DSSA ASSIGNMENT – 2 REPORT
Himanshu Maheshwari (20171033)

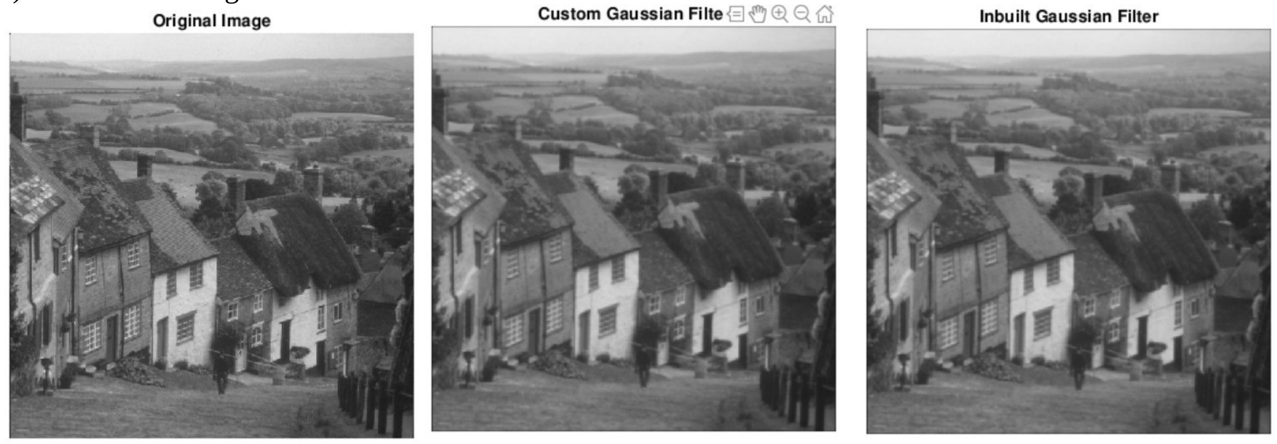
Question – 1

Part-1

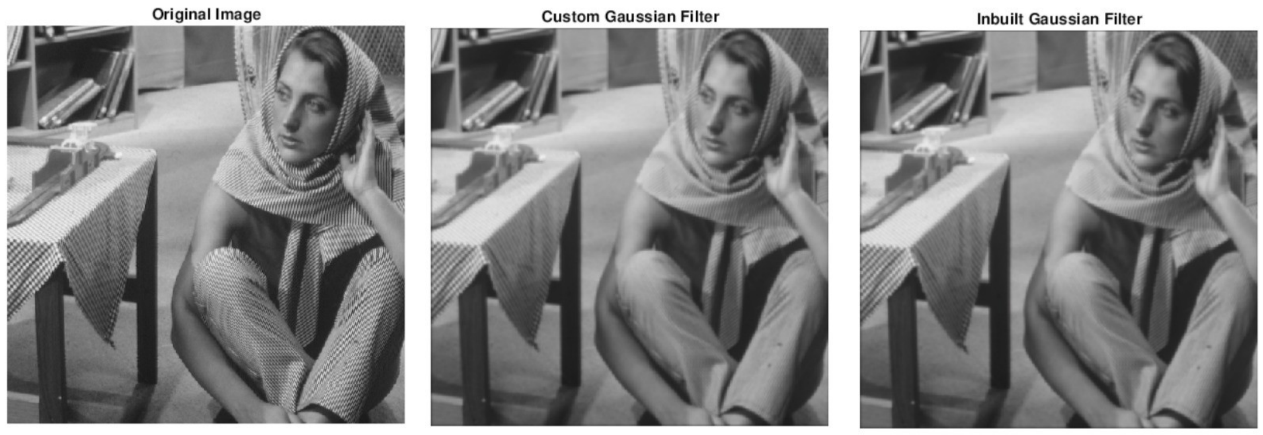
First I convert N to N/2 (use floor to take care of odd values of N). This is done because when we create a matrix of size NxN with value in rows ranging from -N/2 to +N/2 and same in columns. Then I use meshgrid to create x and y matrices. Both of them contins values from -N/2 to +N/2. Then we get the gaussian filter using the formula. Then we normalize the ouput.
We use imfilter to use this filter on image.

Output

1) N = 64 Sigma = 1



2) N = 128 Sigma = 1.2

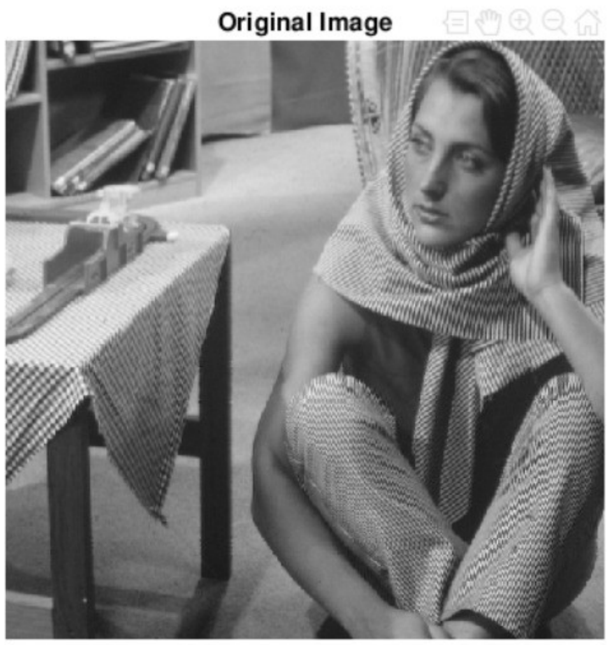


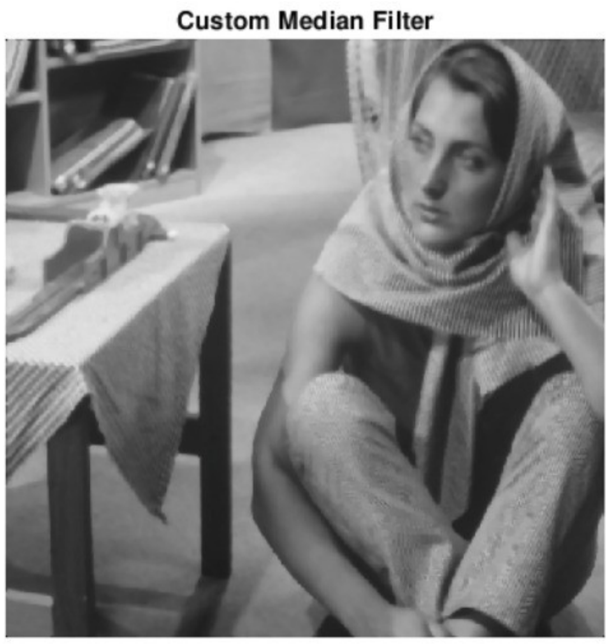
Part-2

First we padarray with so that im2col function would not give an error. Then we use im2col to conver all the values that are in NxN matrices into column. Then we calculate the median of each of the column and store it into med. Then we conver this vector into the image using col2im function.

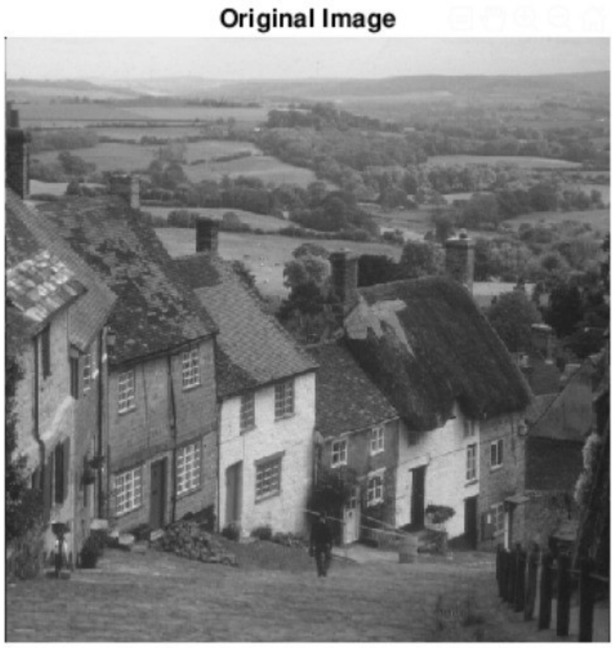
Output

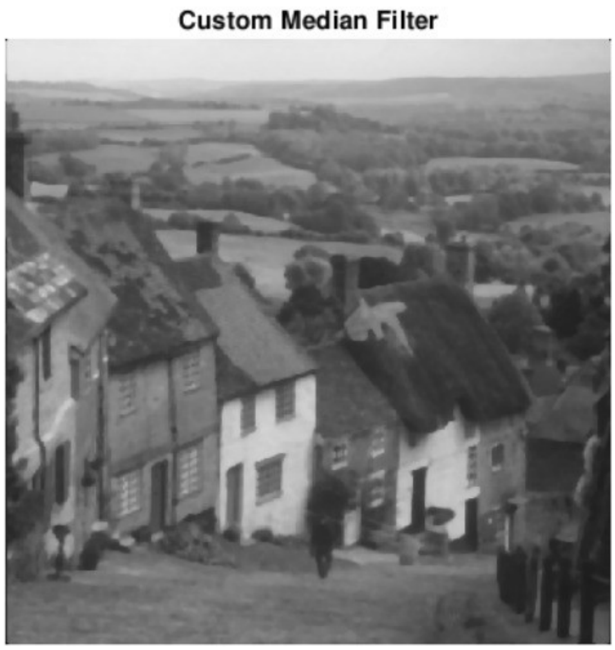
1) N = 5





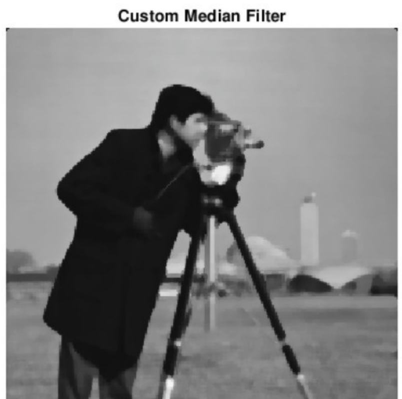
2) N = 5

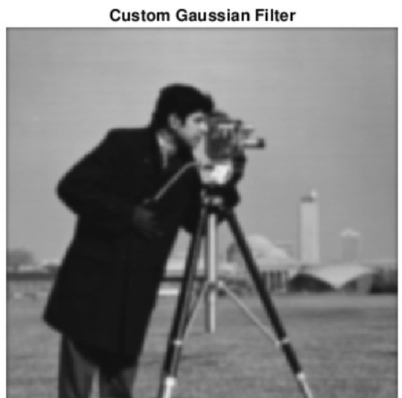




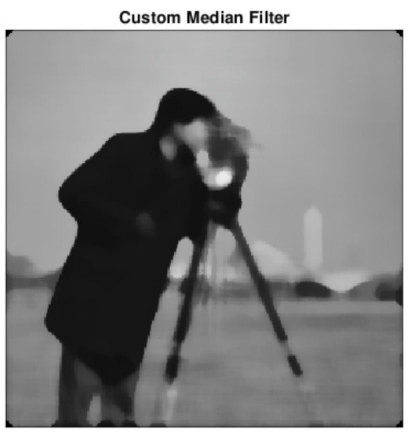
Part – 3

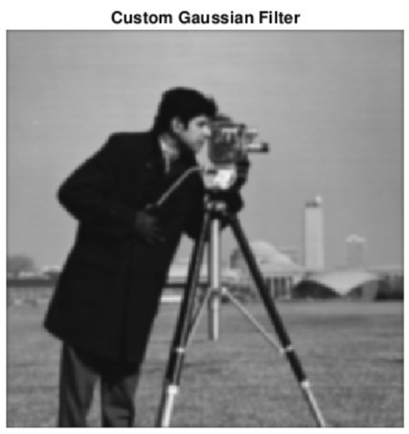
1) For Gaussian : N = 128, Sigma = 1.2 For Median : N = 5





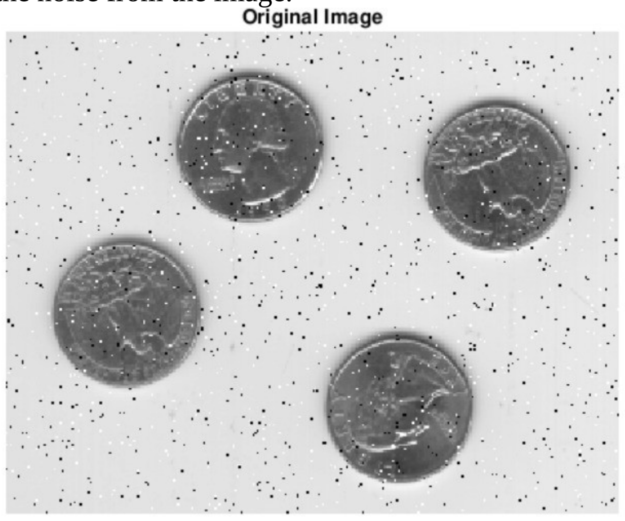
2) For Gaussian : N = 256, Sigma = 1 For Median : N = 10

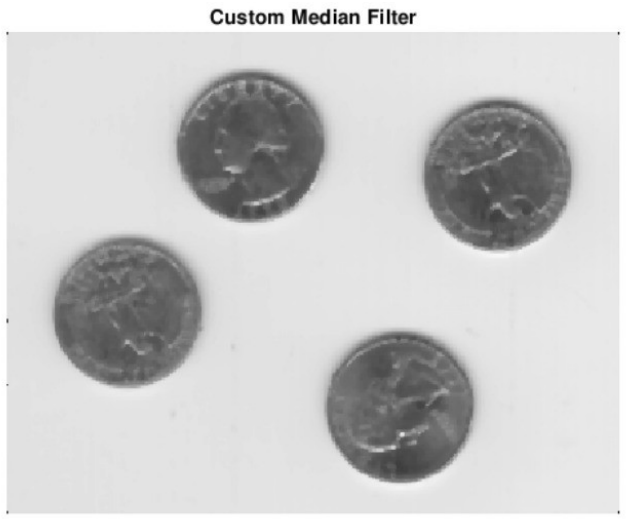




Part – 4

The noise which is there in the image is called salt and pepper noise i.e there are some sharp and sudden disturbances in the image. Now the best filter to remove such noise is median filter as the median filter block of size NxN replaces the central value of the NxN neighbourhood with the median value and thus remvoing the noise from the image.

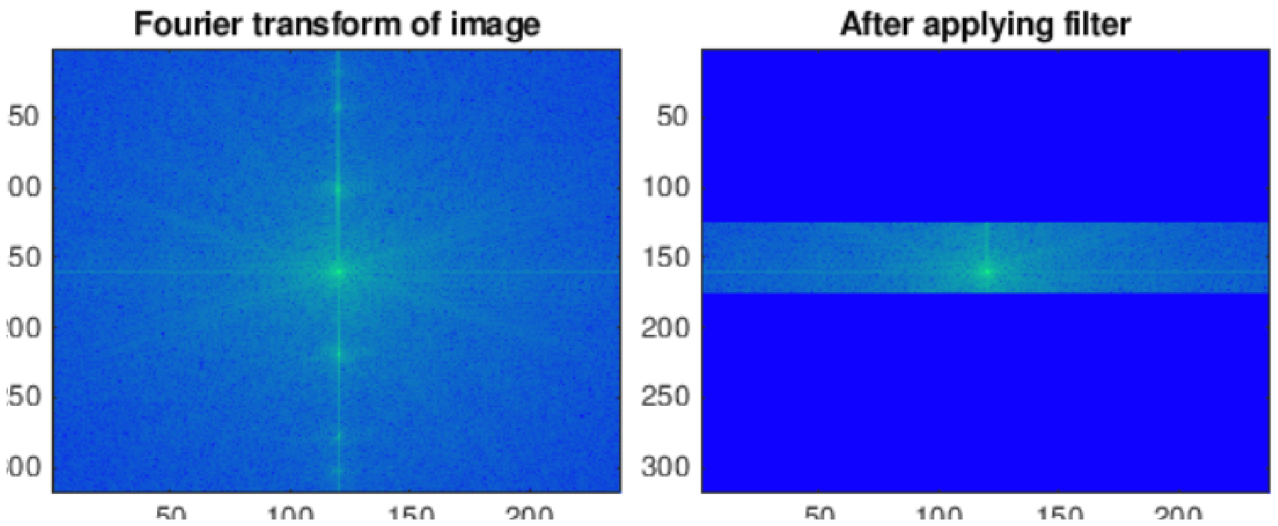




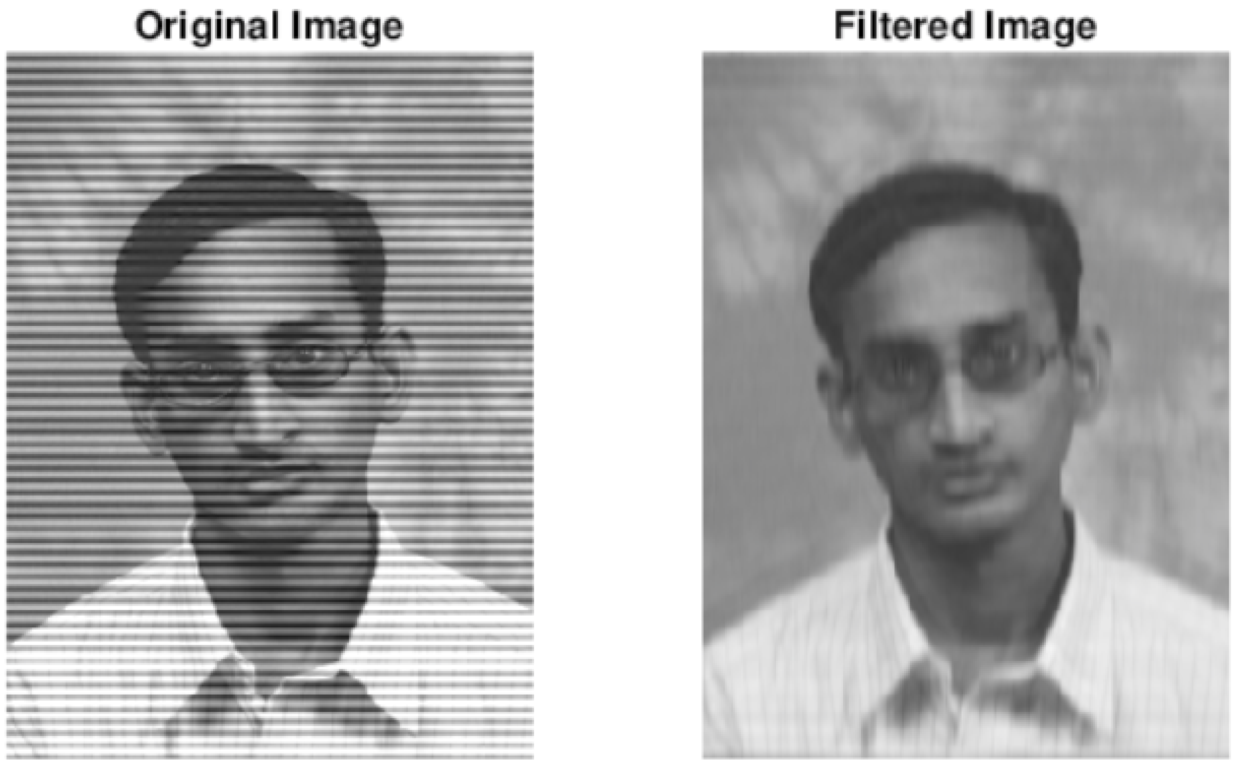
Part – 5

First I take the fft of the image and does the fft shift. Now I see peak at the center(the peak of image), and several other peaks(of noise). So the main work is to remove these peaks. After seeing the fourier transform I found out the signal that needs to be clipped, which is from rows 1 to 125 and rows 176 to 318 and thus I create the matrix clipper to remove that part. After removal I show the image.

Fourier Transfrom:



Result



Question – 2
Part – 1
Now there can be two case:
Case 1) All the filters are applied recursively and after each filter the image is zero padded. In that case the width of the image after applying ith filter would be : W(i) = (W(i-1) – F + 2Z) / S + 1, where W(i) is the width of the image and other parameters are same as that given in the question. Similarily height (H(i)) after applying ith filter would be H(i) = (H(i-1) – F + 2Z) / S + 1. Here W(1) = (W - F + 2Z)/S +1 and H(1) = (H – F + 2Z)/S+1 where H and W are original height and widht of the image.

Case 2) If the filters are applied independent of each other i.e. we apply ith filter on the image, then jth filter on the image(i.e. original image only, not on the image that we get after applying ith filter) then the widht is (W – F + 2Z)/S +1 and Height is (H -F + 2P)/S + 1 where H and W are orignial height and width of the image.

Part – 2

For one convolution the number of multiplications at each step is F^2 and the number of addition is (F^2-1). Thus the total number of additions and multiplications for one input channel is ($2F^2$-1)*(output width)*(output height).
Case 1) Total operations will be ∑(W(i)) * (H(i)) * ($2F^2$-1) where i varies from 1 to number of channels.
Case 2) Total number of operations will be (W * H * ($2F^2$-1) * number of channels), where W is widht of the image and H is height of the image.

Question – 3

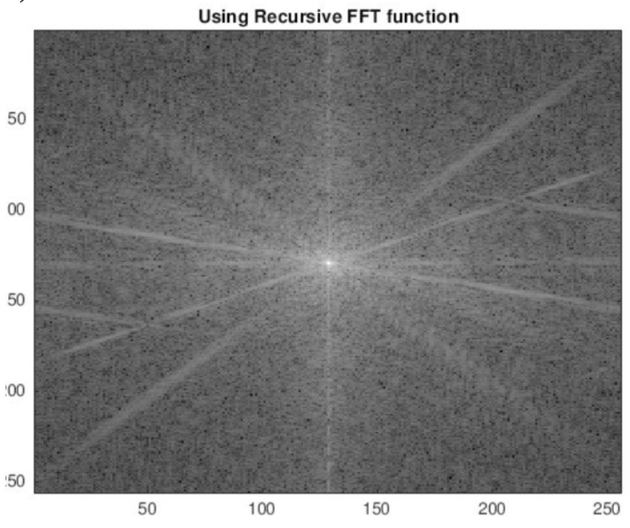
Here it is assumed that the fugutive pressed the keys for 1 sec each. Now first we get the number of seconds did the fugutive talked for which is represented by ‘len’. Then we read each of the file corresponding to each digit and store them in an array. Now we use cosine similarity to find the similarity between two vectors. So for each second we find the cosine similarity between the samples at that second and each of the dial tone sound. Now say that fugutive has pressed 3 at 5th second then cosine similarity between 5th dial tone (5.ogg) and samples between 5th and 6th second would be maximum(1). Thus we find the maximum among all the cosine similarity and take that number that has maximum cosine similarity and finally display the result .

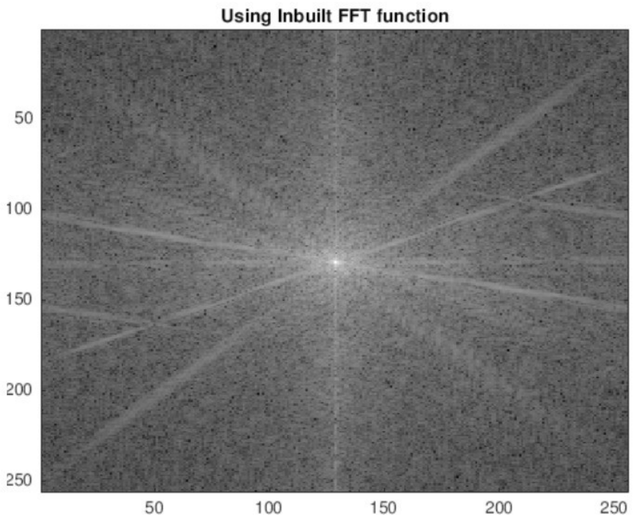
Question – 4

After listening to all the original signal I felt that there are some shar and sudden disturbances in the sound. (Something analogous to salt and pepper noise in images). So I tried various filter, used fft to remove noise yet I found that median filter gave me the best result. So I have used medfilt1 to remove the noise using median filter.

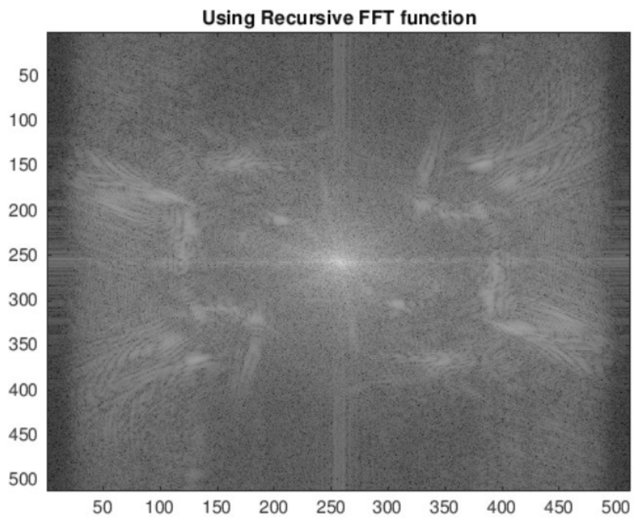
Question – 5
2D Fourier Transform
The 2D fourier transform that I have implemented first calculate the 1D fourier tranform of each row. Then it calculates the fourier tranform of each column to get the final fourier transform of the 2D image. After that it does fft shift and display the 2D fft as an image. The image is shown in grayscale as it gives more prominent idea.
How to calculate 1D Fourier Transform?
The approach that has been used to calculate 1D fourier transform is the recursive one which has a running time of O(NlogN). Now the formula that has been used is the one that was discussed in the class as it can be seen.

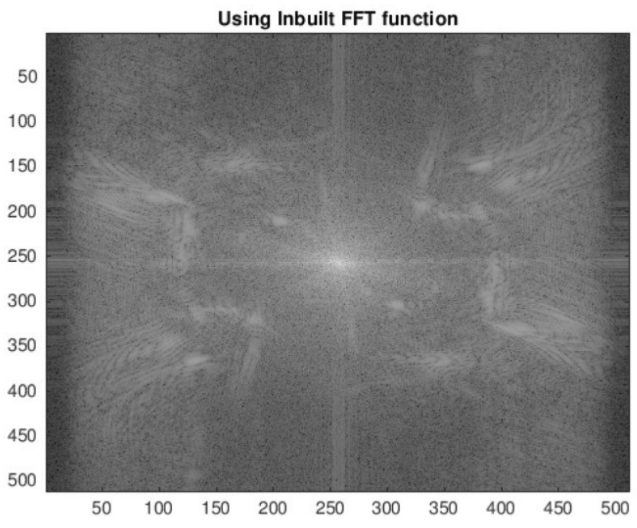
Output :
a) cameraman.tif



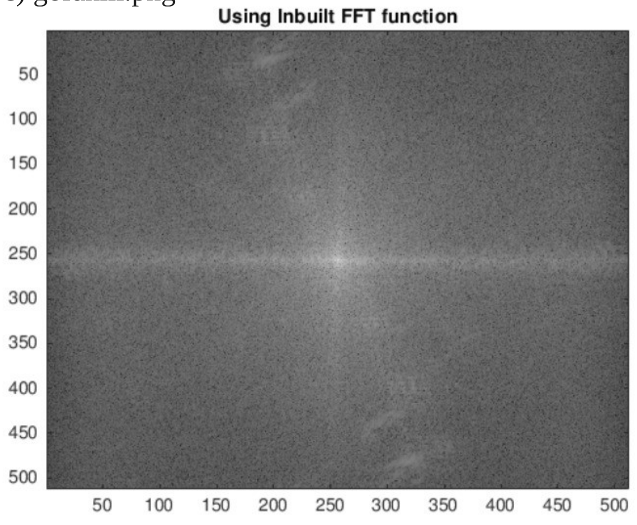


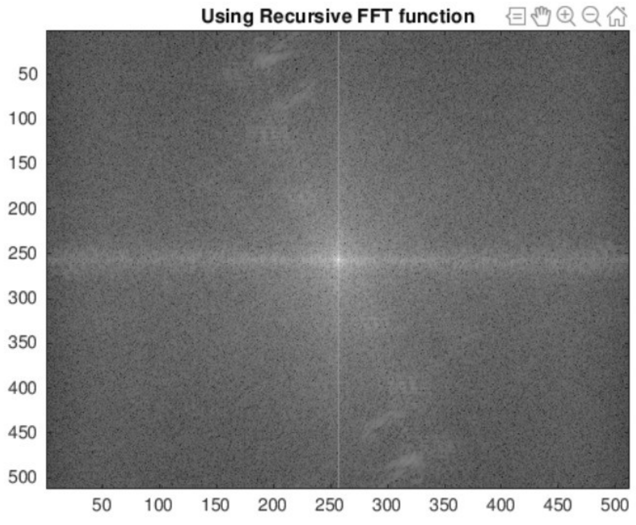
b) barbara.png





c) goldhill.png





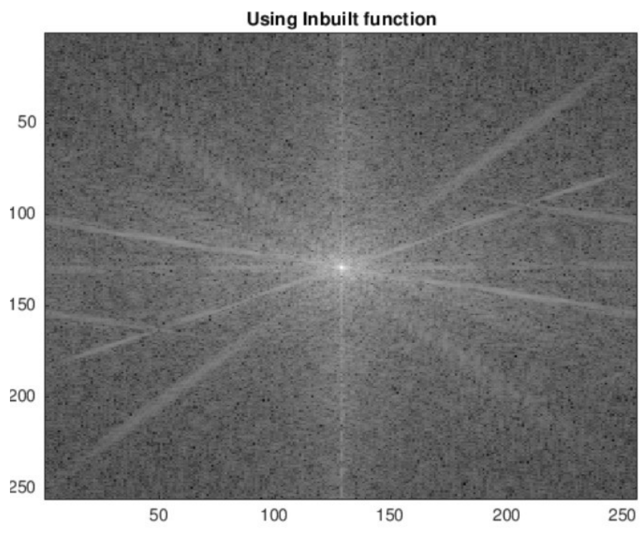
DFT

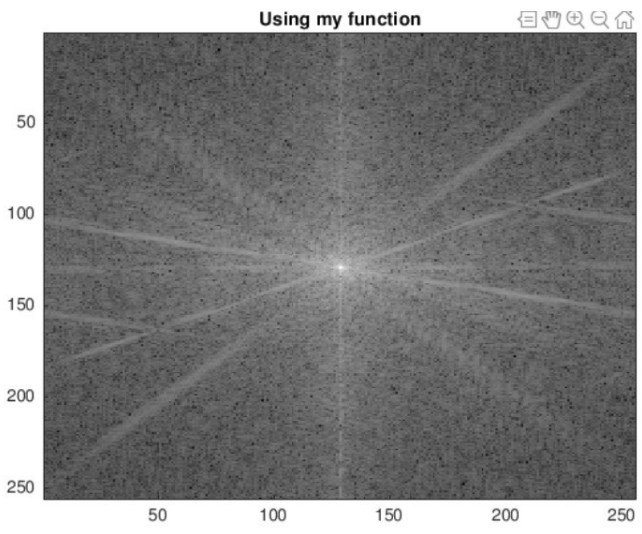
The formula for DFT is : $X_k = W_n.x_N$

now our main task is to calculate W_n matrix. Now to calculate W_n matrix we first calculate temp matrix and then multiply it in exponent. The 'ex' matrix is actually W_n matrix. Thus we have $X_k = W_n.x_N$ and we know x_N and W_n and thus we can calculate X_k which is DFT of x_N. Now to calculate 2D DFT we follow the above steps.

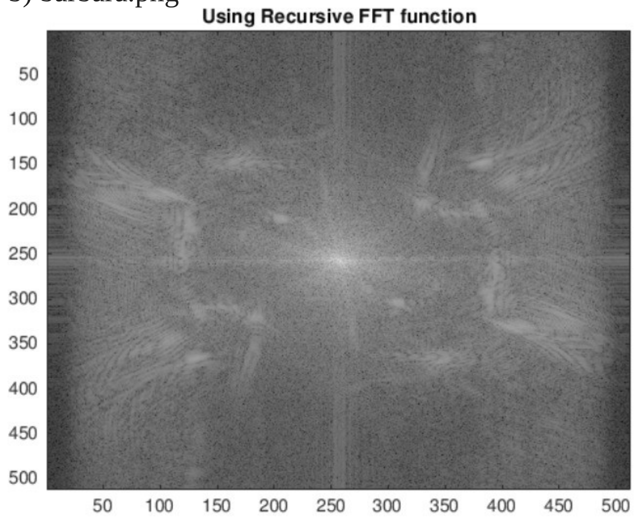
Output:

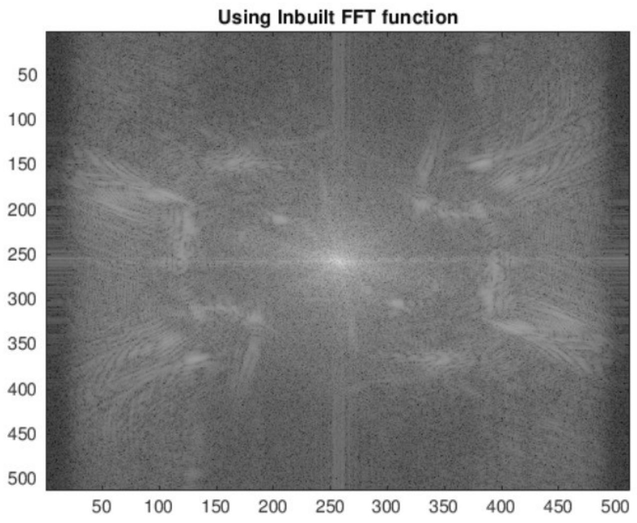
a) cameraman.tif



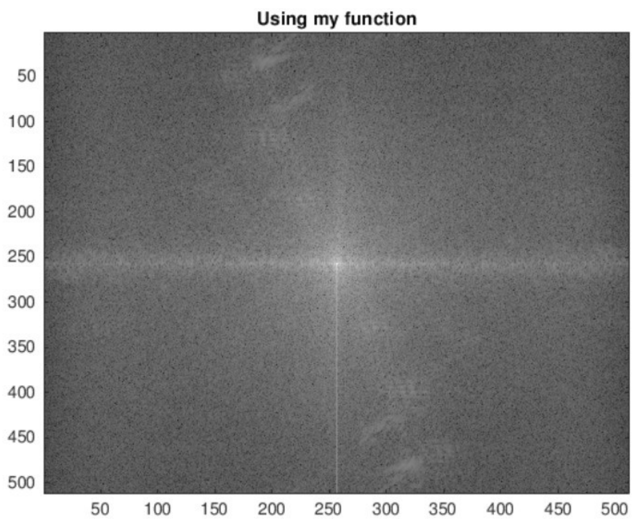


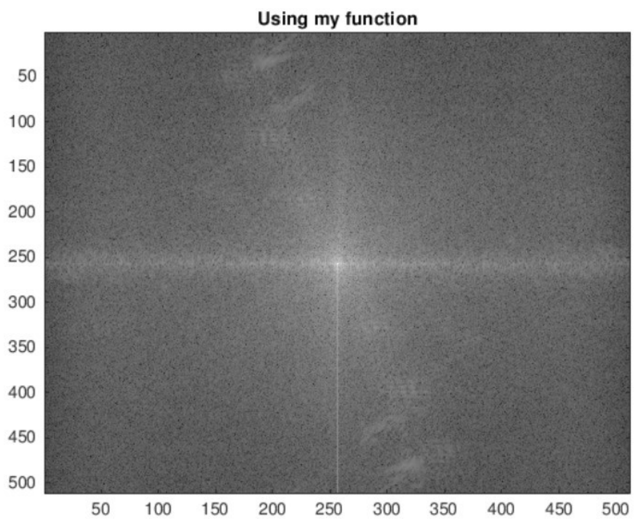
b) barbara.png



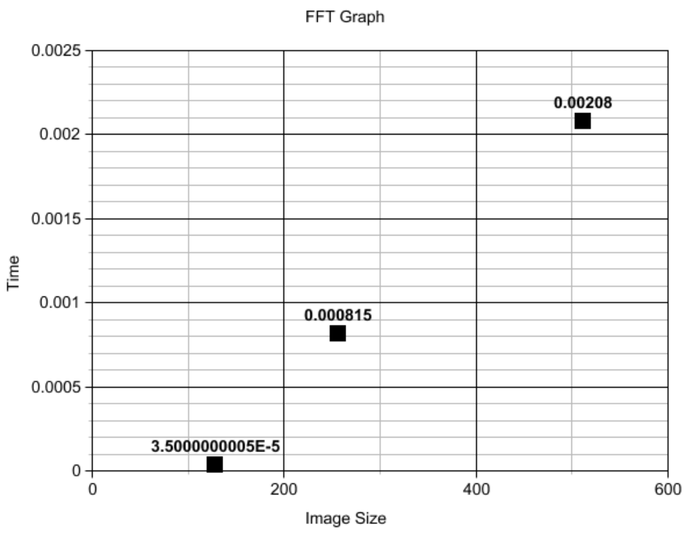


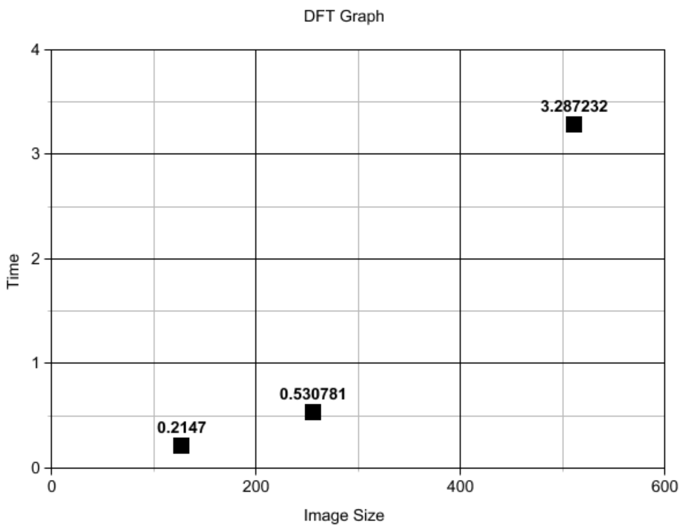
c) goldhill.png





Run Time Comparasion :





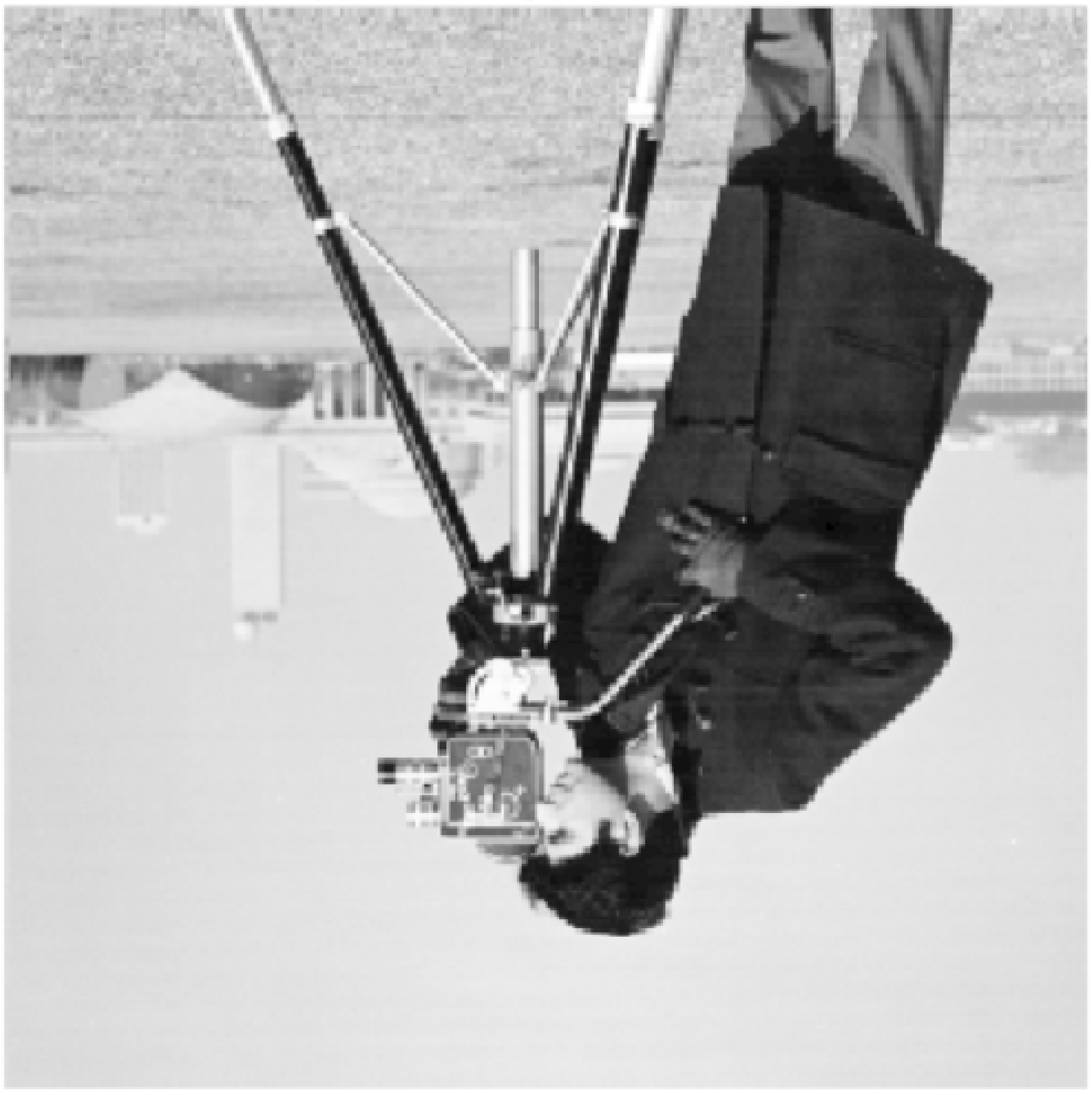
As we can see that FFT is very fast than DFT, which is expected as the time complexity of FFT is O(NlogN) whereas time complexity of DFT is O(N^2).

Question 6

After applying fft twice I observed that the output image is actually the reverse of the input image. Mat2gray converts a matrix into a grayscale image as can be seen. While taking fourier transform twice we are actually multiplying W_N matrix to W_nx vector. Now to get the original image back from W_Nx we multiply it with inverse of W_N, however here we are mutliplying it with W_N. So instead of multiplying the original rows we are actually multiplying their conjugates (We know that $F(k) = F^*(N-k)$, hence the image rotates by 180^o. Also the image becomes lighter as each entry of x is getting divided by N^2. To get the original image, we rotate it by 180^o or by applying fft four times instead of two.

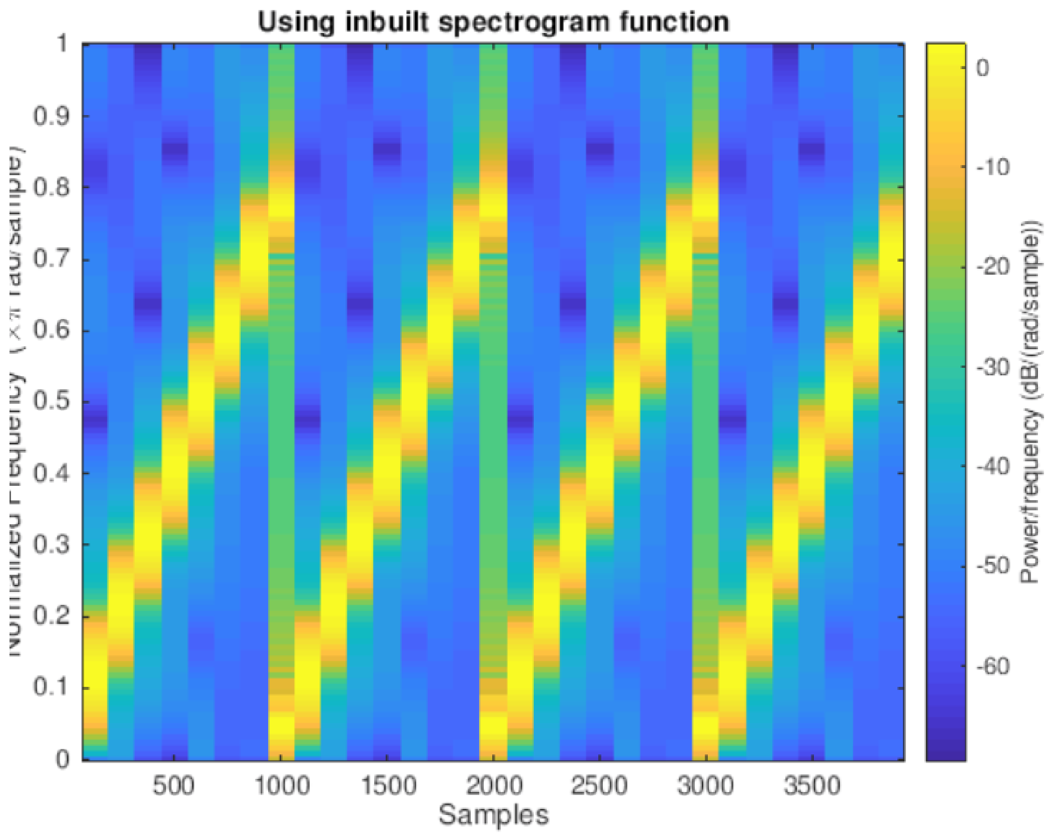
Output :

Output after applying fft twice



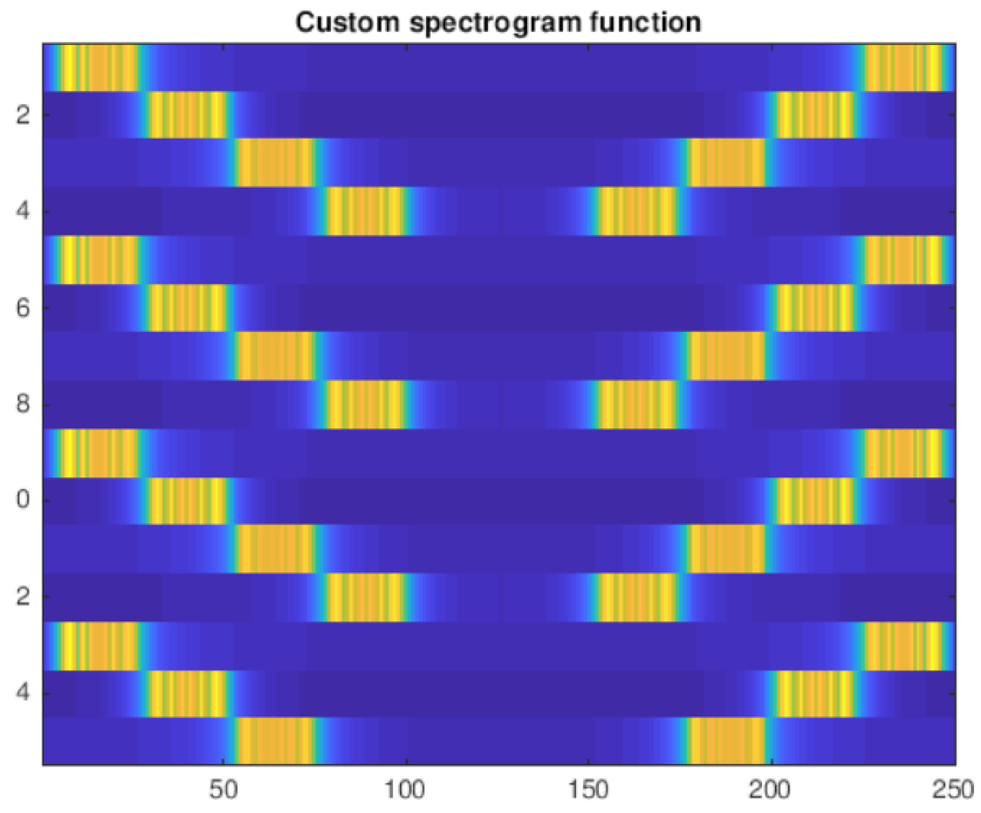
Question 7
Part 1
First we calculate the step lenght i.e the length by which the window moves. Then we have starting point vector which stores the starting point of each window. Then for each window we get the fft, and store it in the output matrix. Now the output matrix has dimensions of total number of starting point x window size.

Output
Using inbuilt:

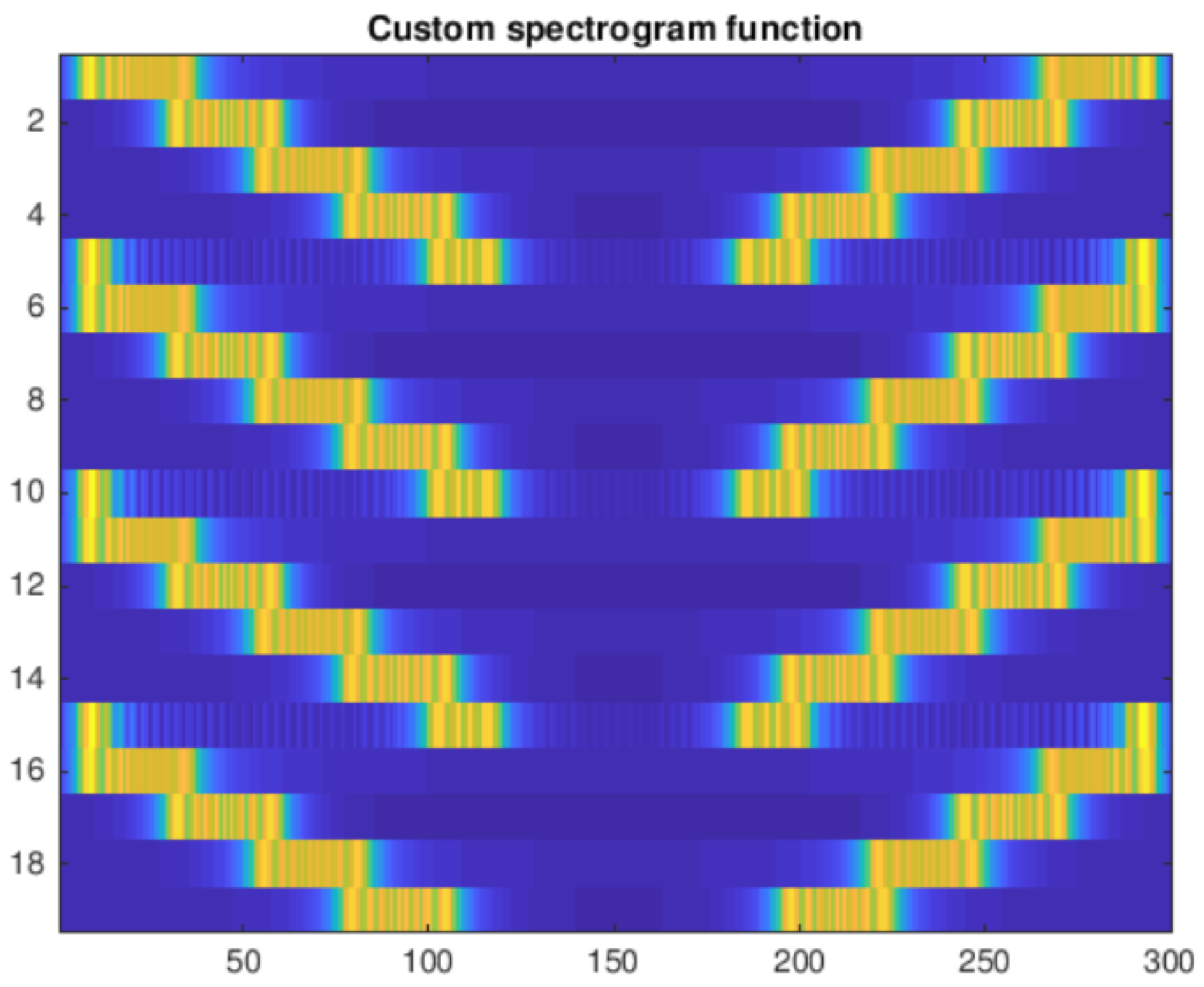


Using Custom:

1) Window size = 250, stride length = 0

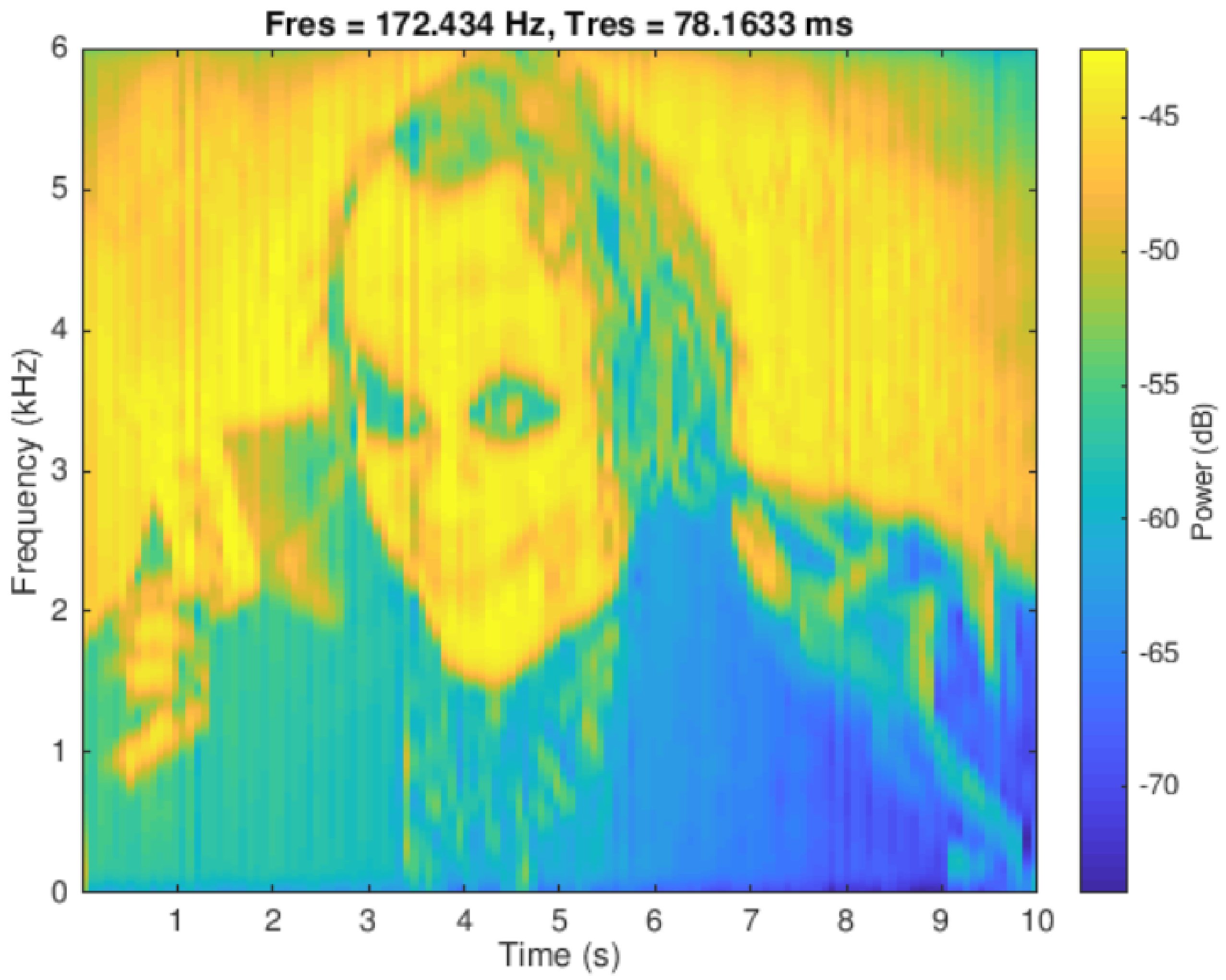


2) Window size – 300, stride length – 100



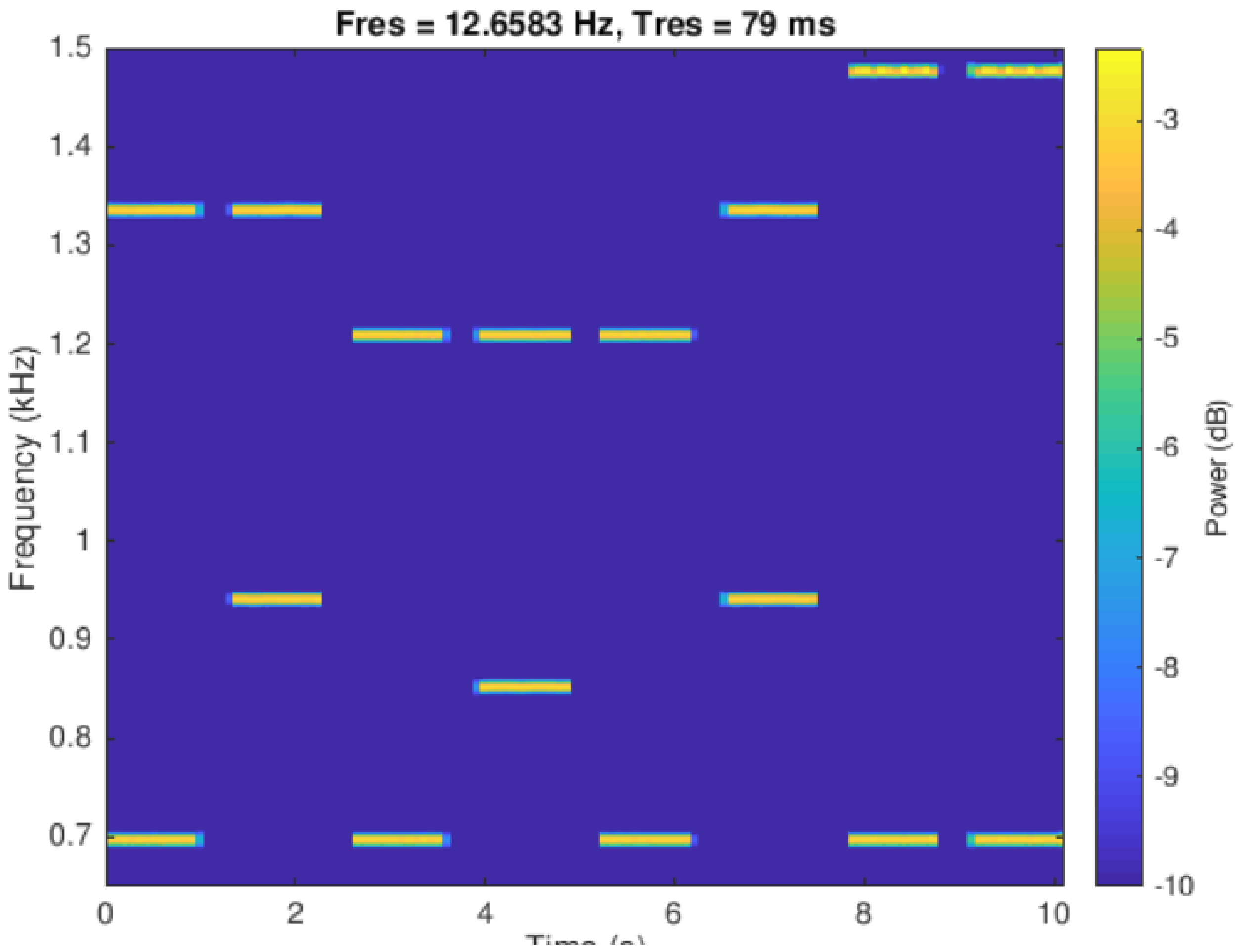
Part 2
Here simple spectrogram of the input message will show that the spectrogram formed is that of joker from the movie “Batman – The dark knight”. The password thus has something to do with joker or maybe the actor’s name , Heath Ledger or maybe his famou diallouge “Why so serious?”.
I have used psepctrum function, which plots spectrogram using power of the signa, because spectrogram function does not allow frequency limit of 0 to 6kHz. Though spectrogram function will producde the same output but a little flattened.

Output :



Part 3
The function dial_tone will take any rollnumber and will produce its sound file as asked in the question. The maximum frequency that is expected is 1477, so I have taken sampling frequency to be 3000 which is more that 2954. Now the signal for each digit is 1 sec and gap between them is 0.3 seconds. Now the rollno which is input to the function is an integer so we need to convert it into an array, which is done in line 6. So now I loop over every digit in the roll number, and create a sine wave corresponding to frequencies specified in DMTF. Next I concatenate the ouptu with the signal just created and also add a wait period of 0 frequecny. If the digit is last digit then the gap is not added.

Output:


Fres = 12.6583 Hz, Tres = 79 ms
Frequency (kHz)
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0
2
4
6
8
10
Power (dB)
-3
-4
-5
-6
-7
-8
-9
-10