EE200 Endsem Assignment

Yash Vardhan Pratihast
yashp24@iitk.ac.in
Roll Number:241205

February 27, 2026

1 Frequency Mixer

We know that grayscale images are a 2D signal and each pixel has a value between 0-255 representing the intensity of colour.Now in order to convert this signal into frequency domain, we'd like to see how fast the pixels change their intensity and in what direction.

Think of the pixel intensities plotting a 3D-graph and that can be decomposed into a sum of multiple sinusoidal waves in the X and Y axes.

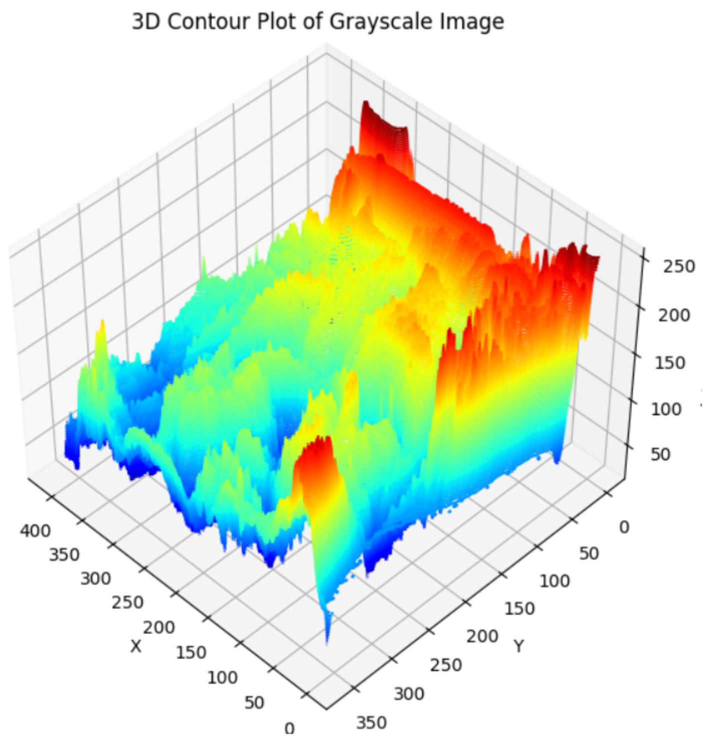


Figure 1: 3D contour of the dog image

Therefore we can apply Fourier Transform on this signal with respect to different axes and here's how we do that

1.1 2D-Fourier Transform

We know for a 1D signal we compute F.T. as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Along the same lines we compute 2D Fourier transform as follows:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

where u nd v are horizontal and vertical frequencies respectively.

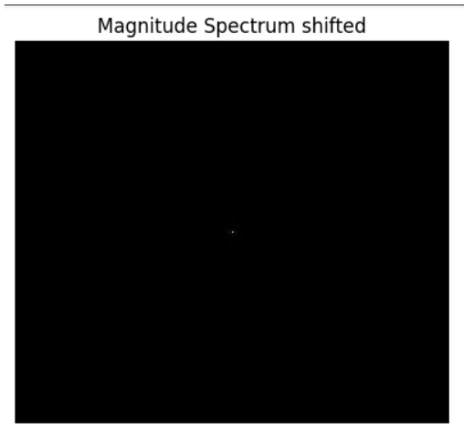
However this has a time complexity of $O(N^2)$,which is why we use Fast Fourier Transform instead with a time complexity of $O(N \log N)$

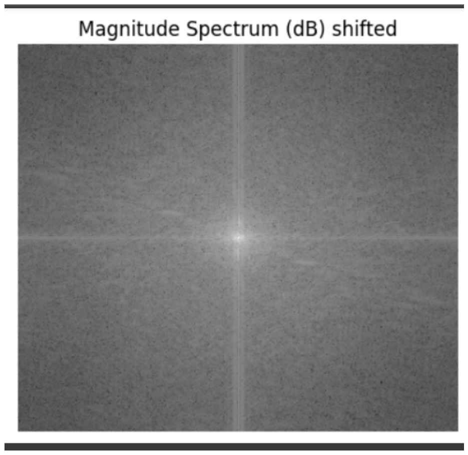
Using FFT what we get essentially is a 2D array with each entry as a complex number, say $Arr[u][v] = A + iB$ which says that there is a sinusoidal wave with horizontal frequency u, vertical frequency v, amplitude $\sqrt{(A^2 + B^2)}$ and phase angle $\tan^{-1}\left(\frac{b}{a}\right)$

By default the the lower frequencies are displayed at the boundaries of the fourier image which doesn't provide much visual data therefore in most implementations the Fourier image is shifted in such a way that the DC-value (i.e. the image mean) F(0,0) is displayed in the center of the image The further away from the center an image point is, the higher is its corresponding frequency.

Simply plotting the magnitude spectra gave us a black image with a white dot in the centre. This is because the DC-value (mean value) is by far the largest component of the image. However, the dynamic range of the Fourier coefficients (i.e. the intensity values in the Fourier image) is too large to be displayed on the screen, therefore all other values appear as black.

This can be fixed by using a logarithmic scale and in this case we use the decibel system:





The holy cross can be attributed to the fact that lower frequencies are often with more magnitude.

Now rotating the original image results in the same rotation in the fourier image as well

1.2 Fusing the two images

As the idea revolves around using high frequency portions of one image and fusing it with low frequency portions of the other, we can choose the dog image for the former one because it carries more finer details such as the wrinkles and edges while the latter can be taken from the cat image.

In order to do this we need a filter and gaussian filter is one such low pass filter

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Say we define,

Idog(x,y)=Dog Image Icat(x,y)=Cat Image G(x,y)=Gaussian Low Pass Filter '

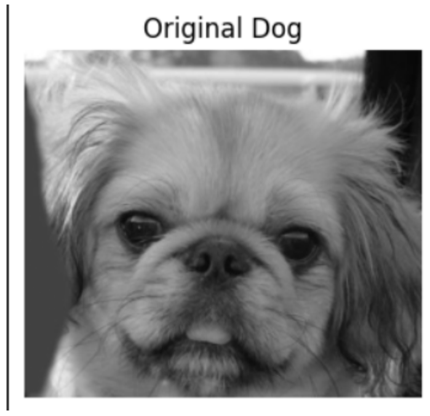
then the final output is,

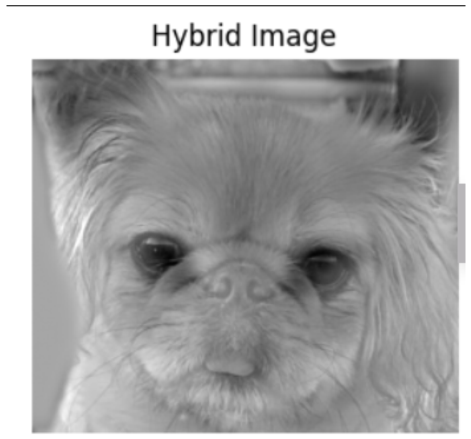
$$H(x,y) = Icat * G + Idog - Idog * G$$

essentially our transfer function is G(x,y) as defined above which has been applied on two different images which have been then combined.

The σ controls the bandwidth of the bell curve and as a result decides the cutoff frequency therefore a large σ in the spatial domain means a smaller frequency range in the frequency domain.

I have suitably chosen high pass sigma as 6 and low pass sigma as 8 to provide the following results





2 Frequency Demixer

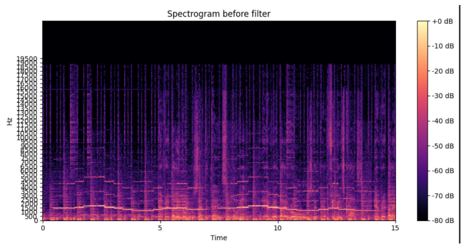
I have been given a song corrupted with piccolo sounds.Now in order to remove the piccolo sounds I would have to first the find out the frequency ranges in which the piccolo is dominant in the audio track. In order to do that I have a few tools:

2.1 Spectrogram

A spectrogram is a time-frequency representation of a signal that illustrates how its spectral content evolves over time.
It is generated by applying the Short-Time Fourier Transform (STFT) to the signal, which partitions the signal into short, overlapping windows and computes the Fourier transform for each segment. This

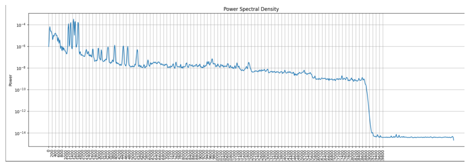
results in a complex-valued matrix whose magnitude indicates the amplitude of each frequency component at various time intervals.

When we correlate with time we see that the bright bands in the 1500 Hz range could be a strong indication of the piccolo.To further analyse we use the power spectral density graph.



2.2 Power Spectral Density

PSD tells us how the power of the signal is distributed over the frequency. Clearly in the audio the piccolo has a loud contribution and therefore shows up as dominant peaks in the graph.



As seen above there are multiple power peaks throughout the signal and these are probably from the piccolo therefore in order to attenuate them we need to design a filter.However that filter should specifically remove only the peaks and notch filters are a good choice for that

2.3 Notch Filter

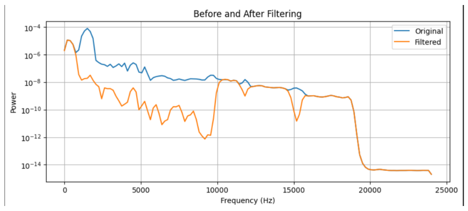
A second-order Butterworth bandstop filter was designed with a specified center frequency and bandwidth. The bandwidth was chosen as 200 Hz after looking at the width of the peaks in the PSD graph.

A Butterworth filter is an IIR (Infinite Impulse Response) filter that delivers a maximally flat frequency response in the passband.

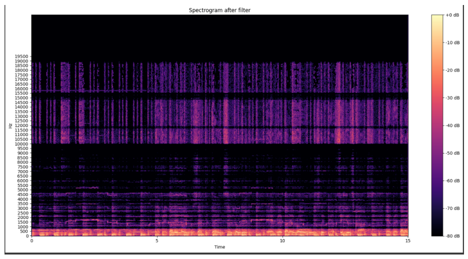
For a Butterworth filter of order n, the squared magnitude response is:

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$

In my code I have used a butterworth filter of order 2 on each of the significant peak frequencies.The final psd after attenuation of most piccolo frequencies is attached:



The final spectrogram doesn't show the yellow dense bands in the piccolo frequency region, but only a few thin bands in the overtone region which could not be removed.


Spectrogram after filter
Hz
Time