Project Report

EE-200

Frequency Mixer and Demixer

Course Project

Submitted by

Aviral Gupta

Instructor:

Dr. Tushar Sandhan

Date of Submission: 30/06/25

Introduction

1. Fourier Transform

The Fourier Transform decomposes an image into its constituent spatial frequencies, enabling analysis of edges, textures, and intensity variations in the frequency domain. The 2D Discrete Fourier Transform (DFT) of an image $I(x,y)$ of size $M \times N$ is:

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

The inverse transform reconstructs the image:

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

Here, $|F(u,v)|$ (magnitude) indicates the strength of frequency components, while $\angle F(u,v)$ (phase) captures spatial structure.

2. Magnitude Spectrum

The magnitude spectrum, $|F(u,v)| = \sqrt{\mathrm{Re}(F)^2 + \mathrm{Im}(F)^2}$, represents spatial frequency strength. Direct visualization is ineffective due to the high dynamic range. Hence, a logarithmic scale is used:

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

A small constant $\epsilon = 10^{-5}$ avoids singularities. This scaling highlights both dominant and subtle frequency features, essential for precise filtering and analysis.

3. Transfer Functions

A transfer function characterises a system's response in the frequency domain, describing how input frequencies are amplified, attenuated, or phase-shifted. In digital signal processing, it helps:

- Analyse system stability and frequency response,
- Design filters through poles and zeros,
- Predict dynamic behaviour in image or audio filtering tasks.

It provides the foundation for understanding and designing frequency-selective operations.

4. Frequency Mixing and Demixing

Frequency mixing is the process of combining frequency components from two or more signals or images to create a fused representation that carries distinct information from each source. In the context of image processing, this typically involves blending low-frequency (structural) components from one image with high-frequency (textural) features from another to create a hybrid image. This technique mirrors the human visual system's interpretation of multi-scale information, perceiving global structure from low frequencies and fine details from high frequencies.

Demixing, conversely, refers to the process of separating or isolating these frequency components. By applying suitable frequency-domain masks, it is possible to extract only the low-frequency content (e.g., smooth regions and shapes) or high-frequency content (e.g., edges and textures) from an image. This separation enables selective manipulation, enhancement, or replacement of specific features.

In this project, frequency mixing was employed to create hybrid images that preserve structural information from one image (e.g., a dog) and superimpose detailed textures from another (e.g., a cat). The effectiveness of this process relies on accurate frequency transformation (via FFT), careful mask design, and inverse reconstruction using IFFT. Together, mixing and demixing form the foundation of our frequency-based fusion pipeline.

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. 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)

4. Fourier Transform and Magnitude Spectra

We computed the 2D Fourier Transforms of both input images using `fft2` to analyse their frequency content. The resulting magnitude spectra were visualised to understand the distribution of low and high-frequency components.

Initially, a linear magnitude plot was generated. However, due to the wide dynamic range, the plot appeared nearly black. Clipping at the 99th percentile revealed that dominant frequencies were concentrated near the corners, representing low-frequency components in the unshifted FFT.

To better visualise frequency content, we used a logarithmic (dB) scale:

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

This compressed the dynamic range, clearly revealing both dominant and subtle features. These observations informed the design of the mask for frequency fusion.

5. Spectrum Centre 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 centre of the image. This operation shifts the quadrants of the frequency domain image, resulting in a more interpretable visualisation:

- The centre 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 centre of the spectrum became visible.
- The visual structure of the magnitude spectrum aligned better with human intuition, showing smoother gradients at the centre and sharper transitions outward.
- This visualisation 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.

6. Effect of Image Rotation on Fourier Spectrum

To further analyse the behaviour 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—characterised 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.

7. Frequency Mixer: Combining Structural and Textural Features

The frequency mixer aims to combine low-frequency (structural) content from one image with high-frequency (textural) detail from another, reflecting the multiscale nature of human visual perception.

- **Dog image:** Supplies low-frequency information representing global shape and smooth intensity transitions.
- **Cat image:** Contributes high-frequency content, capturing sharp edges and fine textures.

The fusion process involved the following steps:

1. Compute the 2D FFT of both images and apply `fftshift` to centre the frequency components.
2. Design complementary circular masks: a low-pass mask for the dog image and a high-pass mask for the cat image.
3. Apply Gaussian blurring ($\sigma = 6$) to the masks to avoid sharp frequency transitions.
4. Multiply each spectrum with its corresponding mask and sum the filtered results.
5. Apply inverse FFT to reconstruct the fused image from the combined spectrum.

The result is a hybrid image where the cat's textural details dominate at close range, while the dog's structural features emerge from a distance, demonstrating the power of frequency-selective fusion.

8. Analysis of Mask Design and Parameters

The fusion technique relies heavily on careful design of frequency masks and parameter tuning for optimal feature separation and integration.

1. Circular Mask Selection

Circular masks were chosen for their rotational symmetry, matching the isotropic distribution of spatial frequencies in natural images. This ensures:

- Equal treatment of all directions in the frequency domain,
- Avoidance of directional artefacts caused by axis-aligned filters,
- Preservation of structural consistency across image regions.

2. Gaussian Blurring of Masks

Ideal binary masks introduce sharp transitions that can cause ringing artefacts in the spatial domain (Gibbs phenomenon). To address this, we applied Gaussian blurring:

- Softened the frequency cutoffs to reduce spatial ringing,
- Enabled smoother transitions between image features,
- Improved perceptual blending of low- and high-frequency content.

3. Radius Selection: 13 Pixels

The radius r determines the cutoff for low- and high-frequency separation:

- **Small radius** (< 10) suppressed most dog image content, making the cat image dominant.
- **Large radius** (> 20) allowed excessive low-frequency content from both images, resulting in blurred superposition.
- **Chosen radius** ($r = 13$) achieved balance: preserving the dog's structure and enhancing it with the cat's textures.

4. Sigma Selection: 6 Pixels

The Gaussian blur standard deviation (σ) affects the mask's transition smoothness:

- **Low sigma** (< 4) retained sharp edges, introducing artefacts.
- **High sigma** (> 8) over-blurred transitions, reducing selectivity between components.
- **Optimal sigma** ($\sigma = 6$) provided a good trade-off: clear separation and minimal ringing.

These parameters were selected based on iterative experimentation, guided by both theoretical expectations and visual evaluation of fusion quality.

9. Transfer Function Design and 2D Mask Plots

The effectiveness of the frequency mixer depends on well-designed transfer functions that control which frequency bands are retained from each image.

1. Transfer Functions Used

Two complementary binary masks were implemented as 2D transfer functions:

- **Low-pass mask (Dog image):** A centered circular mask allowing only low-frequency content.
- **High-pass mask (Cat image):** Defined as the complement of the low-pass mask, passing only high frequencies.

These masks were applied via element-wise multiplication in the frequency domain, allowing selective inclusion of structural and textural components.

2. Implementation Details

- Input images were of the same size: $N \times N$.
- After applying 2D FFT and `fftshift`, the mask was generated centered at $(N/2, N/2)$.
- A radius $r = 13$ defined the cutoff for the low-pass mask; the high-pass mask was computed as $1 -$ low-pass mask.
- Gaussian blur with $\sigma = 6$ was applied to both masks for smooth transitions.

3. Visualisations of Masks and Output

These 2D masks serve as the functional core of the frequency mixer. Their design directly determines which features dominate in the spatially reconstructed hybrid image.

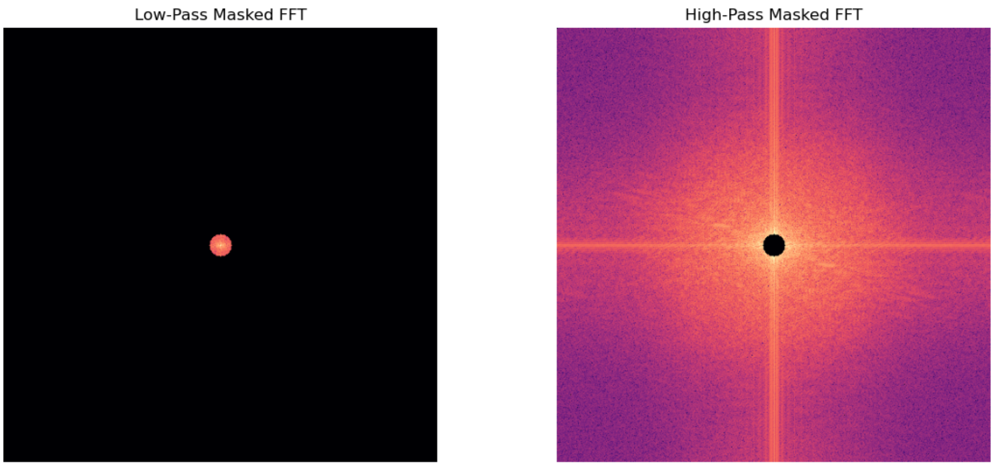

Figure 3: Visualisations of Low-Pass and High-Pass Masks Applied in the Frequency Domain

Figure 4: Fused Image: Dog (Structure) + Cat (Texture)

Q2. Frequency de-mixer: 'Unwanted Solo'

1. Introduction

Digital Signal Processing (DSP) plays a fundamental role in the analysis and modification of signals such as audio, images, and sensor data. In the context of audio processing, DSP techniques allow us to emphasise, suppress, or isolate specific frequency components to enhance sound quality, remove noise, or extract meaningful information.

This experiment aims to investigate the impact of various filter design approaches on an audio signal. Both time-domain and frequency-domain representations are examined to interpret the impact of filtering. Interactive tools are used to dynamically explore parameter changes, allowing for real-time observation of their influence. The experiment culminates with a phase-reversed subtraction technique, demonstrating how specific frequency bands can be selectively attenuated from the original audio.

2. Objective

The objective of this experiment is to explore the fundamentals of digital signal processing by analysing an audio signal in both time and frequency domains. The task involves designing and applying digital filters using different design methodologies to manipulate specific frequency components within the signal. The experiment also aims to visualise the system behaviour through frequency responses, power spectral density plots, and pole-zero representations.

3. Noise and Its Removal in Signal Processing

In signal processing, **noise** refers to any unwanted or irrelevant component that contaminates the original signal and degrades its quality or intelligibility. Noise can originate from various sources such as electronic interference, environmental disturbances, or limitations in recording equipment. In the frequency domain, noise often appears as additional energy spread across a broad range of frequencies, particularly in regions where the true signal has low or no activity.

To remove or reduce noise, filtering techniques are employed. Filters are designed to pass frequency components that belong to the desired signal while attenuating those associated with noise. For instance, if noise primarily occupies higher frequencies, a low-pass filter can be used to suppress it. Conversely, if noise localised in a narrow band, a band-stop filter may be more effective.

4. Preprocessing and Initial Analysis

The experiment began by loading an audio signal using the `librosa` library. If the input signal was in stereo format, it was converted to mono by averaging the channels to simplify further processing.
Next, the signal was analysed in the time domain by plotting its waveform. This provided insight into the amplitude variations over time. The corresponding frequency domain representation was obtained by computing and plotting the magnitude spectrum using the Fourier Transform.

To visualise how the frequency content of the signal evolves, a spectrogram was also generated. The spectrogram offered a time-frequency representation that helped identify dominant frequency bands and regions potentially affected by noise.

These initial visualisations formed the foundation for designing appropriate filters to manipulate or remove specific frequency components from the signal.

5. Observations from Initial Analysis

- The time-domain waveform revealed clear amplitude variations, indicating segments of high and low energy throughout the signal.
- The magnitude spectrum showed that the dominant frequency components were concentrated in the ranges of approximately **100–400 Hz**, **around 1000 Hz**, and a prominent band between **1500–2500 Hz**.

- The spectrogram provided a detailed time-frequency representation, highlighting when and for how long different frequency bands were active. Notably, an additional frequency component was visible in the **1024–4096 Hz** band, possibly indicating structured noise or a distinct tonal element.
- Regions with low amplitude and high-frequency content in the spectrogram appeared in yellow, suggesting transient or low-energy noise.
- These insights guided the design of filters to either attenuate or isolate specific frequency bands for further processing and analysis.

5. Filter Design and Implementation

Following the initial analysis of the audio signal, targeted filtering was required to suppress undesired components while preserving the quality of the underlying signal. To achieve this, a flexible digital filter framework was developed using Python, incorporating both FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) designs.

Filter Types and Structure

Two core approaches to digital filter design were explored:

- **Finite Impulse Response (FIR) Filters:** These filters were constructed using the window method with Blackman windows to ensure effective sidelobe attenuation. FIR filters offer the benefit of a strictly linear phase response, which is particularly desirable for audio processing where phase distortions must be minimised.
- **Infinite Impulse Response (IIR) Filters:** Butterworth filters were chosen for IIR design due to their smooth and monotonic frequency response in both passband and stopband. IIR filters provide more compact implementations for the same level of attenuation but introduce nonlinear phase shifts.

Each filter was designed to suppress or isolate specific frequency bands identified through spectral analysis. To avoid phase distortion, zero-phase filtering was achieved using the `filtfilt()` function from the `scipy.Signal` module, which applies forward and reverse filtering.

Visual and Auditory Feedback

For each filtering configuration, the following analyses andvisualisationss were generated:

- **Frequency Response:** Gain vs frequency plots illustrated the passbands and stopbands of the filter.
- **Time-Domain Signals:** Comparison of original and filtered waveforms provided insight into amplitude changes and signal shaping.
- **Power Spectral Density (PSD):** Welch's method was used to estimate and visualise the power distribution of the filtered signal.
- **Pole-Zero Plot:** The pole-zero configuration of each filter was computed using the transfer function coefficients and plotted to analyse system stability and behaviour.
- **Audio Playback:** Side-by-side audio players enabled subjective evaluation of the filtering effect.

6. Filter Analysis and Observations

To isolate and suppress the intrusive piccolo flute from the original audio signal, a variety of frequency-selective filters were applied. Each filter served a distinct role in analysing the frequency distribution of the signal and guiding the noise removal strategy.

1. Low-Pass Filter (Cutoff = 1000 Hz)

A low-pass filter with a cutoff frequency of 1000 Hz was used to retain the lower frequency components of the audio.

- **Observation:** The filtered output predominantly preserved the vocal elements.
- **Effect:** Most high-frequency instruments, including the intrusive flute, were effectively removed.
- **Inference:** Vocals reside primarily in the lower frequency band, below 1000 Hz.

2. High-Pass Filter (Cutoff = 1500 Hz)

A high-pass filter was applied to remove low-frequency content and retain upper frequencies.

- **Observation:** Vocals were almost completely suppressed; only instruments and flute remained.
- **Effect:** The flute's sound was audible, confirming its higher-frequency placement.
- **Inference:** The intrusive component is concentrated above 1500 Hz.

3. Band-Pass Filter (1200–2500 Hz)

A band-pass filter was used to narrow down the frequency range responsible for the flute interference.

- **Observation:** The output audio consisted almost exclusively of the flute sound.
- **Effect:** Other song components (vocals, harmony) were almost entirely removed.
- **Inference:** The flute energy is highly localised between 1200–2500 Hz, making it suitable for targeted suppression.

4. Band-Stop Filter (500–4500 Hz)

Based on prior results, a band-stop filter was designed to suppress a broader mid-frequency range.

- **Observation:** The flute component was significantly attenuated.
- **Effect:** The remaining audio preserved vocals and higher-frequency instruments with minimal distortion.
- **Trade-off:** Some musical details within the suppressed band were also affected, but overall quality remained acceptable.

5. Phase-Reversal Subtraction Technique

To enhance precision and avoid over-filtering, a custom approach was introduced to subtract the flute component more surgically.

1. The flute component was isolated using a band-pass filter (1200–2500 Hz).
2. Its phase was inverted.
3. The inverted signal was subtracted from the original waveform.

Outcome:

- Allowed targeted removal of the intrusive signal with minimal effect on nearby frequencies.
- Preserved the phase integrity and timbre of the rest of the audio.
- Enabled independent analysis or reuse of the isolated flute component.

6. Summary

- Each filter contributed uniquely to understanding the spectral structure of the audio.
- The band-pass filter precisely localised the intrusive signal.
- The band-stop filter offered a strong baseline for noise suppression.
- The custom subtraction technique provided refined control with minimal audio degradation.

This structured filtering pipeline, combining classical and custom techniques, led to effective demixing with a favourable balance between noise reduction and audio fidelity.

7. Challenges Encountered

A significant challenge encountered during the filtering process was the overlap between the noise and the desired audio content in the frequency domain**. The intrusive sound, primarily due to a piccolo flute, shared a frequency band (approximately 1200–2500 Hz) with some musical elements of the original song.

As a result, applying traditional filters such as band-stop or low-pass to remove the noise also inadvertently attenuated parts of the desired signal. This made it difficult to achieve clean separation without compromising audio quality. The frequency overlap limited the effectiveness of purely frequency-selective filtering, highlighting the need for more nuanced approaches like phase inversion subtraction or time-frequency masking to better preserve the integrity of the original audio.

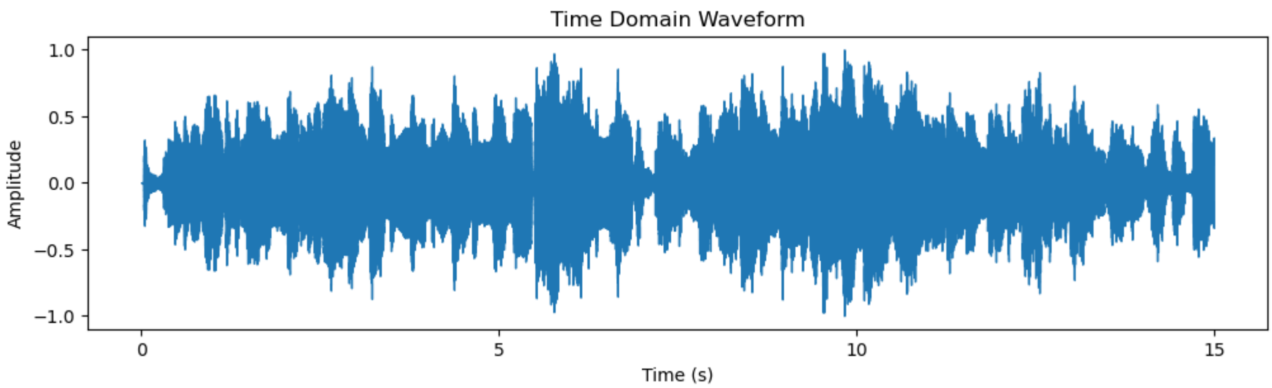

Figure 5: Time-Domain Waveform of the sample audio

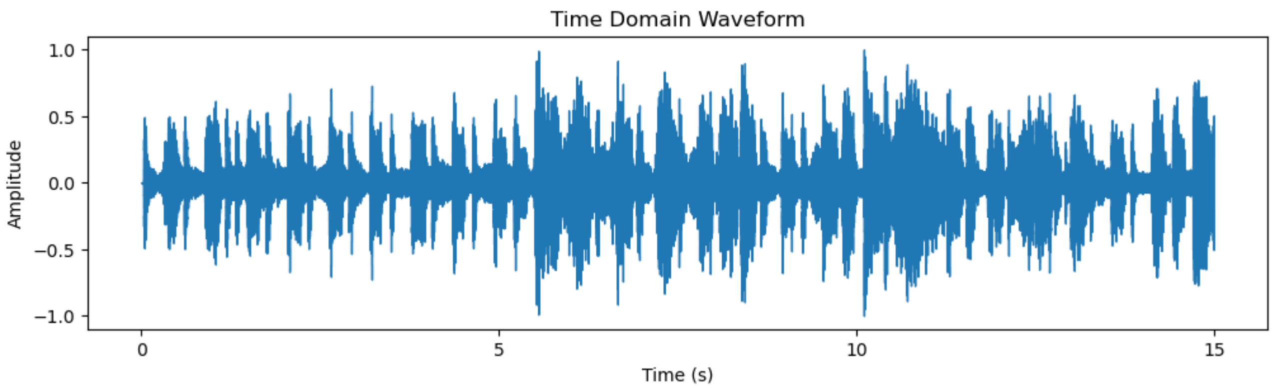

Figure 6: Time-Domain Waveform of the recovered audio