Frequency Mixer: Beauty and the Blur

EE200 End Sem Project - Q1 Report

Archisman Dhar

June 30, 2025

Abstract

This report explores how spatial frequencies play a crucial role in human visual perception. By manipulating the frequency content of images via 2D Fourier Transform and Gaussian filters, we simulate the illusion where different faces (Einstein and Monroe) appear based on viewing distance. A frequency mixer is implemented using FFT-based image fusion, separating and combining high-frequency and low-frequency components to reveal distinct perceptual layers.

1 Introduction

Our eyes process visual scenes by simultaneously interpreting various spatial frequencies. High frequencies account for edges and textures, while low frequencies provide smooth gradients and shapes. This principle is key to illusions like the Einstein-Monroe hybrid, where perception shifts based on spatial resolution and observation distance.

2 Mathematical Background

2.1 2D Discrete Fourier Transform

We know, the DFT of any 1-D signal over time domain:-

$$X(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega n}$$

The inverse transform is:

$$x[n] = \frac{1}{2\pi}\int_{\omega=-\pi}^{+\pi} X(e^{j\omega})e^{j\omega n}d\omega$$

Then, we can extend this idea to generalise the DFT for n-dimensional space as well.
2-D Discrete Fourier Transform (DFT) of an image $I(x, y)$ (which can be thought of as a 2-D aperiodic signal representation in the x-y plain in the form of a pixel matrix where each pixel value ranges from 0-255) is given by:

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

The inverse transform is:

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

2.2 Magnitude Spectrum and Logarithmic Scaling

The raw spectrum $|F(u, v)|$ typically contains very high values at low frequencies and very small values elsewhere. To make mid- and high-frequency details visible, a log or dB scale is used:

$$\text{Magnitude (dB)} = 20 \cdot \log_{10}(|F(u, v)| + \epsilon)$$

This enhances the visibility of small coefficients without distorting large ones, making the spectrum interpretable.

3 Implementation Overview

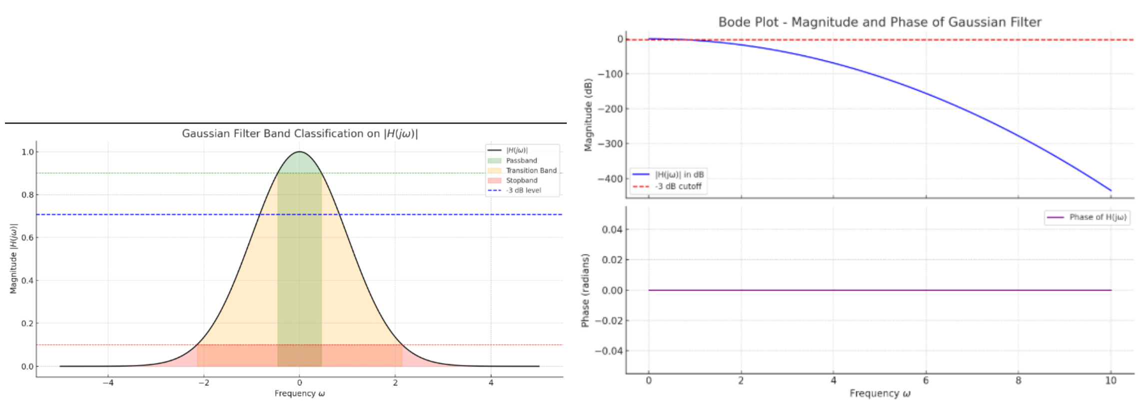
3.1 Fourier Transform and Shifting

```
# 2D FFT
fft = np.fft.fft2(image)
fft_shifted = np.fft.fftshift(fft)

# Magnitude spectra
mag_normal = np.abs(fft)
mag_db = 20 * np.log10(np.abs(fft) + 1e-5)

# Shifted spectra (low-frequency at center)
mag_normal_shifted = np.abs(fft_shifted)
mag_db_shifted = 20 * np.log10(np.abs(fft_shifted) + 1e-5)  # To avoid log(0)
```

3.2 Gaussian Filtering

The normal Gaussian filter in the 1-D frequency domain and its corresponding Bode plots (phase and magnitude) and analytical plots depicting Pass Band, Stop Band and Transition Band are shown below:-

$$H(j\omega) = e^{-\frac{\omega^2}{2\sigma^2}}$$



(a) Band frequencies of $|H(j\omega)|$

(b) Bode plots

The corresponding bandwidth (-3 dB level $\approx$ 70.7% of the peak) is around 1.65 units in the above plot.

I used Gaussian transfer functions (extended version of 1-D Gaussian function in the 2-D space) in 2-dimensional frequency domain as I am working with images:

$$H_{LP}(u,v) = e^{-\frac{u^2+v^2}{2\sigma^2}}, \quad H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

$$where, \quad u = \omega_x \quad and \quad v = \omega_y$$

Used to isolate structure/gradients (low-pass) and details/edges (high-pass).
I chose Gaussian function over any other Transfer function to be used as low pass filter because it gives smooth response in frequency domain, handles overshooting and also has isotropic character.

```
u = np.arange(-size//2, size//2)
v = np.arange(-size//2, size//2)
U, V = np.meshgrid(u, v)
D = np.sqrt(U**2 + V**2)

low_pass_filter = np.exp(-(D**2) / (2 * sigma**2))
high_pass_filter = 1 - low_pass_filter
```

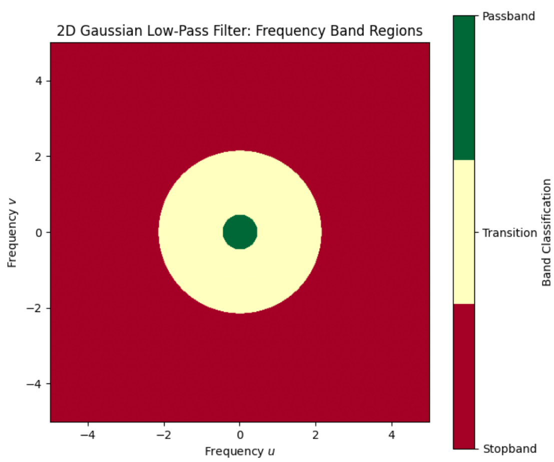


Figure 2: The frequency bands of 2-D Gaussian low pass filter

3.3 Frequency Fusion

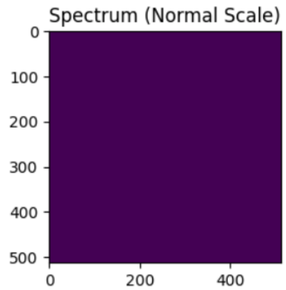
I used the GuassianBlur function imported from OpenCV module for Gaussian Filtering and then fused the filtered images to create the Hybrid image.

```
def hybrid_image(img1, img2, sigma_low=15, sigma_high=5):
    # Low-pass filter: smooth img2
    low_img2 = cv2.GaussianBlur(img2, (0, 0), sigma_low)

    # High-pass filter: detect sharp edges from img1
    low_img1 = cv2.GaussianBlur(img1, (0, 0), sigma_high)
    high_img1 = img1 - low_img1

    # Fusion of filtered images
    hybrid = cv2.addWeighted(low_img2, 0.5, high_img1, 0.5, 0)

    return hybrid, low_img2, high_img1
```

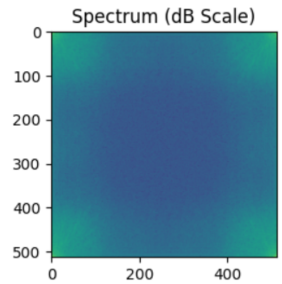
4 Spectrum Visualizations

Original and FFT Plots

(a) Original Image



(b) Magnitude (Normal)

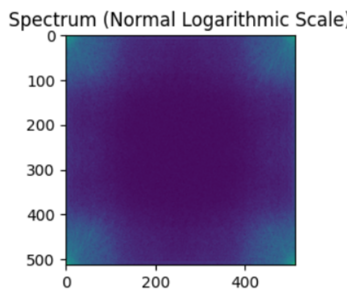


(c) Magnitude (dB)

Figure 3: 2D Fourier Transform: Original Spectrum

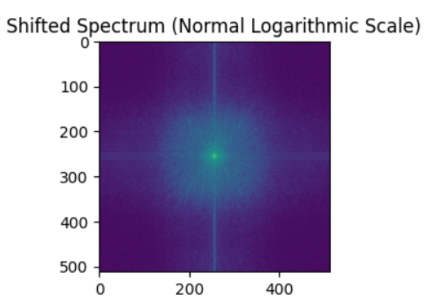
Logarithmic Scaling of Magnitude Spectrum

It is done to improve visibility and capture significance of the frequency-magnitude spectrum in normal scale.

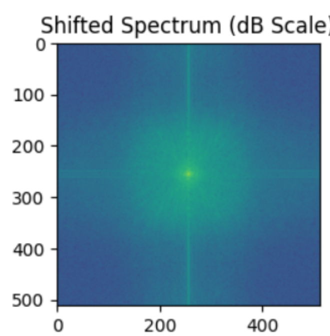


(a) Magnitude(Normal Log)

Shifted Spectrum Centered at Origin



(a) Shifted Spectrum (Log)



(b) Shifted Spectrum (dB)

Figure 5: Shifted Spectra (Low Frequency component at center)

5 Effect of Rotation

Rotating the image by 90°(anti-clockwise, in my code pipeline) results in a corresponding rotation in the frequency domain. This validates the rotational invariance property of the Fourier transform:

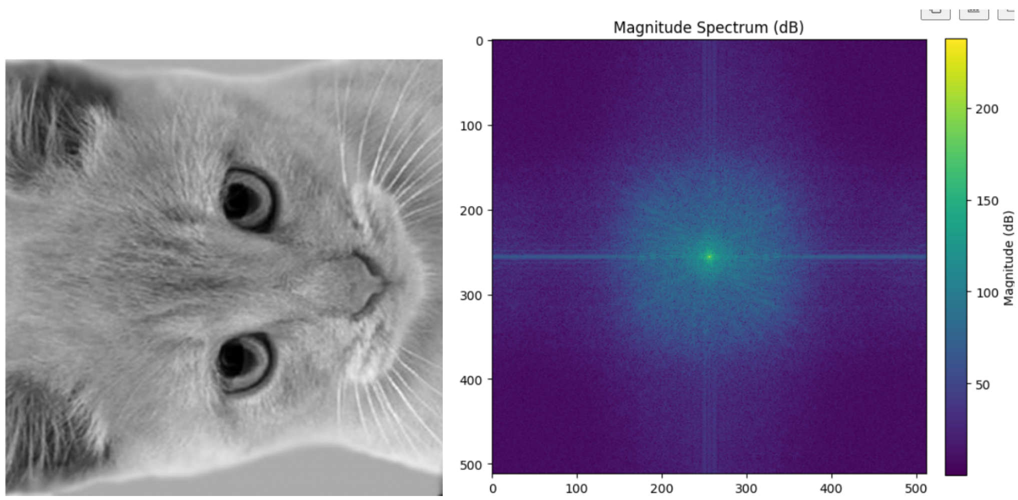


Figure 6: Rotated Image and its Spectrum

Observation: Dominant frequency structures rotate with the image, but frequency magnitudes remain preserved as rotation shifts energy diagonally in the spectrum but preserves frequency magnitude.

6 Gaussian Transfer Functions (3D Visualisation)

I generated the 3D plots of the low pass filter (Gaussian function) showing the Response Gain (peak) and of the high pass filter showing the Response Drop (depth).

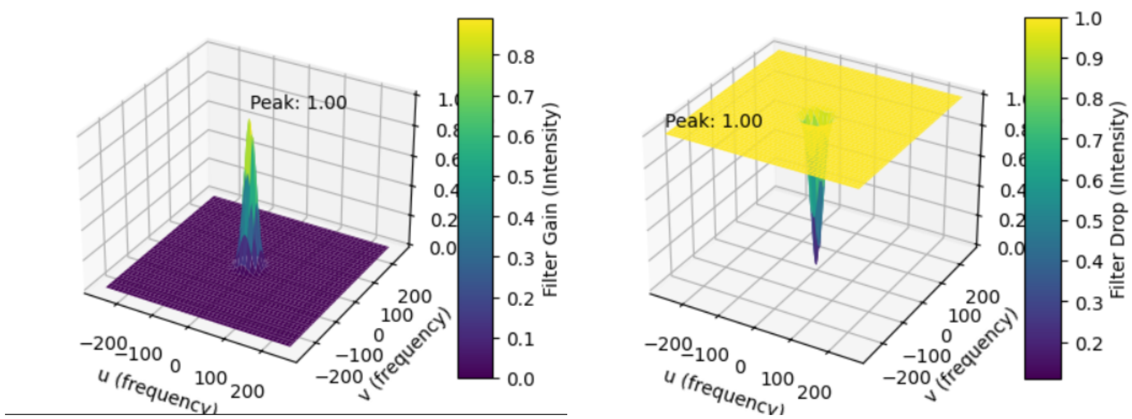


Figure 7: 3D Plots of Gaussian Low-Pass (left) and High-Pass Filters (right)

7 Frequency Mixer Output

Final fusion combines the high-pass of cat's image with low-pass of dog's image to create a perceptually hybrid image. Up close, high-frequency details dominate; from afar, the brain perceives the smoother, low-frequency structure just like the Einstein-Monroe illusion problem mentioned in the question.

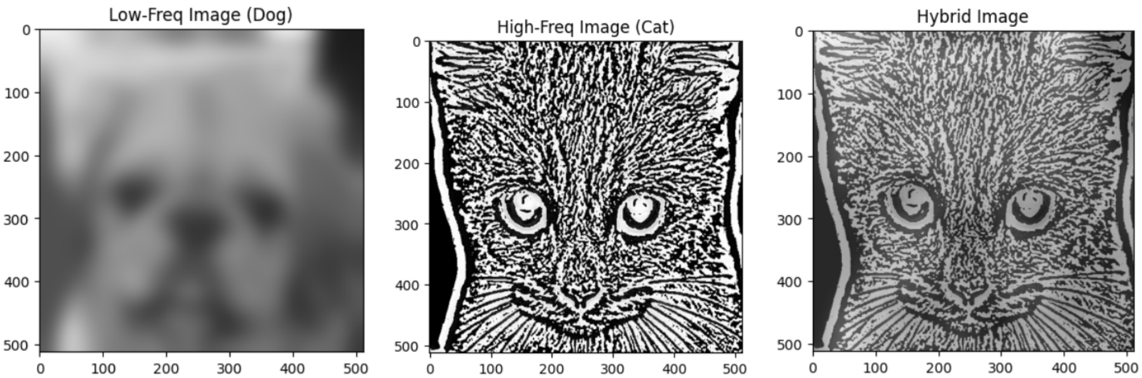


Figure 8: High Frequency filtered Dog's Image (left), High Frequency filtered Cat's Image (centre) and Frequency Mixed (Hybrid/Fused) Image (right)

8 Conclusion

I implemented a frequency-based image fusion system to explore the impact of spatial frequency on perception. The frequency mixer successfully demonstrates how high and low frequencies affect visual details, and validates theoretical properties like rotation invariance, spectrum centering, and the usefulness of log scaling. My Frequency Mixer Model successfully justifies the problem (Einstein-Monroe illusion) mentioned in the question.

9 References

- Gonzalez and Woods, *Digital Image Processing*, 4th Ed.
- A. Oliva and G. Schyns, *Hybrid Image Illusion*, OXFORD UNIVERSITY PRESS
- NumPy and OpenCV official documentation.
- DFT formula is inspired from https://www.di.univr.it/documenti/OccorrenzaIns/matdid/matdid027832.pdf