REPORT

Frequency Mixer: 'Beauty and the Blur'

Bhavit Meena
240272

July 5, 2025

1 Introduction

Our visual system instinctively uses different spatial frequencies to extract relevant details from images, with high frequencies capturing fine details like wrinkles and low frequencies preserving the overall structure. To analyze and manipulate these frequencies in images, we use the 2D Discrete Fourier Transform on the image signal $I(x, y)$.

2 Objective

The objective of this experiment is to merge two images using the 2D Fourier Transform and frequency domain filtering. By applying a low-pass filter to one image and a high-pass filter to another, we aim to create a hybrid image that demonstrates how different frequency components contribute to the overall perception of images.

We also compare the magnitude and dB spectra of the images, plot them, and rotate the images to observe and compare their Fourier Transforms with the original images.

3 Theory

An image $I(x, y)$ can be treated as a 2D discrete signal with intensity variations along both the x and y axes. To analyze its frequency components, we use the 2D Discrete Fourier Transform (DFT), which converts the spatial domain image into the frequency domain.

The 2D Discrete Fourier Transform (DFT) of an image $I(x, y)$ of size $M \times N$ is defined as

$$X(k, l) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) \cdot e^{-j2\pi\left(\frac{kx}{M} + \frac{ly}{N}\right)}$$

where:

- $I(x, y)$ is the intensity at spatial coordinates (x, y),
- $X(k, l)$ is the frequency domain representation at frequencies (k, l),

- j is the complex term.

The inverse 2D Discrete Fourier Transform, which reconstructs the image from its frequency components, is given by:

$$I(x,y) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} X(k,l) \cdot e^{j2\pi\left(\frac{kx}{M} + \frac{ly}{N}\right)}$$

These equations allow us to analyze and process images in the frequency domain for tasks such as filtering, compression, and hybrid image formation in this experiment.

A low-pass filter (LPF) in the frequency domain allows low-frequency components to pass while attenuating high frequencies, preserving the overall structure and smooth variations in an image.

A high-pass filter (HPF) retains high-frequency components while suppressing low frequencies, emphasizing fine details and edges.

dB Visualization of Spectra: The magnitude of the Fourier Transform often contains a wide dynamic range, making it difficult to visualize details using linear scaling. To address this, a logarithmic scale in decibels (dB) is used for visualization, given by:

$$M_{dB}(k,l) = 20 \cdot \log_{10}(|X(k,l)|)$$

where $M_{dB}(k,l)$ represents the magnitude spectrum in decibels and $|X(k,l)|$ is the magnitude of the Fourier Transform at frequency indices k, l. This scaling compresses the range and allows both low and high magnitude components to be visible simultaneously, aiding in effective frequency domain analysis.

4 Implementation

This section explains the practical steps followed to analyze, process, and combine two images in the frequency domain using Python. The implementation is based on the use of the 2D Discrete Fourier Transform (DFT), frequency filters, and inverse transformations. Each step is described in detail below.

1. **Importing and Preprocessing Images:** The input images were loaded using the PIL (Python Imaging Library). Since frequency domain analysis works best on grayscale images, the images were converted to grayscale using `.convert('L')`.
   Grayscale conversion reduces computation and isolates structural intensity features by removing color. [FIG.1]

2. **Normalization:** Pixel values were normalized to the range $[0,1]$ by dividing by 255. This ensures that the Fourier transform returns values in a consistent numerical range, improving numerical stability when combining images later.

3. **Applying 2D Fourier Transform:** The 2D DFT was computed using `np.fft.fft2`.
   This transformed each image from the spatial domain to the frequency domain. The result is a complex-valued array where the magnitude represents how much of a particular frequency is present in the image and the phase encodes orientation and positioning.

4. **Magnitude and dB Spectrum Visualization:** To better visualize the frequency content of each image, the magnitude of the Fourier Transform was computed using `np.abs()`, and the logarithmic dB spectrum was calculated as:

$$M_{dB}(k, l) = 20 \cdot \log_{10}(|X(k, l)|)$$

   This helps highlight frequency details that are otherwise too faint to notice in the raw spectrum. The dB scale compresses dynamic range, enabling both strong and weak frequencies to be visible in the same plot.[FIG.2]

5. **Shifting Low Frequencies to the Center:** By default, the zero-frequency (DC) component of the Fourier Transform, which represents the average brightness of the image, is located at the corners of the frequency domain output. To make interpretation easier, we used `np.fft.fftshift` to rearrange the low frequencies to the center of the spectrum, and the high frequencies move to the corners.

   This centralization is important because:

   - It allows us to easily design and apply frequency filters (like Gaussian LPF and HPF) symmetrically around the origin.
   - It makes the visualization of the frequency content more intuitive, as the important low-frequency information is visually centralized.
   - It helps in understanding how frequency components are distributed spatially, which is critical while analyzing or modifying images in the frequency domain.

6. **Analysis of Translation in Frequency Domain:**

   Upon translating the cat image in the spatial domain, the 2D Fourier Transform was computed to analyze its frequency characteristics. It was observed that the magnitude spectrum of the translated image remained visually identical to that of the original image. This confirms a fundamental property of the Fourier Transform: *translation (shifting) in the spatial domain affects only the phase component of the frequency representation while leaving the magnitude spectrum unchanged.*[FIG.3]

7. **Image Rotation and Analysis:** To test the frequency domain's response to spatial rotation, In the practical the cat image was rotated 90° anticlockwise using `.rotate(90)`.The rotated image's FFT magnitude

spectrum was compared to the original, and as expected, the frequency domain representation also rotated by 90°, confirming the property that spatial rotation causes equivalent rotation in the frequency domain.We use $\log(1 + M)$ when visualizing the magnitude spectra to compress the dynamic range for clearer interpretation while avoiding the undefined condition of $\log(0)$, ensuring that both low and high frequency components are visible in the plotted spectra without distorting relative differences. [FIG.4]

8. **Gaussian Filter Design (LPF and HPF):** A Gaussian low-pass filter (LPF) was created using:

$$G_{LP}(u, v) = \exp\left(-\frac{(u - u_0)^2 + (v - v_0)^2}{2\sigma^2}\right)$$

where (u_0, v_0) is the center of the frequency grid and σ controls the bandwidth. The high-pass filter (HPF) was derived as $G_{HP} = 1 - G_{LP}$. The LPF preserves smooth structures and removes sharp transitions, while HPF retains edges and fine details.

9. **Filtering the Images in Frequency Domain:** The frequency representation of the cat image was multiplied element-wise with the LPF, keeping its low-frequency components. Similarly, the dog image's frequency data was multiplied with the HPF to retain its fine details. This separation was essential for building a perceptually meaningful hybrid image.

10. **Combining the Filtered Images:** The filtered frequency representations were added together. Optionally, weights (e.g., α, β) could be used to control the dominance of each image:

$$H(u, v) = \alpha \cdot \mathrm{Cat}_{LPF}(u, v) + \beta \cdot \mathrm{Dog}_{HPF}(u, v)$$

In notebook, equal weights were used after normalization.

11. **Inverse Fourier Transform and Image Reconstruction:** The combined frequency domain image was inverse-shifted using `np.fft.ifftshift` and transformed back to the spatial domain using `np.fft.ifft2`. The result was a complex array, and only the real part was used to display the final hybrid image.

5 Observations

The following key observations were made during the practical implementation:

- The magnitude and dB spectra of the input images showed that low-frequency components are densely concentrated near the center (after shifting) and represent the overall smooth structure of the image. The high-frequency components are distributed outward and represent fine details and edges.
- When the cat image was rotated by 90°, its Fourier Transform magnitude spectrum also rotated by 90°, demonstrating that rotation in the spatial domain leads to a corresponding rotation in the frequency domain. This matches the theoretical prediction.
- A spatial translation was applied to the cat image to study its effect on the frequency domain. It was observed that the magnitude spectrum remained visually identical to that of the original image, confirming that spatial shifts affect only the phase spectrum while leaving the magnitude unchanged.
- The designed low-pass and high-pass Gaussian filters successfully isolated the desired frequency regions: the cat image contributed low-frequency content (smooth structure) and the dog image contributed high-frequency details (edges and fine texture).
- The final hybrid image clearly demonstrated the expected behavior:
  - When viewed from a short distance, the high-frequency details (dog features) dominated the perception.
  - When viewed from far or with squinted eyes, the low-frequency structure (cat features) became more prominent.

  IMAGES

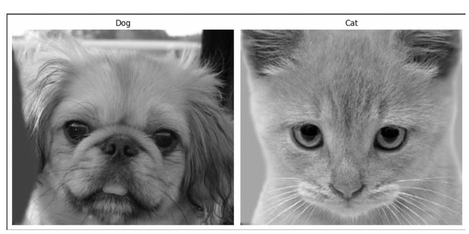

Figure 1: GRAYSCALE IMAGES

H

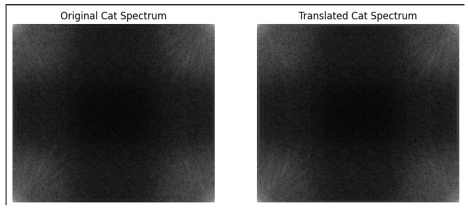

Figure 3: Translation

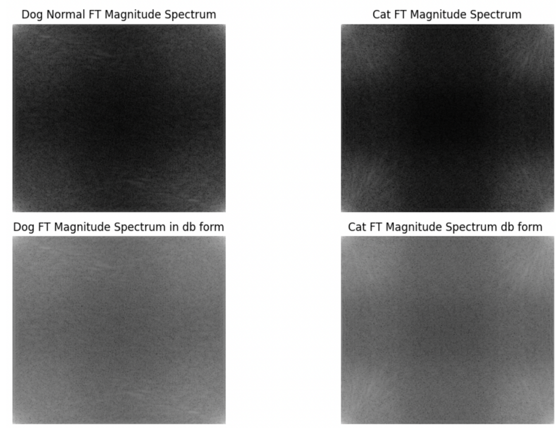

Figure 2: MAGNITUDE AND DB SPECTRA

[b]0.45

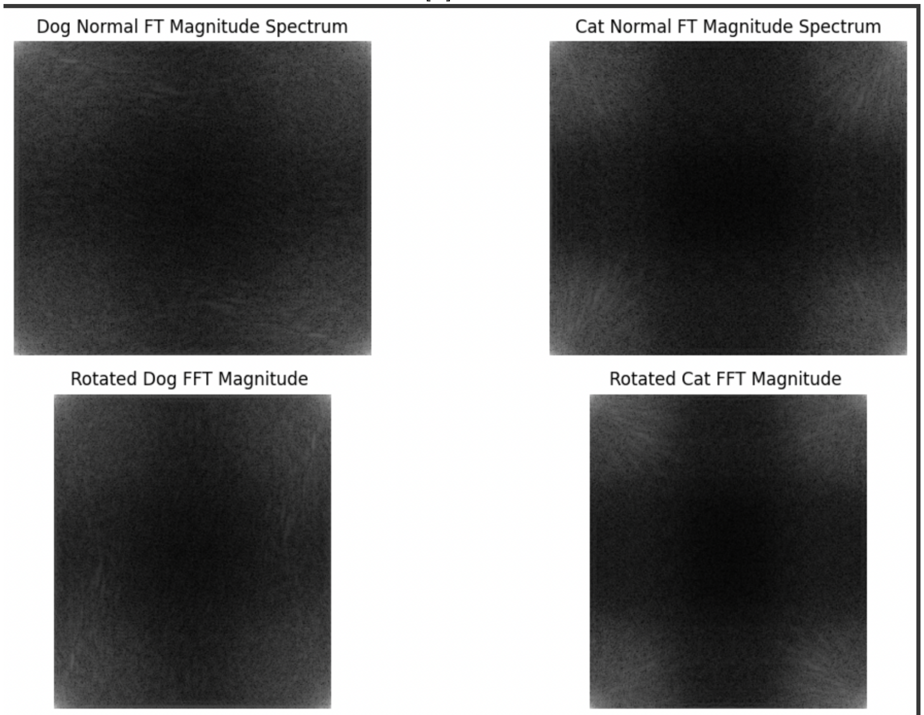

Figure 4: Comparison of Rotated vs Original Spectra

[b]0.45

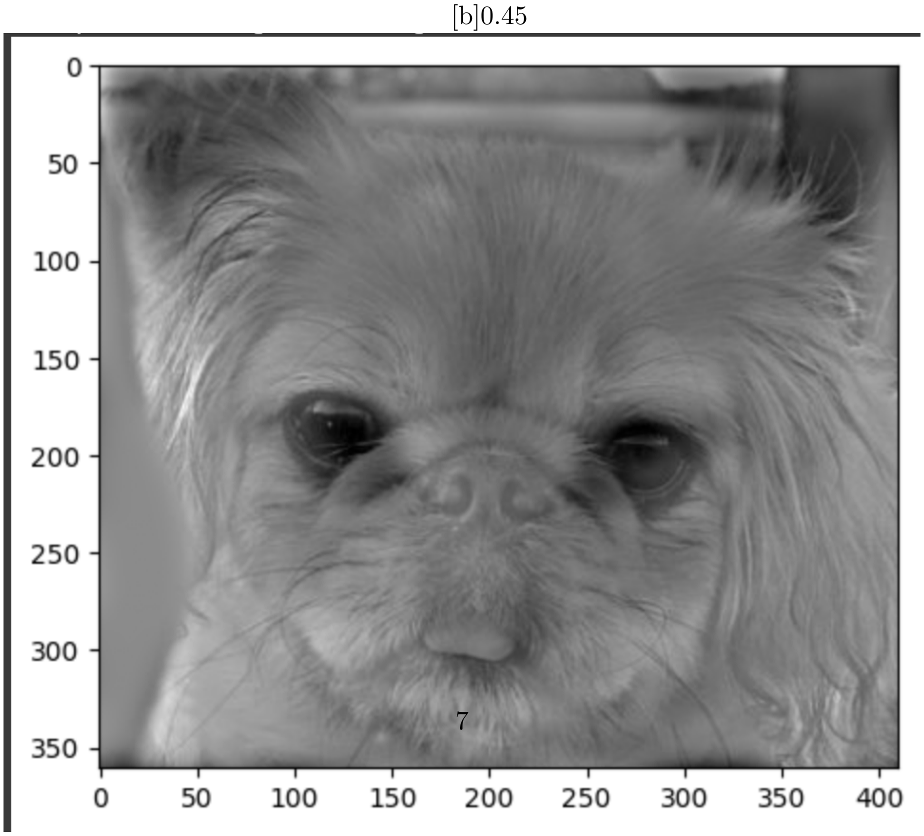

Figure 5: FINAL IMAGE

6 What If?

In this section, we explore alternative methods, design choices, and their potential effects on the frequency mixing practical.

6.1 Using Ideal Filters Instead of Gaussian Filters

In our practical, we used Gaussian filters due to their smooth roll-off and reduced ringing artifacts in the spatial domain. However, if we had used ideal low-pass and high-pass filters with sharp cutoffs in the frequency domain, we could have achieved stricter frequency separation. The drawback of using ideal filters is the introduction of ringing artifacts in the spatial domain (Gibbs phenomenon), which would have caused visible ripples around sharp edges in the reconstructed image, reducing visual quality.

6.2 Changing the Bandwidth (σsigma) of Filters

The value of σ in Gaussian filters directly affects the cutoff bandwidth. Using a smaller σ (ex σ ¡ 8)for the low-pass filter would pass fewer high frequencies, resulting in a blurrier image, while a larger σ(ex σ ¿ 10) would retain more details. Similarly, adjusting σ in the high-pass filter would change the amount of fine details from the second image that appear in the combined image. Experimenting with different σ values , we got $\sigma = 9$ almost good for our practical.

6.3 Exploring Color Image Frequency Mixing

Our practical used grayscale images to focus on spatial frequencies, WHAT If we had used RGB color images, we would need to apply frequency domain filtering separately to each channel, which could lead to color artifacts if not handled carefully. Exploring color image frequency mixing could extend the experiment to practical applications in artistic image blending and texture transfer.

6.4 Limitations of the Current Method

While the frequency mixer effectively demonstrates how low and high frequencies contribute to image perception, it depends on precise alignment and size matching of the images in the frequency domain. Additionally, the method does not handle misalignment, occlusions, or non-uniform illumination well, which could affect real-world applications. It also requires careful selection of filter parameters to balance between clarity and smoothness in the output.

Exploring these alternatives would enhance understanding of the frequency domain's role in image processing and improve the flexibility of the frequency mixing technique for various applications.

7 Conclusion

In this experiment, we successfully implemented a frequency mixer using the 2D Fourier Transform to combine the low-frequency content of one image with the high-frequency details of another. Through plotting the magnitude and dB spectra, applying Gaussian filters, and observing the effects of translation and rotation, we validated key frequency domain properties practically. The final hybrid image demonstrated the desired behavior of revealing different features at varying viewing distances, illustrating the power of frequency-based image analysis and processing in