Project Report

EE-200

Frequency Mixer and Demixer

Course Project

Submitted by

Aviral Gupta

Instructor:

Dr. Tushar Sandhan

Date of Submission: 30/06/25

Q1: Frequency mixer: 'Beauty and the Blur'

1. Introduction

This exercise explores the principles of spatial-frequency analysis by implementing a frequency-domain image mixer. Inspired by hybrid images such as the classic Einstein–Monroe illusion, this method combines the fine details from one image (high frequencies) with the broad, smooth structural elements from another (low frequencies). The resulting fused image illustrates how our visual perception interprets different frequency bands differently, emphasising coarse structure at a distance and fine detail upon closer viewing. The experiment will involve spatial pre-processing, frequency-domain masking, and inverse frequency transformation.

2. Objective

The objective is to design and implement a frequency mixer that demonstrates the separation and recombination of spatial-frequency components. Specifically, this includes:

- Applying a Gaussian blur in the spatial domain to selectively suppress high-frequency content.
- Computing the 2D Fourier transforms (FFTs) of two complementary images.
- Designing and applying frequency-domain masks (low-pass and high-pass) to isolate specific frequency bands.
- Combining masked frequency components and reconstructing the resulting image via the inverse FFT.

This procedure highlights the relationship between spatial-frequency content and visual perception, providing insight into frequency-based image processing methods.

3. Fourier Transform

The Fourier transform provides a mathematical framework for analysing signals or images in terms of their frequency content. By converting an image from its original spatial representation to a frequency-domain representation, we gain insight into the distribution of different spatial frequencies, such as edges, textures, and smooth gradients.
In general, the two-dimensional discrete Fourier transform (2D DFT) of an image $I(x, y)$ of size $M \times N$ is given by:

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

The inverse Fourier transform, used to reconstruct the image from its frequency-domain representation, is defined as:

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

The resulting frequency-domain representation $F(u, v)$ is complex-valued, where:

- **Magnitude** $|F(u, v)|$ describes the strength of each frequency component.
- **Phase** $\angle F(u, v)$ captures the positional information or spatial arrangement.

Typically, the magnitude spectrum is visualised to identify significant spatial frequencies present in the image, informing subsequent image processing steps such as filtering or fusion.

4. What is FFT and how it helps

The Fast Fourier Transform (FFT) is a computationally efficient algorithm to calculate the Discrete Fourier Transform (DFT), significantly reducing the complexity from $O(N^2)$ to $O(N \log N)$. The FFT converts a signal or image from its original domain (time or space) into the frequency domain, decomposing it into sinusoidal components characterised by amplitude and phase.
Using FFT helps by:

- identifying frequency components within an image.
- Facilitating effective filtering by allowing easy manipulation of specific frequency bands.
- Accelerating computations, making frequency-domain analyses practical for large images.
- Enabling straightforward reconstruction from frequency to spatial domains via inverse FFT.

In our project, the FFT allowed precise isolation and manipulation of image frequency bands, enabling the effective fusion of complementary image details and structures.

5. Magnitude Spectra

The magnitude spectrum of an image, defined as the absolute value of its Fourier transform, reveals the spatial frequency content present within the image. Formally, the magnitude spectrum is computed as:

$$|F(u,v)| = \sqrt{\mathrm{Re}(F(u,v))^2 + \mathrm{Im}(F(u,v))^2}$$

Where $\mathrm{Re}(F(u,v))$ and $\mathrm{Im}(F(u,v))$ represent the real and imaginary parts of the Fourier transform, respectively.
When attempting to visualise this magnitude spectrum directly on a linear scale, the resulting image typically appears almost entirely black. This occurs because the spectrum values span an extremely wide dynamic range, with a few dominant low-frequency components having significantly larger magnitudes than the high-frequency ones. Consequently, smaller but still important magnitude values become indistinguishable after normalisation.
To address this issue, a logarithmic (dB) scale is typically employed, which compresses the dynamic range and enhances visibility across all frequency magnitudes:

$$|F(u,v)|_{\text{dB}} = 20\,\log_{10}\left(|F(u,v)| + \epsilon\right)$$

Here, the small constant ϵ prevents logarithmic singularities. This representation makes it possible to clearly visualize both dominant and subtle frequency features simultaneously.
Interpreting magnitude spectra thus allows precise analysis and manipulation of frequency-domain components, which is critical to the successful implementation of our frequency mixer.

In our case, we used $\epsilon = 10^{-5}$ so that it does not affect our plot and solve the error.

6. Libraries Used

The implementation of the frequency mixer and Fourier analysis was carried out using Python in a Jupyter Notebook environment. The following Python libraries were used:

- **NumPy**: Used for efficient numerical computations and array manipulation, including Fourier transforms via `numpy.fft`.
- **Matplotlib**: Employed for plotting and visualizing images, spectra, and graphs throughout the analysis.
- **OpenCV (cv2)**: Utilized for reading and preprocessing images, applying Gaussian blur, and handling image resizing and rotation.

These libraries provided a flexible and powerful framework to perform Fourier-based frequency decomposition, design and apply frequency masks, and reconstruct fused images efficiently.

7. Input Images

Input Images We were provided with two grayscale input images:

- `cat_gray.png` – a grayscale image of a cat, containing fine texture details and sharper features.
- `dog_gray.png` – a grayscale image of a dog, exhibiting smoother textures and broader structural patterns.

These two images were used as the base inputs for frequency decomposition and fusion. The objective was to extract high-frequency details from the cat image and low-frequency structure from the dog image, and then combine them into a single perceptually meaningful output using frequency-domain techniques.

Figure 1: Cat Image (High-Frequency)

Figure 2: Dog Image (Low-Frequency)

8. Fourier Transform and Magnitude Spectra

The first step involved computing the 2D Discrete Fourier Transform (DFT) of both input images using the `fft2` function from the `numpy.fft` module. This function efficiently converts spatial-domain images into their corresponding frequency-domain representations.
We then computed and visualized the magnitude spectra of the transformed images. Two types of magnitude visualizations were generated:

- **Linear Magnitude Spectrum:** The raw magnitude $|F(u, v)|$ was normalized and plotted. However, due to the extreme dynamic range in the frequency data, the linear plot appeared completely black. This occurred because a small number of frequency components had very large magnitudes (on the order of 10^7), while the majority were much smaller—causing the normalization to suppress almost all visible contrast.

  To diagnose the issue, we clipped the magnitude values at the 99th percentile before normalization. This revealed that the few high-valued components were located in the corners of the spectrum, corresponding to the DC and low-frequency components in the unshifted FFT.

- **Logarithmic (dB) Magnitude Spectrum:** To address the visualization challenge, we plotted the magnitude on a logarithmic scale using:

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

  where ϵ is a small constant to avoid logarithmic singularities. This transformation compressed the dynamic range, making both dominant and subtle frequency features visible. The dB spectrum confirmed that the most significant energy was concentrated near the corners.

These observations were instrumental in understanding the frequency behavior of each image and guided the design of masks for the fusion process.

9. Spectrum Center and Frequency Shifting

By default, the output of the 2D Fourier Transform (`fft2` from NumPy) places the lowest frequency (DC component) at the top-left corner of the spectrum. This layout is not intuitive for visual analysis, as the important low-frequency information is not centrally located.
To address this, we applied the `fftshift` function from `numpy.fft`, which reorders the spectrum so that the zero-frequency component is moved to the center of the image. This operation shifts the quadrants of the frequency domain image, resulting in a more interpretable visualization:

- The center of the shifted spectrum now represents low frequencies (coarse structures).
- Frequencies increase radially outward, with high frequencies (edges and fine details) located near the corners.

Observations:

- After applying `fftshift`, the concentration of energy in the center of the spectrum became clearly visible.
- The visual structure of the magnitude spectrum aligned better with human intuition, showing smoother gradients at the center and sharper transitions outward.
- This visualization was instrumental in designing circular masks for filtering specific frequency ranges and confirmed that our fusion strategy would effectively isolate desired features.

All subsequent magnitude spectra were plotted with `fftshift` applied, ensuring that our frequency-domain operations aligned with the spatial characteristics of the images.

10. Effect of Image Rotation on Fourier Spectrum

To further analyze the behavior of frequency-domain representations, we rotated one of the input images (cat or dog) by 90° counter-clockwise in the spatial domain using OpenCV's `cv2.rotate` function. We then computed and plotted its 2D Fourier magnitude spectrum using the same procedure: applying `fft2`, `fftshift`, and logarithmic scaling.
Observations:

- **Identical Spectral Shape:** The overall structure of the magnitude spectrum—characterized by a bright central region and concentric lobes—remains unchanged after rotation. This indicates that the distribution of frequency magnitudes is preserved.
- **90° Rotation Match:** Every pattern or feature visible in the spectrum of the rotated image appears rotated by exactly 90° counter-clockwise compared to the original. This confirms that the frequency domain undergoes the same geometric transformation as the spatial domain.
- **Spatial–Frequency Correspondence:** The experiment validates a key property of the 2D Fourier Transform: a 90° counter-clockwise rotation in the spatial domain results in a corresponding 90° counter-clockwise rotation in the frequency domain, without altering magnitude values.

These observations reinforce the geometric consistency of the Fourier Transform and aid in understanding how spatial transformations reflect in frequency space—an essential insight for designing frequency-selective filters and image fusion systems.

11. Frequency Mixer: Combining Structural and Textural Features

The frequency mixer is the central component of our project. Its objective is to combine two images in such a way that one contributes the low-frequency (structural) content, while the other contributes the high-frequency (textural) detail. This mimics how the human visual system interprets multi-scale information — using coarse structures for overall shape recognition and fine textures for surface details.
In our implementation:

- The **dog image** (`dog_gray.png`) was used as the source of **low-frequency content**, representing broader shapes and structures.

- The **cat image** (`cat_gray.png`) was used as the source of **high-frequency content**, representing edges and fine textures.

To achieve this, we performed the following steps:

1. Computed the 2D FFT of both images and applied `fftshift` to center the frequency spectra.
2. Designed ideal circular frequency masks — a low-pass mask for the dog image and a high-pass mask for the cat image.
3. Optionally applied Gaussian blurring in the spatial domain to enhance low-frequency dominance before transformation.
4. Multiplied each spectrum with the respective mask and summed the filtered spectra.
5. Applied inverse FFT to reconstruct the fused image.

This fusion method produces a hybrid image that appears differently depending on viewing distance — detailed textures from the cat dominate when viewed closely, while broader shapes from the dog emerge from afar. This effect highlights the power of frequency-based image manipulation.

12. Transfer Function Design and 2D Plots

The core of the frequency mixing system lies in the design of appropriate 2D frequency-domain transfer functions — specifically, binary masks that control which parts of each image's frequency spectrum contribute to the final output.

1. Transfer Functions Used

We designed two complementary transfer functions:

- **Low-Pass Mask (Dog Image):** A circular binary mask that passes only the low-frequency components (i.e., values near the center of the shifted spectrum) and blocks the high-frequency details.
- **High-Pass Mask (Cat Image):** A binary mask that is the complement of the low-pass mask. It passes high-frequency components (i.e., edges and fine textures) and blocks the structural, low-frequency content.

These transfer functions were applied to the Fourier-transformed images using element-wise multiplication. This frequency-selective filtering allowed us to retain smooth structures from the dog image and sharp details from the cat image.

2. Implementation Details

- The images were of the same size, say $N \times N$.
- After computing the 2D FFT and applying `fftshift`, we generated the masks centered at $(N/2, N/2)$.
- A circular region of radius r (empirically chosen, e.g., $r = 40$) was used to define the low-pass region.
- The high-pass mask was obtained as $1 -$ low-pass mask.

3. 2D Transfer Function Plot

The visual appearance of the transfer function is shown below:
This 2D binary mask act as frequency selector and form the functional core of the mixer system. Their design directly controls the spatial features preserved in the final hybrid image.

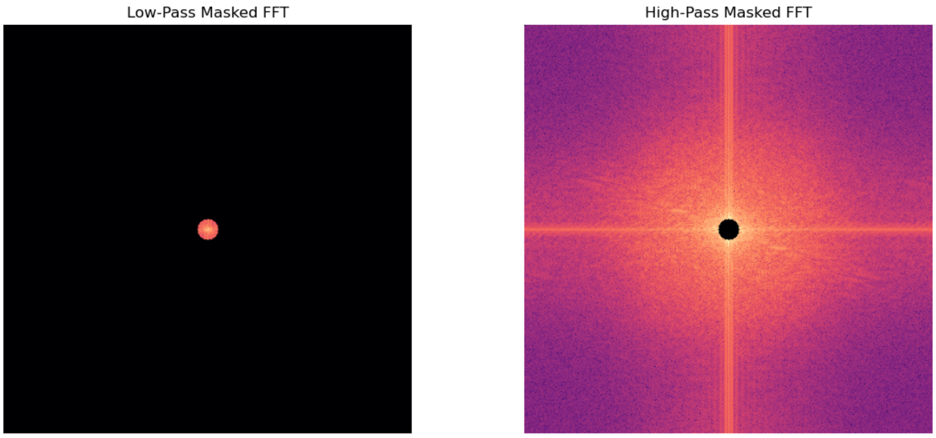

Figure 3: Fused Image: Combination of Dog (Low-Frequency) and Cat (High-Frequency) Features

Figure 4: Fused Image: Combination of Dog (Low-Frequency) and Cat (High-Frequency) Features