DSAA ASSIGNMENT-3 REPORT

Name: Himanshu Maheshwari
Roll No: 20171033

Using Two Days Extension.

Problem 1)
1. Coefficient corresponding to size of house: 141.8342
Coefficient corresponding to number of rooms: -6361.9
Constant: 77731
Predicted value of price of house with area of 1400 sq. Meters and 4 bedroom is: 250851.4552

2. No normalizing did not help in any way. The L2 norm without normaliztion is 125572.1203 and with normaliztion is 125572.1203. As L2 norm is equal to L1 norm so normalization does not help.

3. Mean of size of house: 2000.7
Mean of number of rooms: 3.1702
Mean of cost of house: 340412.6596

Predicted value of cost when we take house size of 2000.7 and number of rooms as 3.1702 is 341327.5792. Now this is not equal to mean of cost of house so it does not pass through the regression line/plane generated. However this difference is very low.

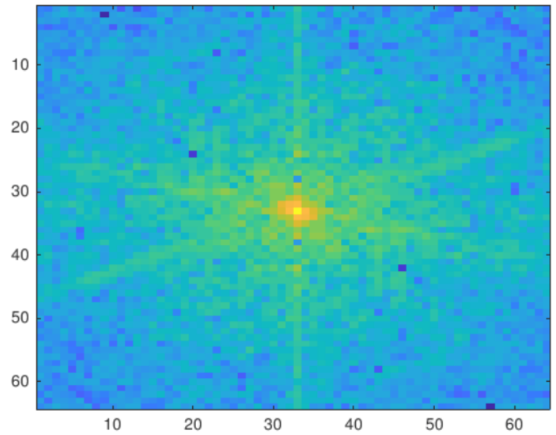
4. No the same method could not be used to solve this problem if the number of rows in the data is 1 million because here to get the vector 'b' we use matrix multiplication which is of order $O(N^3)$. Now this will involve huge calculation and thus the time needed to compute this will also increase very much making it infeasible to use.

Problem 2)
The two images that are used is 'barbara.png' and 'cameraman.tif'
1. iDFT[FH] is not equal to f*h as it can be seen by there average squared difference(which is calculated below and is very high).

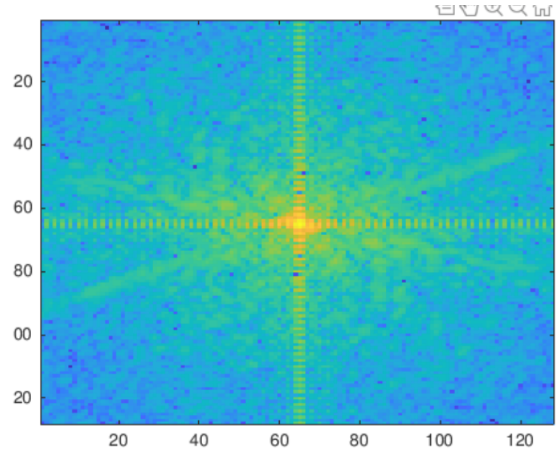
2. The average squared difference between pixel values in iDFT[FH] and the central 256 X 256 portion of f * h is 1.1108×10^{11}.

3. When we zeropad the original images to dimension (511 X 511) and then calculate iDFT[FH] and compare it with f*h, we find that the iDFT[FH] and f*h are alomst similar and their squared difference between the pixel values is almost zero. The reason could be that when we zero pad linear convolution converts to circular convolution for which the convolution theorem holds.

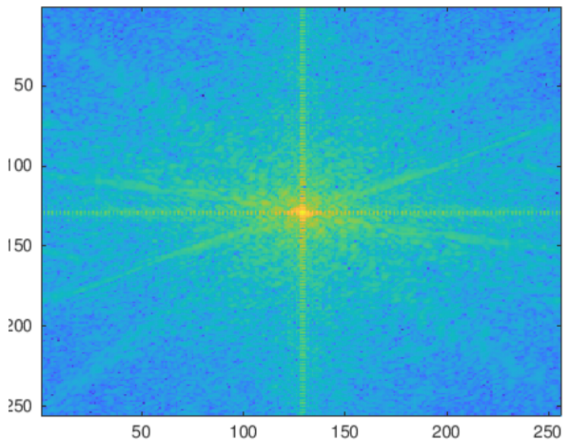
Problem 3)



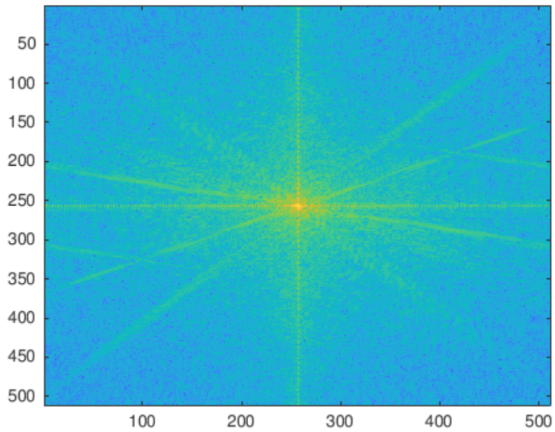
ORIGINAL IMAGE



128 X 128



256 X 256



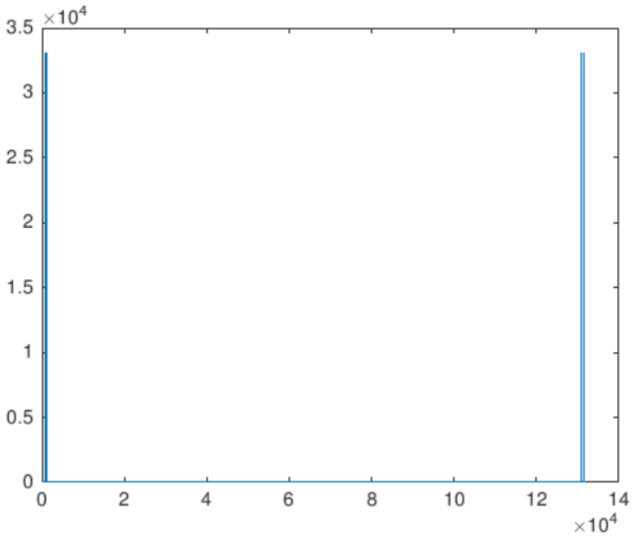
512 X 512

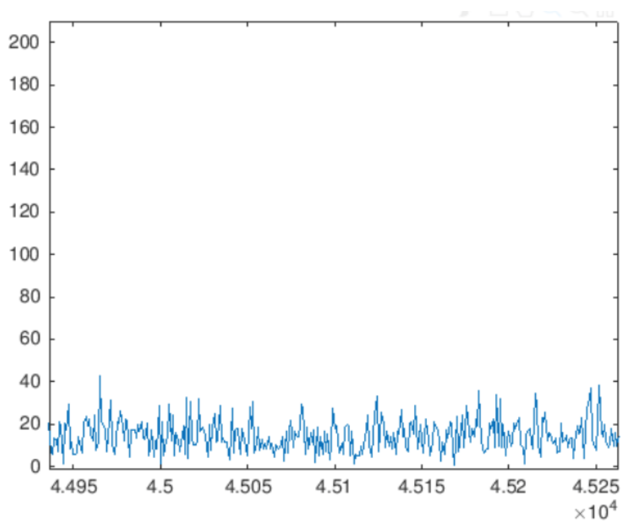
We observe that the center portion remains the same even after padding and we see vertical and horizonatal disturbances in the singal. These occur as we introduce discontinuity when we pad with zeros. The effect is seen as sinc function in the Fourier Domain

Also by zero padding we get smoother looking fft. The reason is that the original image in all four of them remains the same however we have zero padding in the right and bottom. Because of the zero padding we get longer FFT result vector. A longer FFT result vector has more data points and thus result into a smoother looking spectrum.

Problem 4)

First we take FFT of the X and plot it. Here two peak frequency and noise of very little amplitude throughtout the signal, as shown.

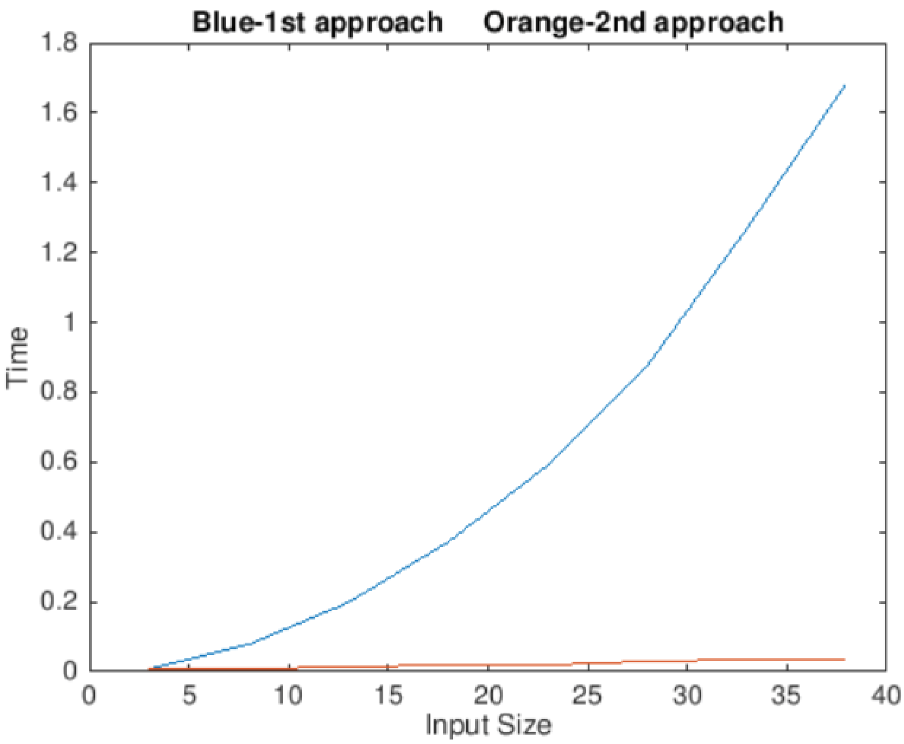




Now as it is clear from the FFT diagram that we have high frequency noise so we use low pass filter to remove these noise. As we observe that there are two frequency of value 882 and 1321. Thus any filter that inhabits frequecy higher thatn 1321 will do. I have used lowpass filter with cutoff frequency of 1500.

Problem 5)
It is given that there is an overlap of 3 to 5 seconds. So first we take 5 seconds(maximum overlap) of segment at starting and ending of all the sounds. Then to compare them we take xcorr of each and every starting and ending segment and take out top 4 values from it and store their sequence and thus in this way we get the required sequence which is **3,5,1,2,4**.

Problem 6)
If you have two numbers a and b such that both are not equal then their average will always come between them. Which means that the average will be less than the higher value and thus the higher value gets diminished while the lower value increases and thus in a way implementing low pass filter. When we use adjacent pixels to calculate average we observe that the computational time decreases significantly as shown in the figure below.



1st approach is by recalculating the average everytime.
2nd approach is by using adjacent pixel for calculating average.

Result:

Problem 7)
For finding gradient descent I have implemented two approach: Batch Gradient Descent and Stochastic Gradient Descent. Now for both the approaches I have run the gradient descent 100 times . On running the algoriths I had found that stochastic gradient descent approach is much more better than batch gradient descent approach as the root mean square difference in batch gradient descent(1346.5) is higher compared to stochastic gradient apporach(1182.7). Also there is mini batch gradient descent which I have not implemented but will perform better than both of the above apporaches.
Thus we have:

batch gradient descent < stochastic gradient descent < mini batch gradient descent