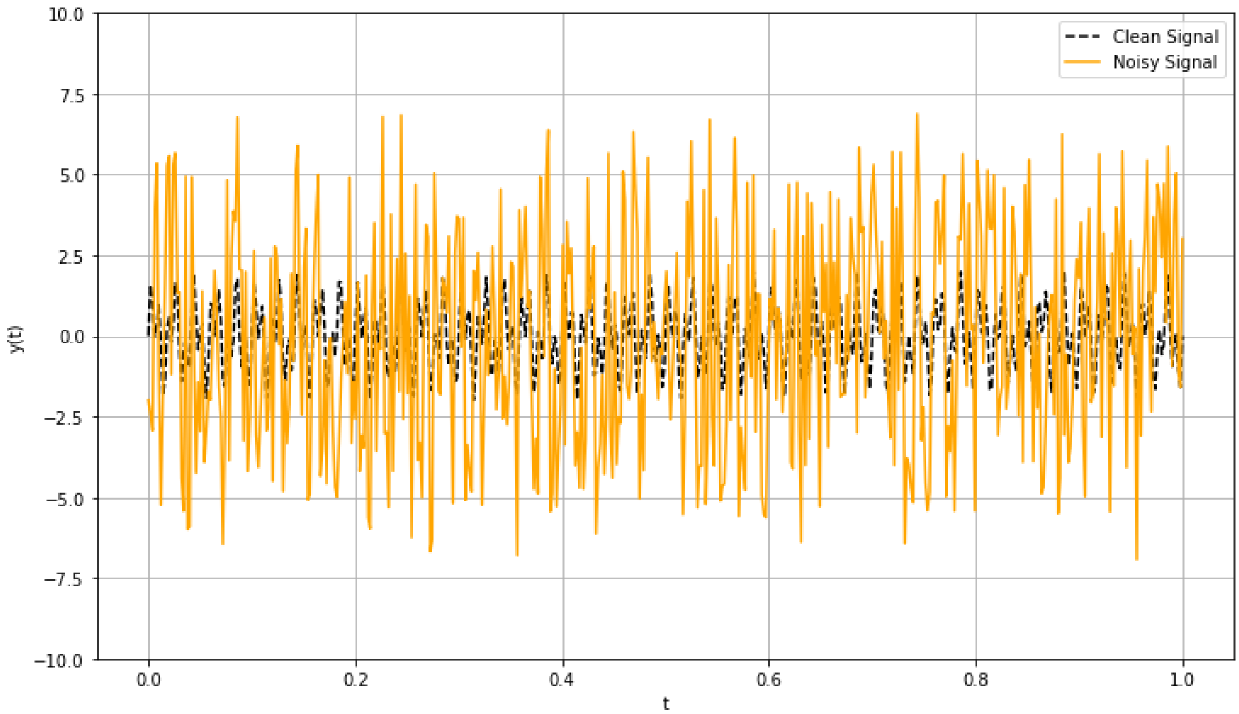
DSP Assignment 1

Parth Kalkar (p.kalkar@innopolis.university)

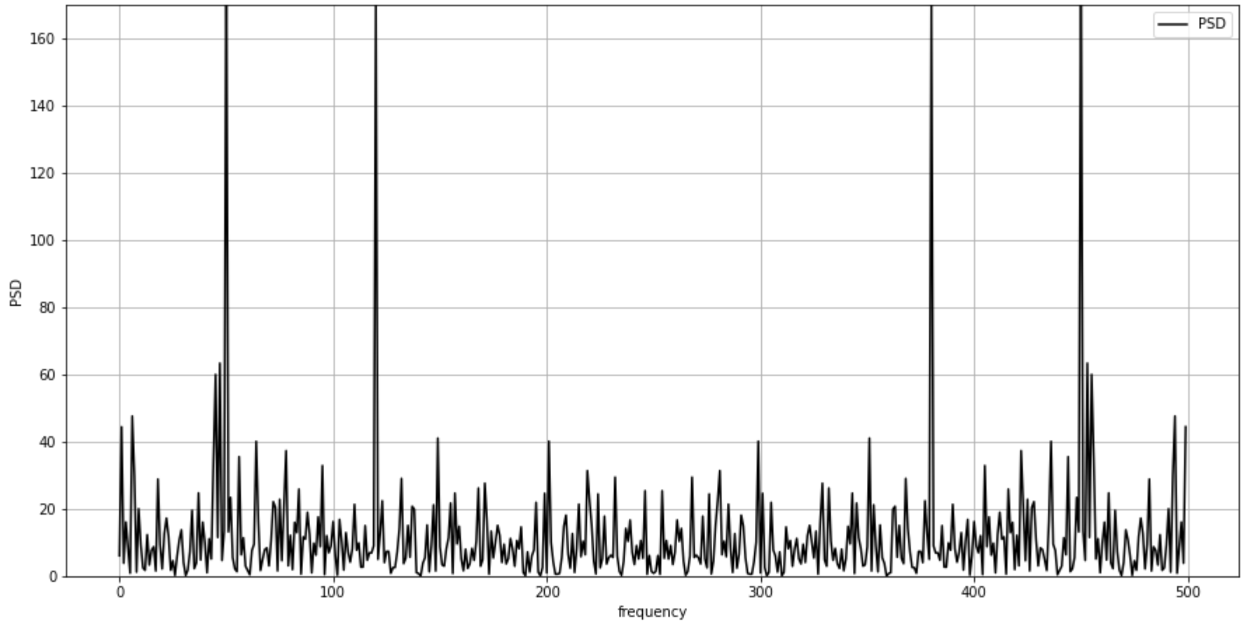
Task 1: Noise Reduction

1. The signals are as follows : $y(t) = sin(2\pi 50t) + sin(2\pi 120t)$ and $y\epsilon(t) = y(t) + \epsilon$. They are defined in the unit time domain $T = [0, 1]$. The ϵ represents some random noise added to the original signal. Here ϵ is chosen as a random integer between $(-5, 5)$. After discretizing the domain T in n evenly spaced subintervals, both functions can be visualized as follows:

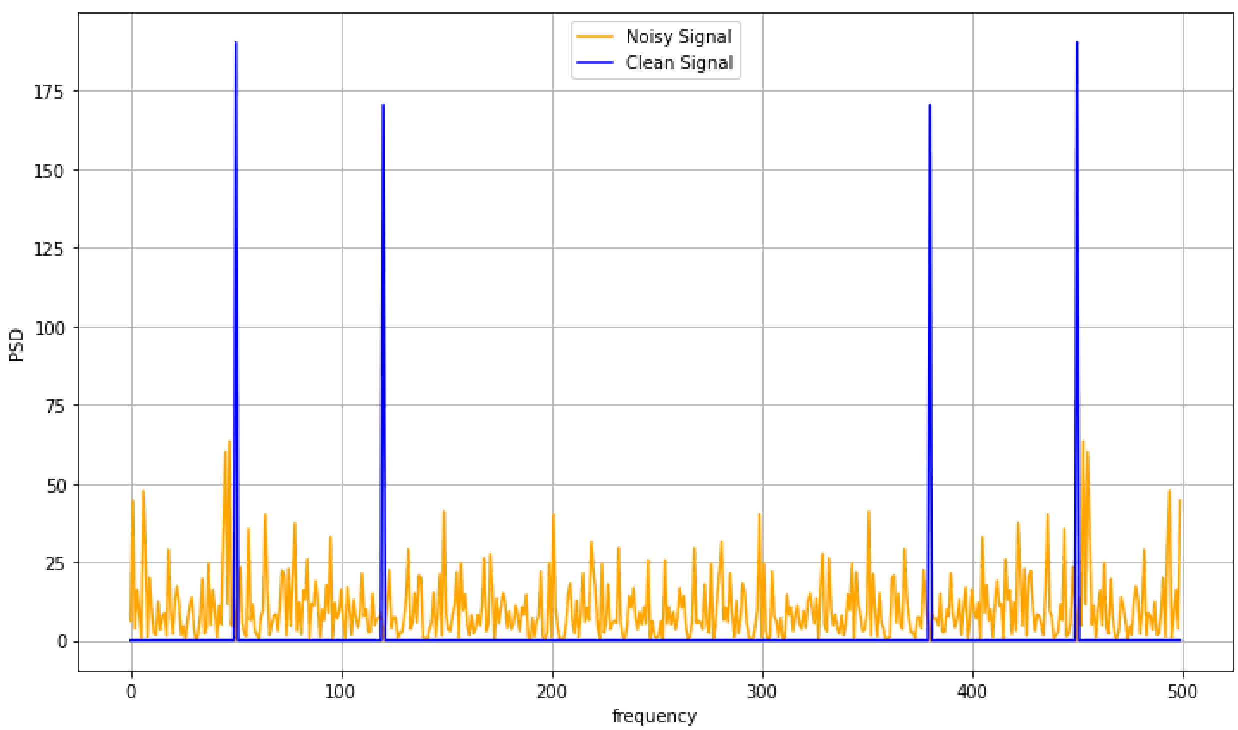


2. Computing the Fast Fourier Transform of the discretized function $y\epsilon(t)$.
3. Computing the power density spectrum (PSD) of z by multiplying it element-wise by its conjugate, and dividing it by the number of points n . After visualizing the PSD,

we get:

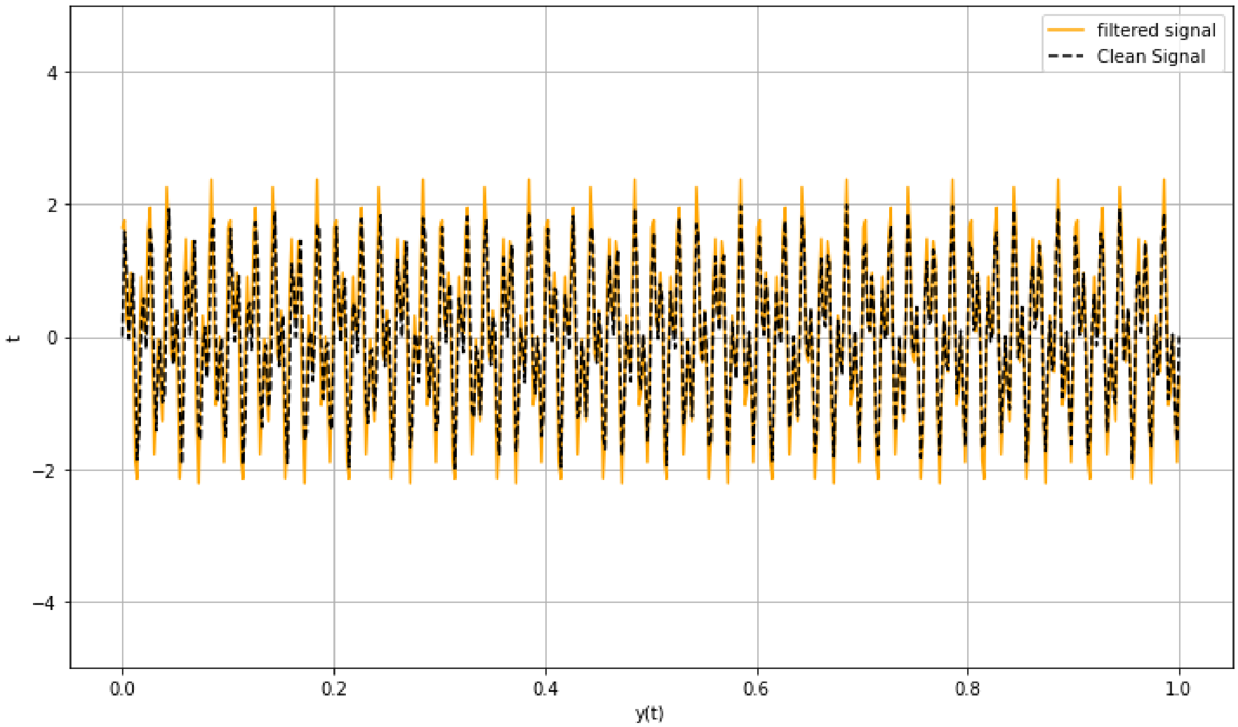


4. Based on my observations, I set a PDS-threshold $\tau = 100$ to keep just the most representative frequencies, i.e. just keep the elements in the PDS that are less than threshold



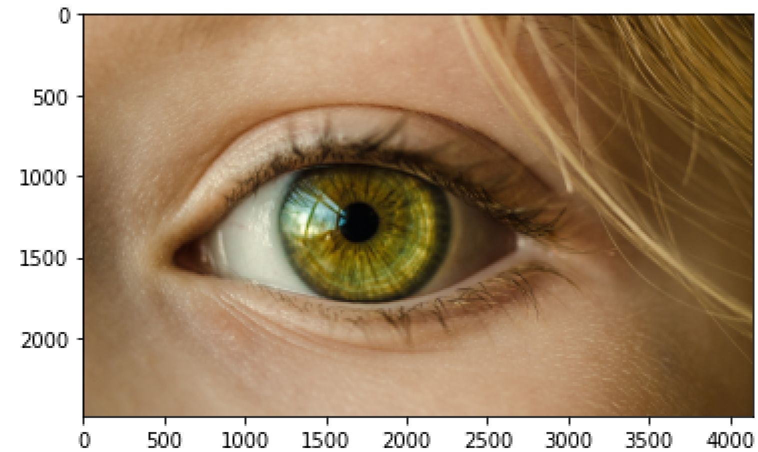
5. Returning to the discretized function $y\hat{}\epsilon(\omega)$, and using the indices as a filter. Setting to zero the elements whose index is not in our set of indices. Then computing the

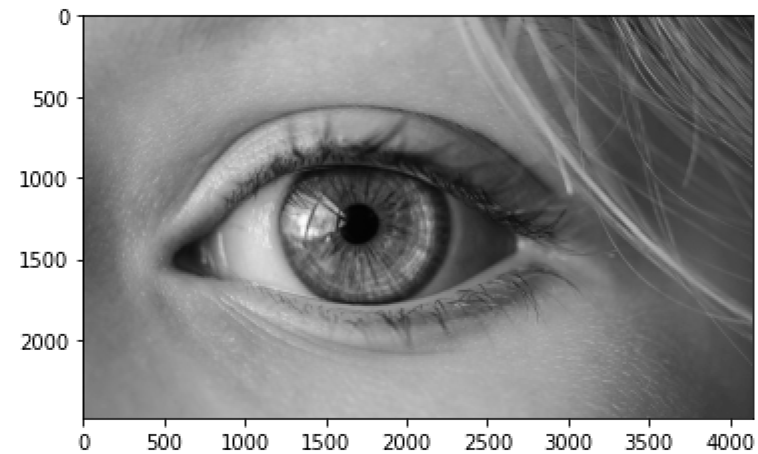
Inverse Fast Fourier Transform. This is our reconstructed function, we get:



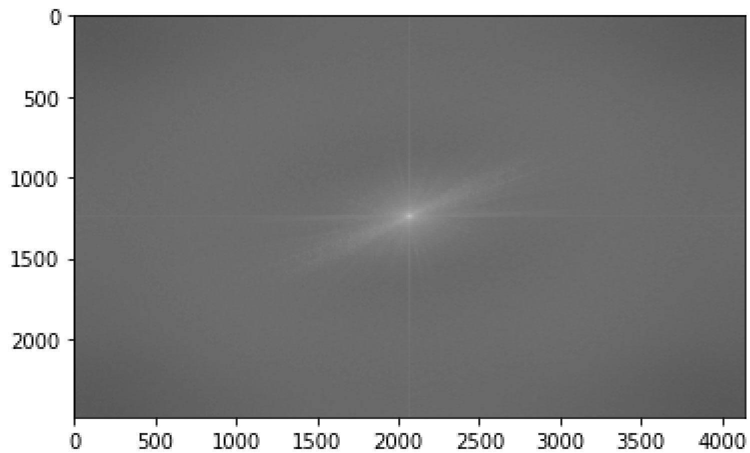
Task 2: Image Compression

1. Visualizing our image, and converting it from RGB format to a grayscale format, i.e. the dimensions of our original picture will be reduced by one dimension $A_{w \times h \times c} \rightarrow A_{w \times h}$ where *w*, *h*, and *c* stand for weight, height and channel, respectively.

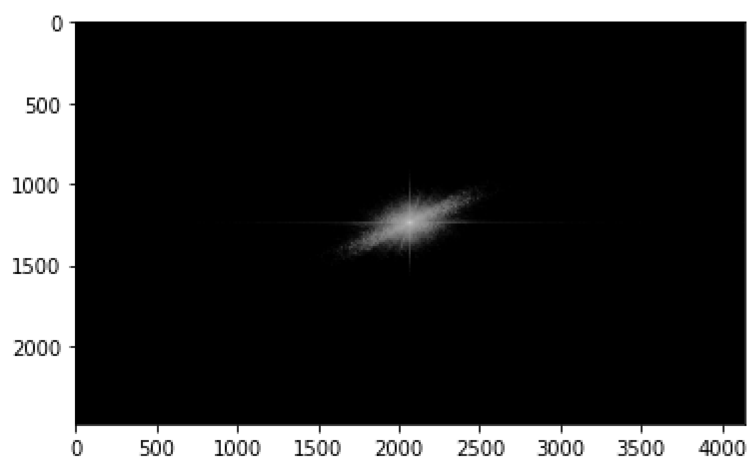




2. Computing the Fast Fourier Transform in two dimensions since we are working now with a two-dimensional array to get the spectrum of our image, then we shift the zero-frequency to the center of that spectrum.
3. Computing the magnitude a.k.a. module, absolute value, etc of the shifted spectrum to work with a homogeneous spectrum.
4. Computing the natural logarithm by adding 1 to the argument of our spectrum. This is done to avoid values close to 0 that could make the operation blow up. After visualizing the spectrum we get:

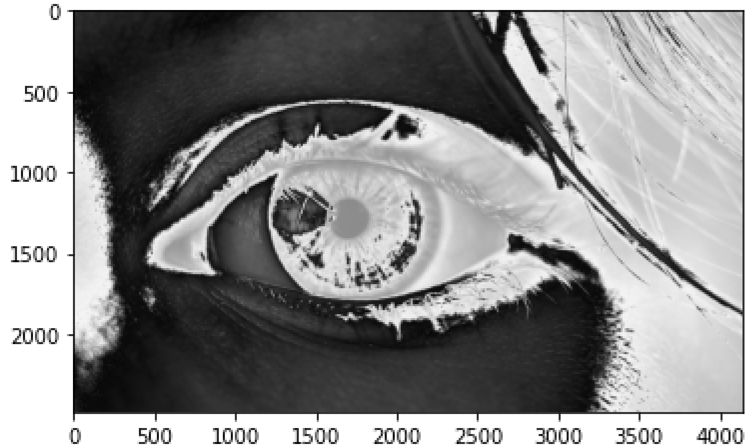


5. Reducing the dimension of our original shifted matrix in one array of one single dimension with the length being the product of the weight and the height of $F^2(A)_{w \times h}$, i.e. we are reshaping it.
6. Computing the magnitude of our reshaped array and sorting it.
7. Setting a value for $\tau = 1/100$. Keeping in mind that it should be sufficiently small, i.e. $\tau \ll 1$, multiplying it by the length of our array to keep the values that are useful for the reconstruction, then approximating it to the smaller integer.
8. Defining our threshold as that number whose index is given by *b* in our sorted array *a*. Zeroing out all small coefficients by keeping just the indices of the elements of $|F^2(A)_{wh \times 1}|$. We get:



9. Finally we return to our initial matrix $F^2(A)_{w \times h}$ and select those indices and apply the shifting, absolute value and the logarithm as before and display it.

10. As everything was done correctly, and the selected threshold was appropriate, we see our reconstructed image as follows:



Video Demonstration:

[Link]