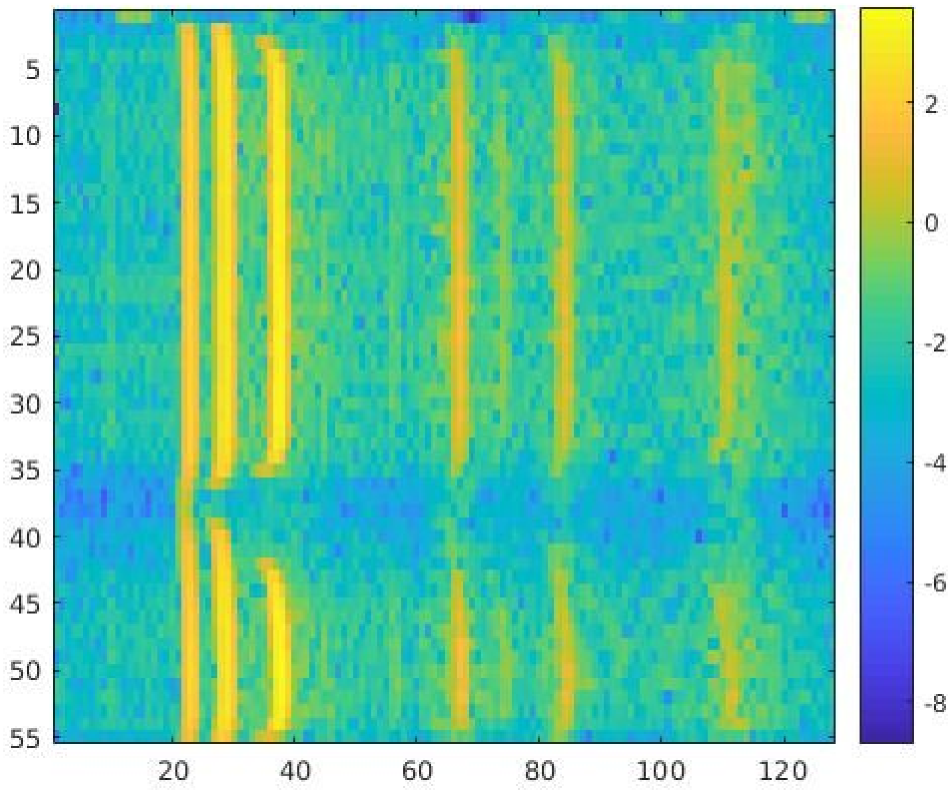
- My Created Function



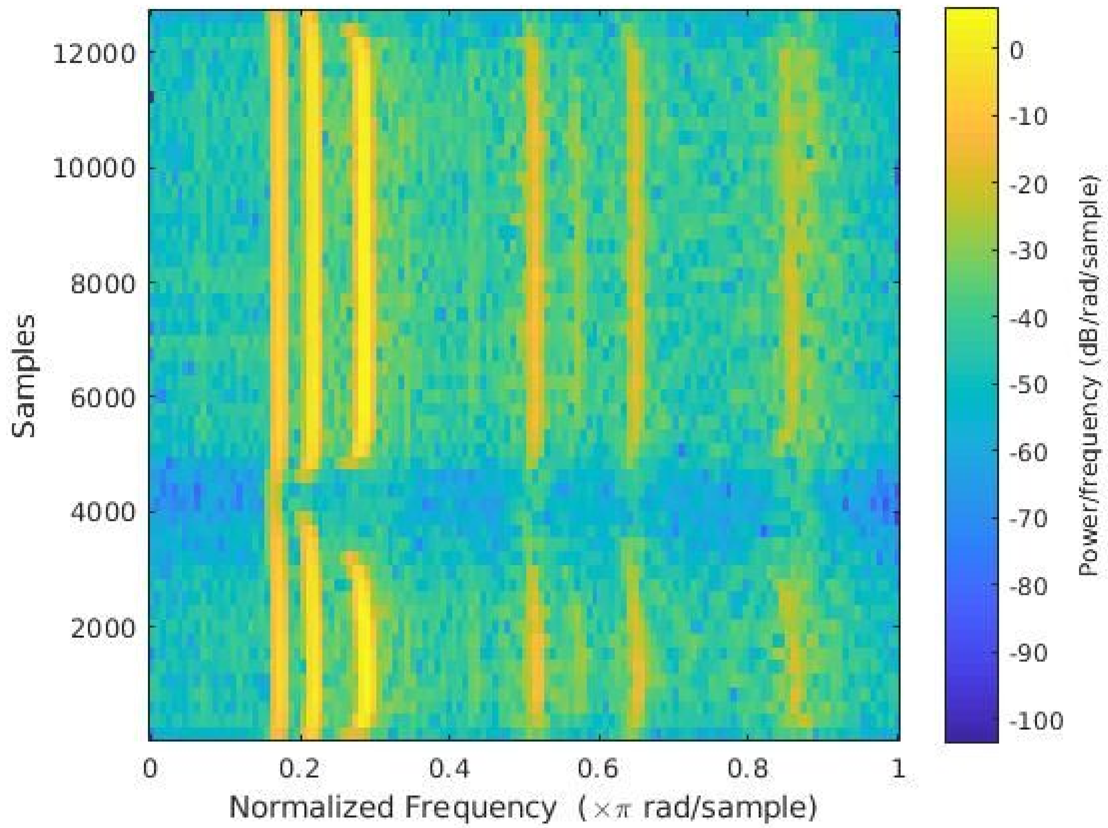
- Inbuilt Function



2. WindowLen:256, StrideLen:20
   - My Created Function

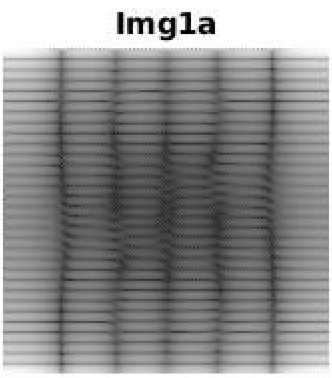


- Inbuilt Function

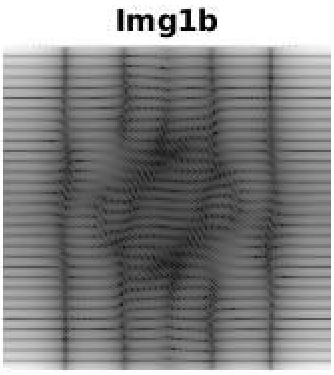


QUESTION - 4: fft2 And Noise Removal

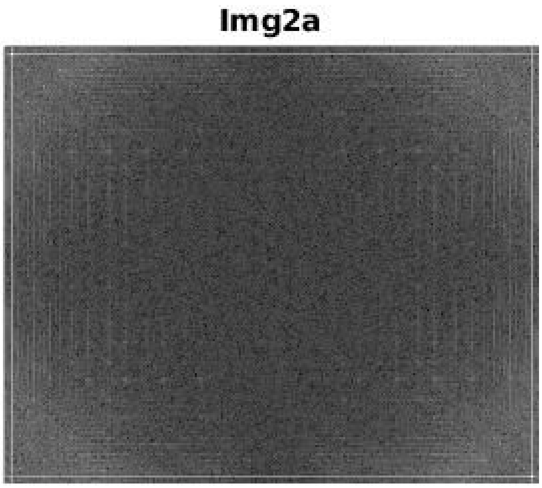
- Img1a.png



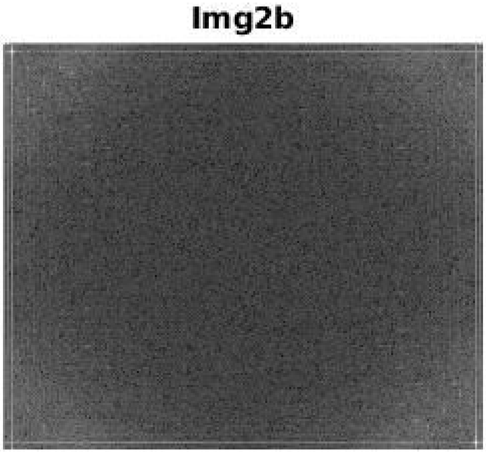
- Img1b.png



- Img2a.png



- Img2b.png



- Img3.png Noise Removal:

In this part, I have used the median filter (The whole code of which I got on the internet, and I took a lot of reference from that code). At the end, I have used the inbuilt functions *imadjust* and *imsharpen* to brighten and sharpen the image.

NOTE: I also tried to use lesser number of values in the filter, which gave a very dark image, from which was actually very hard to find out the noise-cancelled image.

QUESTION - 5: Eavesdrop

In this question, we are given an audio file containing tones of dialled numbers. We have to identify the number dialled from the tone, and we are given the individual dial tones. So, first we have stored all the individual dial tone signals and get the information about the given audio signal using *audioinfo* function. Using this we get the Total Samples present in the audio signal and the Sample Rate. Using this information we get the required window length to work on.

We are then iterating from one sample to another sample in the given audio signal and matching frequency of the audio sample with the stored individual dial tone using the dot product to find the max of the match of the dialed number. If there is a match then we are combining that individual dial tone to the result. The iteration will be there till the end of the window length.

At the end of the iteration, the result we get will give us the number dialled on the phone.

QUESTION - 6: Decryption Messages

In this question, we are given a encrypted message and a key. We have to decrypt the message using this key so that decrypted message is meaningful. So, first we have to convert the signal to frequency domain using the fourier transform. Since n = 4 , so we have the segmented the signal into 4 parts. On observing the plot of fft of the signal we can see that the signal is flipped at (length of signal)/2.

We have segmented the signal into 4 parts i.e segment 1 is from 1+length/2:5*length/8, segment 2 is from 1+5*length/8:3*length/4, segment 3 is from 1+3*length/4:7*length/8, and segment 4 is from 1+7*length/8:length.

Since there are 4 segements, there will be 24 different ways of permuting the key. So we are iterating and generating 24 wav files and out of 24 files we will be getting at least 1 file containing the meaningful message.

The decoded messages are:

1. If you are good at something, never do it for free.
2. Why so serious?
3. Let's put a smile on that face.