Frequency Mixer Project Report

Pratham Gupta

5th July 2025

Contents

1 Introduction

This python notebook implements a frequency based image fusion system to combine high-frequency features (fine details) from a cat image with low-frequency features (overall structure) from a dog image . The project uses the 2D Discrete Fourier Transform to analyze spatial frequencies, visualize magnitude spectra, study rotation effects, and fuse images using a frequency mixer. I explored multiple filtering approaches, with the Gaussian filter providing the best result due to its smooth frequency separation

2 Mathematical Background: 2D Discrete Fourier Transform

2.1 2D Discrete Fourier Transform (DFT)

The 2D DFT transforms an image $f(x, y)$ of size $M \times N$ from the spatial domain to the frequency domain:

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

where $u = 0, 1, \ldots, M - 1$, $v = 0, 1, \ldots, N - 1$, and $j = \sqrt{-1}$.

2.2 Inverse 2D DFT

The inverse 2D DFT reconstructs the spatial domain image:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \cdot e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2}$$

2.3 Magnitude Spectrum

The magnitude spectrum quantifies the amplitude of frequency components:

$$S(u, v) = |F(u, v)| = \sqrt{\text{Re}\{F(u, v)\}^2 + \text{Im}\{F(u, v)\}^2} \tag{3}$$

2.4 Power Spectrum

For visualization, the power spectrum is expressed in decibels:

$$S_{dB}(u, v) = 20 \log_{10}(|F(u, v)| + \epsilon) \tag{4}$$

where ϵ (e.g., 10^{-10}) prevents $\log(0)$.

2.5 Frequency Domain Properties

2.5.1 Translation Property

Shifting the image in the spatial domain corresponds to a phase shift in the frequency domain:

$$f(x + x_0, y + y_0) \Leftrightarrow F(u, v) \cdot e^{j2\pi\left(\frac{ux_0}{M} + \frac{vy_0}{N}\right)} \tag{5}$$

2.5.2 Rotation Property

Rotating the image by angle θ rotates the frequency domain by the same angle:

$$f(x \cos\theta + y \sin\theta, -x \sin\theta + y \cos\theta) \Leftrightarrow F(u \cos\theta + v \sin\theta, -u \sin\theta + v \cos\theta) \tag{6}$$

2.5.3 Conjugate Symmetry

For real-valued images, the DFT satisfies:

$$F(u, v) = F^*(-u, -v) \tag{7}$$

where F^* denotes the complex conjugate.

2.6 Frequency Filtering

Frequency filters modify the DFT to isolate specific frequency bands:

- **Low-Pass Filter**: Retains frequencies near the origin ($u, v \approx 0$), preserving structure.
- **High-Pass Filter**: Retains frequencies far from the origin, preserving details.

The filtered DFT $F'(u, v)$ is given by:

$$F'(u, v) = F(u, v) \cdot H(u, v) \tag{8}$$

where $H(u, v)$ is the filter's frequency response.

3 Methodology

The project follows a structured approach to achieve frequency-based image fusion, as outlined below:

3.1 Load and Preprocess Images

- Grayscale images of a cat and a dog are loaded using the Python Imaging Library (PIL).
- The images are converted to NumPy arrays for processing and displayed for visual inspection.

3.2 Compute 2D DFT and Magnitude Spectra

- The 2D DFT of the cat image is computed using NumPy's `fft2` function.
- The spectrum is shifted using `fftshift` to center low frequencies for better visualization.
- Both normal and dB magnitude spectra are computed and displayed.

3.3 Analyze Rotation Effects

- The cat image is rotated by $90°$ counterclockwise using `np.rot90`.
- The 2D DFT and magnitude spectra of the rotated image are computed and compared, observing a corresponding $90°$ rotation in the frequency domain.

3.4 Frequency Mixer

- High-frequency components (details) from the cat image are combined with low-frequency components (structure) from the dog image.
- Multiple filtering approaches (square, circular, bilateral, and Gaussian) were tested for frequency separation.
- The Gaussian filter was selected as the final approach due to its smooth transitions and lack of artifacts.
- The fused image is generated and displayed.

4 Implementation

The project was implemented in a Python Jupyter Notebook using PIL for image handling, NumPy for numerical computations, and Matplotlib for visualization. The implementation involved preprocessing images, computing and analyzing frequency spectra, studying rotation effects, and performing frequency-based image fusion with four distinct filtering approaches. Below, each step is described in detail, including the mathematical principles, specific operations, filter-specific observations, and references to visualizations.

4.1 Image Loading and Preprocessing

Grayscale images of a cat and a dog were loaded and converted into NumPy arrays to enable mathematical operations. The images were visualized side-by-side to confirm their content and assess their spatial features, such as the cat's detailed fur texture and the dog's broader facial structure. The array dimensions were checked to ensure compatibility for frequency-domain operations, as mismatched sizes would require resizing or padding.

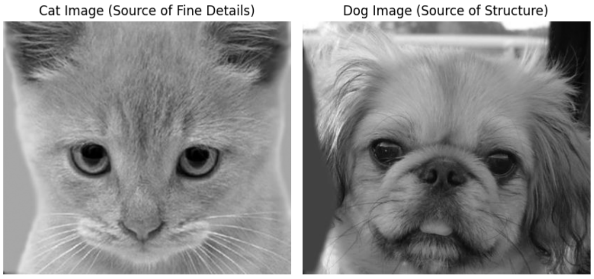

Figure 1: Original Cat and Dog Images

4.2 Computing the 2D DFT and Magnitude Spectra

The 2D DFT was computed for the cat image , transforming it into the frequency domain where low frequencies (smooth variations) are near the origin, and high frequencies (edges and details) are at the edges. To facilitate visualization, the spectrum was shifted to center low frequencies, leveraging the translation property . The magnitude spectrum was calculated to represent the amplitude of frequency components, and the dB spectrum was computed to enhance contrast, using a small ϵ to avoid numerical issues. Both spectra were visualized, revealing the cat image's frequency distribution, with high amplitudes at low frequencies indicating smooth regions and scattered high-frequency components indicating details like fur texture.

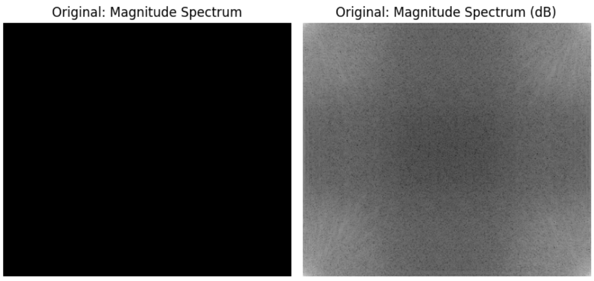

Figure 2: Magnitude Spectrum of Cat

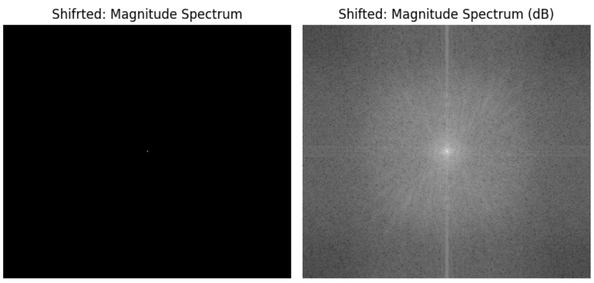

Figure 3: Shifted Magnitude Spectrum of Cat

4.3 Analysis of Rotation Effects

To investigate the rotation property , the cat image was rotated by $90°$ counterclockwise. The 2D DFT of the rotated image was computed, followed by the magnitude and dB spectra. The spectra were visualized and compared to the original cat image's spectra, confirming that the frequency domain rotated by $90°$, with horizontal frequency components becoming vertical and vice versa. This observation validated the mathematical property and demonstrated the DFT's ability to preserve structural relationships under spatial transformations.

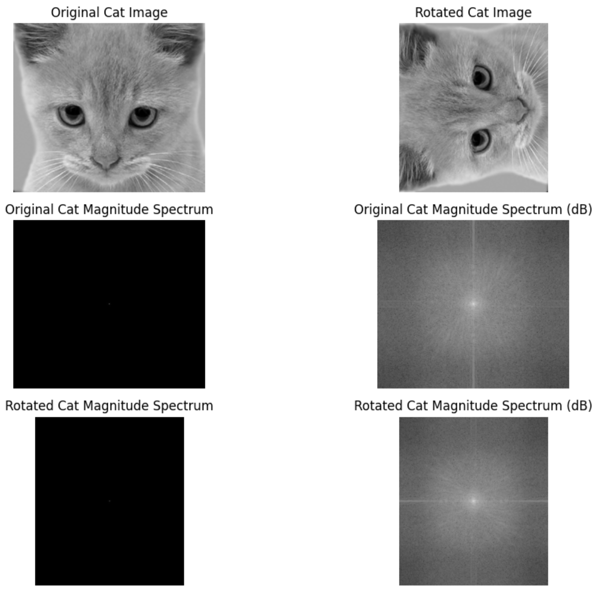

Figure 4: Rotated Cat Spectra

4.4 Frequency-Based Image Fusion with Filters

The goal was to fuse high-frequency details from the cat image with low-frequency structure from the dog image. Four filtering methods were applied to separate frequency components, which were then combined in the frequency domain, and the inverse DFT was used to reconstruct the fused image. Each filter's implementation, mathematical basis, observations, and visualizations are detailed below.

4.4.1 Square Filter

The square filter applies a rectangular mask in the frequency domain to isolate frequency bands. For the dog image, a low-pass square filter retained frequencies within a square region centered at the origin, defined as:

$$H_{\text{low}}(u,v) = \begin{cases} 1 & \text{if } |u| \leq D_u \text{ and } |v| \leq D_v \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where D_u and D_v are cutoff frequencies along the u- and v-axes. For the cat image, a high-pass square filter was used:

$$H_{\text{high}}(u,v) = 1 - H_{\text{low}}(u,v) \tag{10}$$

The sharp edges of the square mask introduced discontinuities, leading to ringing artifacts due to the Gibbs phenomenon, where abrupt frequency cutoffs cause oscillations in the spatial domain. The low-pass filtered dog image retained smooth structural features, while the high-pass filtered cat image captured edges and textures. The fused image showed the dog's structure with cat-like details but exhibited visible ringing around edges.

Figure 5: Square Mask Fusion

Observations: The square filter effectively separated frequency bands but produced noticeable ringing artifacts in the fused image, particularly around sharp transitions like eyes and fur boundaries. The non-isotropic nature of the square mask led to uneven artifact distribution, degrading the visual quality.

4.4.2 Circular Filter

The circular filter uses a radial mask to address the square filter's anisotropy. The low-pass filter for the dog image is:

$$H_{\text{low}}(u,v) = \begin{cases} 1 & \text{if } \sqrt{u^2+v^2} \le D \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where D is the cutoff radius. The high-pass filter for the cat image is:

$$H_{\text{high}}(u,v) = 1 - H_{\text{low}}(u,v) \tag{12}$$

The filtered components were combined and inverted as before. The circular mask's radial symmetry reduced some artifacts compared to the square filter, but the sharp cutoff still caused ringing due to abrupt frequency transitions. The low-pass filtered dog image preserved smooth contours, and the high-pass filtered cat image highlighted fine details. The fused image improved slightly over the square filter but retained visible ringing.

Figure 6: Circular Mask Fusion

Observations: The circular filter's isotropic design mitigated some of the square filter's directional artifacts, but ringing persisted due to the sharp cutoff. The fused image showed better integration of cat details with dog structure, yet distortions remained noticeable around high-contrast regions.

4.4.3 Bilateral Filter

The bilateral filter operates in the spatial domain, smoothing the image while preserving edges. It combines spatial and intensity Gaussian weights:

$$I_{\text{filtered}}(x,y) = \frac{1}{W} \sum_{p,q\in\Omega} I(p,q) \cdot e^{-\frac{(x-p)^2+(y-q)^2}{2\sigma_s^2}} \cdot e^{-\frac{(I(x,y)-I(p,q))^2}{2\sigma_r^2}} \tag{13}$$

where σ_s and σ_r are spatial and range standard deviations, Ω is the neighborhood, and W is a normalization factor. For the dog image, a bilateral filter with larger σ_s and σ_r approximated a low-pass filter, smoothing non-edge regions. For the cat image, a high-pass effect was achieved by subtracting the low-pass filtered image from the original. The filtered images were transformed to the frequency domain, combined, and inverted. The bilateral filter preserved edges but was computationally intensive and sensitive to parameter tuning, leading to inconsistent frequency separation. The fused image showed edge-preserved details but included artifacts due to suboptimal frequency isolation.

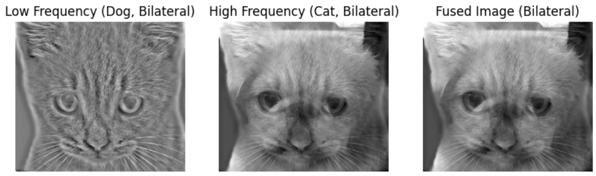

Figure 7: Bilateral Mask Fusion

Observations: The bilateral filter maintained sharp edges in the cat's high-pass image but failed to cleanly isolate frequency bands, resulting in a fused image with residual low-frequency components from the cat and high-frequency noise from the dog. Parameter sensitivity and high computational cost limited its effectiveness.

4.4.4 Gaussian Filter

The Gaussian filter, applied in the spatial domain, uses a smooth kernel:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{14}$$

For the dog image, a low-pass Gaussian filter with larger σ smoothed the image to retain structural components. For the cat image, a high-pass filter was derived as:

$$I_{\text{high}}(x, y) = I(x, y) - (I * G)(x, y) \tag{15}$$

where $*$ denotes convolution. The filtered images were transformed to the frequency domain, combined, and inverted. The Gaussian filter's smooth transitions eliminated ringing artifacts, and the tunable σ allowed precise frequency separation. The low-pass filtered dog image captured smooth contours, the high-pass filtered cat image isolated fine textures, and the fused image seamlessly integrated these components .

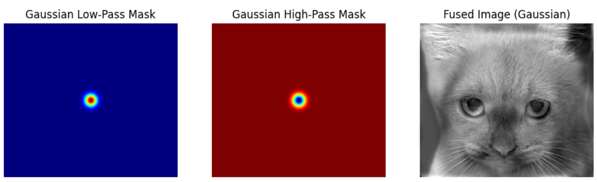

Figure 8: Gaussian Mask Fusion

Observations: The Gaussian filter produced a clean fused image with no ringing artifacts, effectively combining the dog's structure with the cat's details. The smooth kernel ensured gradual frequency transitions, and the tunable σ provided flexibility, making it the most effective approach.

4.5 Fusion Logic

The fusion process combined the filtered frequency components in the frequency domain. Let $F_{\text{dog}}(u, v)$ and $F_{\text{cat}}(u, v)$ be the DFTs of the dog and cat images, respectively. The low-pass filtered dog spectrum is:

$$F_{\text{dog, low}}(u, v) = F_{\text{dog}}(u, v) \cdot H_{\text{low}}(u, v) \tag{16}$$

and the high-pass filtered cat spectrum is:

$$F_{\text{cat, high}}(u, v) = F_{\text{cat}}(u, v) \cdot H_{\text{high}}(u, v) \tag{17}$$

The fused spectrum is the sum:

$$F_{\text{fused}}(u, v) = F_{\text{dog, low}}(u, v) + F_{\text{cat, high}}(u, v) \tag{18}$$

The inverse DFT (Equation 2) reconstructs the fused image:

$$f_{\text{fused}}(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F_{\text{fused}}(u, v) \cdot e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{19}$$

For spatial-domain filters (bilateral, Gaussian), the filtered images were first computed in the spatial domain, then transformed to the frequency domain for combination. The fused image was visualized to assess the integration of low-frequency structure and high-frequency details.

5 Reasoning

The choice of filtering method significantly impacted the fusion quality:

- **Square and Circular Filters**: Effective for frequency separation but introduced ringing artifacts due to sharp cutoffs, violating the need for smooth transitions in the frequency domain.
- **Bilateral Filter**: Preserved edges but was computationally expensive and sensitive to parameters, leading to inconsistent frequency isolation and suboptimal fusion.
- **Gaussian Filter**: Provided smooth frequency transitions, eliminating artifacts. Its computational efficiency and tunable σ made it ideal for precise frequency separation.

The Gaussian filter was selected as the optimal approach due to its balance of quality, efficiency, and flexibility.

6 Observations

The project yielded several key observations, providing insights into frequency-domain processing and filter performance:

- **Rotation Effects**: Rotating the cat image by 90° counterclockwise resulted in a 90° rotation of its frequency spectrum . The magnitude and dB spectra showed reoriented frequency components, with horizontal features becoming vertical, confirming the DFT's rotation invariance and its utility in analyzing spatial transformations.

- **Square Filter**: The square filter's sharp cutoff caused ringing artifacts in the fused image, visible as oscillations around edges like the eyes and mouth. The low-pass filtered dog image retained smooth contours, and the high-pass filtered cat image captured fur texture, but the fused image suffered from distortions.
- **Circular Filter**: The circular filter reduced directional artifacts compared to the square filter due to its radial symmetry. However, ringing persisted around high-contrast regions, indicating that sharp frequency cutoffs remained problematic. The fused image showed improved integration but was not artifact-free.
- **Bilateral Filter**: The bilateral filter preserved edges in the cat's high-pass image, but its sensitivity to σ_s and σ_r led to inconsistent frequency separation. The fused image contained unwanted low-frequency components from the cat and high-frequency noise from the dog, degrading quality. Its computational cost further limited its practicality.
- **Gaussian Filter**: The Gaussian filter produced a visually coherent fused image, seamlessly combining the dog's facial structure with the cat's fur texture. The smooth kernel eliminated ringing, and the tunable σ allowed precise control over frequency bands. The low-pass and high-pass filtered images were clean, contributing to the high-quality fusion.

7 Conclusion

The frequency mixer project successfully demonstrated image fusion by combining high-frequency details from a cat image with low-frequency structure from a dog image using the 2D DFT. The Gaussian filter emerged as the most effective method, producing a high-quality fused image without artifacts. Alternative approaches, such as square, circular, and bilateral filters, highlighted the importance of smooth frequency transitions to avoid visual distortions.