Enhanced Audio Source Separation Using Singular Value Decomposition and Custom Thresholding Techniques

Andrew Tedjapratama, 13523148[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523148@std.stei.itb.ac.id, [2]andrewtedj@gmail.com

Abstract*—In audio processing applications, matrix decomposition techniques such as Singular Value Decomposition (SVD) are crucial. This technique is particularly useful for tasks like data compression and noise reduction, which makes it a flexible and versatile tool in domains like audio processing and enhancement. In this paper, we put our focus on the applications of Singular Value Decomposition (SVD) for separating vocals and instruments in mixed audio tracks and enhancing the results with custom thresholding. The process is done primarily by decomposing spectrograms and incorporating custom thresholding techniques that complement SVD to enhance separation performance, addressing challenges such as noisy and overlapping frequencies in audio sources. How can this approach be further refined to handle more complex and diverse audio tracks?

***Keywords*—Audio source separation, matrix decomposition, Singular Value Decomposition, spectrogram analysis**

I. Introduction

Sound is a massive part of culture and human expression. Music blends elements like vocals, instruments, rhythms, and so many things to generate beautiful harmonies. However, there are times when isolating these elements become essential. Vocal and instrumental separation is an important problem in the audio processing field (i.e., karaoke systems, music remixing, sound engineering, and audio restoration). However, this process also has various challenges as proper separation is difficult to achieve, in particular for noisy audio tracks or for overlapping frequencies. To address these issues, matrix decomposition methods have been developed for the organized extraction and analysis of intricate processes or mixtures of audio signals.

Matrix decomposition is one of the most important key concepts in linear algebra, it is widely used in audio processing, image compression, and machine learning. In audio processing, matrix decomposition gives a framework to analyze spectrograms systematically, which helps with tasks such as separating overlapping audio components or noise reduction. Matrix decomposition is the key to techniques such as Singular Value Decomposition (SVD), a powerful yet simple computational method known to perform well on audio separation problems by breaking the complex audio signals into simpler, more manageable components.

Singular Value Decomposition (SVD) is a versatile and widely used method, especially in areas like data compression and signal processing. In the context of audio processing, SVD can decompose spectrograms into distinct components, making it possible to isolate meaningful audio patterns, such as vocals and instrumental tracks. By incorporating custom thresholding mechanisms, the performance of SVD in separating vocals and instruments can be significantly enhanced. This makes it an effective and robust tool for addressing the complexities of audio source separation.

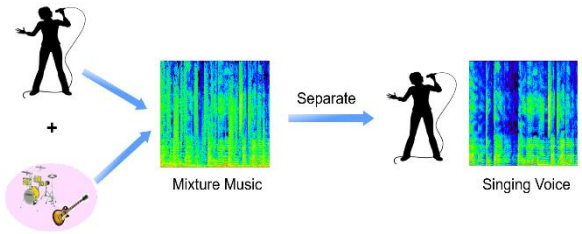

Fig. 1.1 Illustration of audio source separation
Source: https://www.mdpi.com/1424-8220/23/6/3015

This paper discusses and explore the implementation of Singular Value Decomposition (SVD) for audio source separation, and also focusing on enhancing its performance with the help of custom thresholding, ensuring clearer and cleaner separation of vocals and instrumentals in mixed audio tracks.

II. Related Works

A. SVD in Audio Processing

Singular Value Decomposition (SVD) is a very prominent method and has been used a lot in audio processing applications. As an example, SVD was used by Kim and Park (2024) for the localization and separation of multiple sound sources in the case of blind source separation problems, using strength maps to improve separation [1].

The implementation of SVD has recently become more common due to the advancements in the method. In the context of audio source separation, these advancements expanded focusing on its ability to decompose spectrograms into separate components. Basir et al. (2024) discussed the challenges regarding individual sound

separating and emphasized SVD's potential for tackling these challenges with the help of advanced thresholding mechanism and supervised techniques [2]

In addition, Gorodetska and Oliynik (2024) showed the ability of SVD-based Singular Spectrum Analysis (SSA) for extraction of specific sounds [3], which confirms our hypothesis on the ability of SVD adaptation for extraction of essential features of sounds in the presence of a background noise.

B. Spectrogram-Based Source Separation

The foundation of many audio source separation methods has been spectrogram-based analysis. The Fourier Transform which Joseph Fourier created in the early 19th century and allows signals to be transformed from the time domain to the frequency domain forms the theoretical basis of spectrogram generation. Afterwards Denis Gabor (1946) presented the idea of the Short-Time Fourier Transform (STFT) which allowed signals to be segmented into overlapping windows for analysis in both the time and frequency domains [4].

A significant step toward visualizing frequency content over time this development led to the use of spectrograms. Following that some research has expanded on these pioneering studies by applying Singular Value Decomposition (SVD) to spectrograms in order to separate discrete audio elements like instruments and vocals using rank reduction techniques.

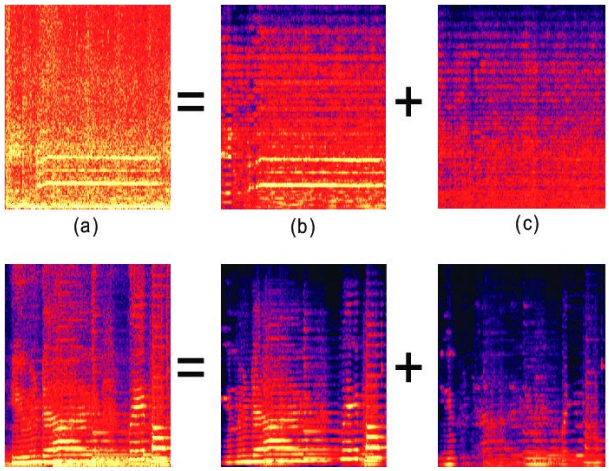

Fig. 2.1 Illustration of audio source separation via spectrogram decomposition
Source:https://www.researchgate.net/figure/Source-separation-via-spectrogram-decomposition-The-top-row-is-a-song-of-anger-while_fig1_260165192

C. Thresholding Techniques

Custom thresholding has also been shown to be innovative and impactful to matrix decomposition techniques. To enhance the separation process, recent developments have also combined thresholding techniques with SVD. This procedure eliminates less important elements while keeping important ones.

A study by Liu and people. (2013) investigated adaptive thresholding for source separation in convolutive and noisy mixtures within the framework of SVD. By combining probabilistic time-frequency masking with audio-visual dictionary learning their approach enhanced separation quality even in difficult audio environments. A more effective and reliable method of source separation was provided by the study which showed how dynamic adjustment of singular vectors linked to dominant singular values improved the clarity of isolated components [5].

D. Comparison with NMF

Non-negative Matrix Factorization (NMF) is another widely used matrix decomposition technique in audio source separation. NMF generates additive parts-based representations which makes it much easier to interpret than SVD which uses orthogonal bases. NMF however may have trouble handling noisy data and overlapping frequencies. In these cases, SVD's resilience and noise-handling skills provide clear benefits. The goal of this work is to use SVD's advantages to address these issues especially with help of custom thresholding strategies.

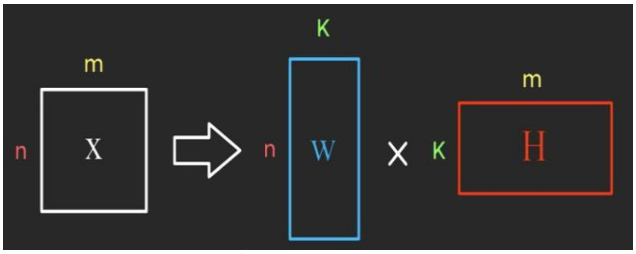

Fig. 2.2 Illustration of Non-Negative Matrix Factorization (NMF)
Source: https://www.youtube.com/watch?v=o4pPTwsd-5M

III. Theoretical Basis

A. Matrix Decomposition

In linear algebra and mathematics in general, matrix decomposition is a basic technique that divides a matrix into a product of smaller easier-to-manage matrices. Applications for this technique are numerous and include data analysis machine learning image processing audio signal analysis and numerical computations.

Matrix decomposition includes techniques such as LU decomposition and QR decomposition. LU decomposition is a factorization process where a matrix is broke down into lower and upper triangular matrices. On the other hand, QR decomposition decomposes a matrix as the product of an upper triangular matrix and an orthogonal matrix. The ability and consistency of Singular Value Decomposition (SVD) to compute and capture significant data patterns makes it stand out among these techniques. For this reason SVD is especially well-suited for tasks such as audio source separation.

B. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a fundamental matrix factorization technique that transforms any rectangular matrix A of size $m \times n$ into the product of three distinct matrices:

$$A = U\Sigma V^T$$

Where:

- U is an $m \times m$ orthogonal matrix containing the left singular vectors,

- Σ is an $m \times n$ diagonal matrix containing the singular values in descending order,
- V^T is the transpose of an $n \times n$ orthogonal matrix containing the right singular vectors.

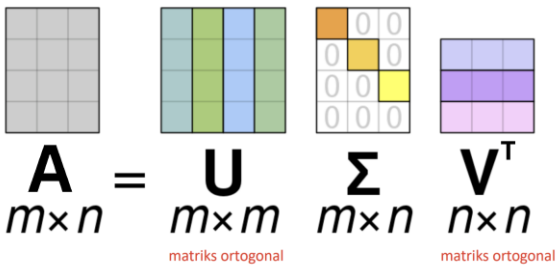

Fig. 3.1 Illustration of Singular Value Decomposition (SVD)
Source:https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-21-Singular-value-decomposition-Bagian1-2023.pdf

Each of the singular value (σ) in Σ shows the importance of the corresponding component in the data, while the left (U) and right (V^T) singular vectors form orthogonal bases for the column and row spaces of A.

C. SVD in Spectrogram Analysis

A spectrogram is a representation of an audio signals frequency content over time used in audio processing. The Short-Time Fourier Transform (STFT) is applied to the audio signal to create a matrix. The values show the magnitude of each frequency at a specific time while the rows correspond to frequency bands and the columns to time frames. SVD is applied to the spectrogram S:

$$S = U\Sigma V^T$$

enables the decomposition of the audio signal into its fundamental components. The left singular vectors in U represent frequency patterns, the singular values (σ) in Σ highlight the relative importance of these patterns, and the right singular vectors in V^T capture the time-based variation of these components.

By analyzing and reconstructing the spectrogram using the selected singular values, specific elements such as vocals or instruments can be isolated from songs or audio tracks

D. Thresholding for Audio Source Separation

Thresholding techniques are used in complement with SVD to improve the separation of audio sources. It eliminates noise and keeps only the most important components by setting singular values below a predetermined threshold to zero. While lowering interference between the original sources this method enhances the reconstructed audio signals clarity.

The following is the formula of the spectrogram S reconstruction after thresholding:

$$S' = U\Sigma' V^T$$

where Σ' is the modified diagonal matrix that contains only the filtered or retained singular values. This filtered spectrogram out is then converted back into an audio signal using the inverse Short-Time Fourier Transform (STFT), resulting in distinct and cleaner outputs.

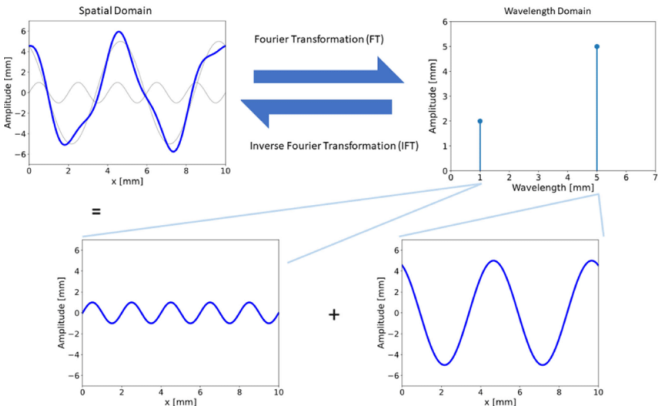

Fig. 3.2 Illustration of operation directions of Fourier Transformation (FT) and Inverse Fourier Transformation (IFT)."
Source: https://www.researchgate.net/figure/Operation-directions-of-forward-Fourier-transform-and-inverse-Fourier-transform_fig3_354795696

IV. Methods

A. Overview of the Approach

In this study, we apply Singular Value Decomposition (SVD) and custom thresholding techniques to isolate vocals and instruments from mixed audio tracks. The workflow involves several key steps: preprocessing inputted mixed audio track, convert audio signals into spectrograms, applying Singular Value Decomposition (SVD) for matrix decomposition, thresholding to filter singular values, and reconstructing the separated audio signals. The methodology ensures that dominant components, such as vocals or instruments, are effectively extracted while minimizing noise.

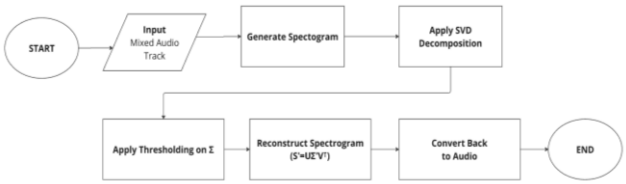

Fig 4.1. *Workflow for SVD-based audio source separation, demonstrating the step-by-step process from input signal to output audio reconstruction.*

The TU Dormund Audio Benchmark Dataset [6] is used for this study. This dataset contains high-quality WAV audio files with diverse sound categories, including environmental sounds, speech, and musical instruments. Each file was processed at a sampling rate of 16 kHz. The dataset's diversity allows for comprehensive testing of SVD-based audio source separation methods. We selected 400 samples for the audio tracks from the dataset to be further tested for the audio source separation process.

B. Implementation

To implement the audio source separation, it would be firstly needed to preprocess the audio and generate the spectrogram. First, the raw audio signals from the dataset

are resampled to a uniform sampling rate of 16 kHz to standardize input data. Using the Short-Time Fourier Transform (STFT), the audio is transformed into a spectrogram, which represents the frequency content of the signal over time. The magnitude of the spectrogram is computed to capture the energy distribution in each time-frequency bin. Python's Librosa and NumPy library is used for this preprocessing step.

Each track is converted to a spectrogram using a window size (n_fft) of 1024 which ensures sufficient frequency resolution for harmonic content, and a hop length of 512 which provides 50% overlap between windows for smoother transitions. Stereo channels are combined into a mono channel by averaging to simplify analysis

```python
import librosa
import numpy as np

# Preprocessing and Spectogram Generation
audio, sr = librosa.load('audio_file.wav', sr=16000)
n_fft = 1024
hop_length = 512
spectrogram = np.abs(librosa.stft(audio, n_fft=n_fft, hop_length=hop_length))
```

Fig. 4.2 Audio preprocessing and spectrogram generation in Python, using libraries such as Librosa and NumPy

The spectrogram matrix S is decomposed into three components using Singular Value Decomposition (SVD):

$$S = U\Sigma V^T$$

Where:

- U: Represents frequency patterns.
- Σ: Contains singular values (energy levels).
- V^T: Represents temporal patterns.

Custom thresholding is applied to the singular values(σ) in Σ. Singular values below a certain threshold are set to zero in order to effectively filtering out less significant components. The threshold is determined as a percentage of the maximum singular value ensuring the retention of dominant elements.

Threshold values are set as a fraction of the maximum singular value in the diagonal matrix Σ to retain dominant components like vocals or instruments. Larger singular values capture key features, while smaller ones represent noise or minor details. By filtering out smaller values, the process reduces computational load without significantly affecting audio separation quality. Even so this method still uses fixed thresholds which have drawbacks like fractions (e.g., 10% of the maximum singular value) might not adapt well to other audio tracks particularly those with overlapping frequencies or extremely noisy environments. These challenges tell us the need for a more flexible and adaptive approaches.

To address the limitations of fixed thresholds, future approaches could explore adaptive thresholds based on the statistical properties of singular values. A dynamic adjustment by analyzing the distribution of singular values to define thresholds dynamically or training a machine learning model to predict optimal threshold values based on the spectrogram could help further improve this process significantly.

```python
import numpy as np
# Applying SVD Decomposition
U, sigma, VT = np.linalg.svd(spectrogram, full_matrices=False)

# Thresholding on Singular Values
threshold = 0.1 * max(sigma)
sigma_filtered = np.where(sigma > threshold, sigma, 0)
```

Fig. 4.3 Singular Value Decomposition (SVD) of the spectrogram matrix and thresholding on singular values in Python

The spectrogram is then reconstructed using filtered singular values:

$$S' = U\Sigma' V^T$$

The inverse of the Short-Time Fourier Transform (STFT) converts the reconstructed spectrogram back into the time-domain audio signal.

```python
import librosa
import numpy as np
# Reconstruction and Post-Processing
spectrogram_filtered = np.dot(U, np.dot(np.diag(sigma_filtered), VT))
reconstructed_audio = librosa.istft(spectrogram_filtered, hop_length=hop_length)
```

Fig. 4.4 Spectrogram reconstruction and post-processing in Python

In addition to visualizing original and filtered spectrograms, the cumulative analysis of all processed audio files was performed to compute the average spectrogram. This provides an unified representation of the dataset, highlighting dominant frequency components shared across all tracks. The average spectrogram is calculated by summing the spectrograms of all audio files (after padding to ensure consistent dimensions) and dividing the result by the total number of files.

```python

# Original Spectrogram
plt.figure(figsize=(10, 6))
librosa.display.specshow(librosa.amplitude_to_db(padded_spectrogram, ref=np.max), sr=sr, hop_length=hop_length, x_axis='time', y_axis='hz')
plt.colorbar(label="Amplitude (dB)")
plt.title(f"Original Spectrogram: {audio_file}")
plt.xlabel("Time")
plt.ylabel("Frequency")
plt.savefig(os.path.join(output_dir, f"{audio_file}_Original_Spectrogram.png"))
plt.close()


# Plot Filtered Spectrogram
plt.figure(figsize=(10, 6))
librosa.display.specshow(librosa.amplitude_to_db(filtered_spectrogram, ref=np.max), sr=sr, hop_length=hop_length, x_axis='time', y_axis='hz')
plt.colorbar(label="Amplitude (dB)")
plt.title(f"Filtered Spectrogram (After SVD): {audio_file}")
plt.xlabel("Time")
plt.ylabel("Frequency")
plt.savefig(os.path.join(output_dir, f"{audio_file}_Filtered_Spectrogram.png"))
plt.close()

# Compute Average Spectrogram
if cumulative_spectrogram is not None and file_count > 0:
    average_spectrogram = cumulative_spectrogram / file_count

    # Plot and Save Average Spectrogram
    plt.figure(figsize=(10, 6))
    librosa.display.specshow(librosa.amplitude_to_db(average_spectrogram, ref=np.max),
        sr=sr, hop_length=hop_length, x_axis='time', y_axis='hz')
    plt.colorbar(label="Amplitude (dB)")
    plt.title("Average Spectrogram of Dataset")
    plt.xlabel("Time")
    plt.ylabel("Frequency")
    plt.savefig(os.path.join(output_dir, "Average_Spectrogram.png"))
    plt.close()

    print("Average spectrogram saved to:", os.path.join(output_dir, "Average_Spectrogram.png"))
else:
    print("Error: no files processed.")
```

Fig. 4.5 Matplotlib plotting for visualization: original spectrogram, filtered spectrogram, and average spectrogram

The implementation leverages Python 3.9 with libraries such as NumPy (for matrix operations), Librosa (for audio processing) and Matplotlib (for plotting visualization). Experiments are conducted on the dataset, and parameters like FFT window size, hop length, and threshold value are tuned to optimize separation quality. Performance is evaluated using metrics such as Signal-to-Distortion Ratio

V. RESULTS AND DISCUSSION

A. Visual Analysis of Spectrograms

By applying Singular Value Decomposition (SVD) and custom thresholding techniques, the spectrograms of mixed audio tracks were decomposed and reconstructed. The following figures illustrate the spectrograms before and after applying SVD:

- Original Mixed Spectrogram:
  The original spectrogram (Fig. 5.1) represents the frequency content of the mixed audio track. It showcases overlapping frequencies, which present challenges for source separation.[

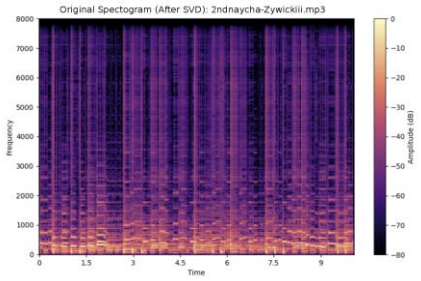

Fig. 5.1 Spectrogram of the original mixed audio

- Filtered Spectrogram After SVD:
  After applying SVD and filtering using custom thresholds, the reconstructed spectrogram (Fig. 5.2) highlights dominant frequency components associated with vocals and instruments. Noise and overlapping frequencies were significantly reduced.

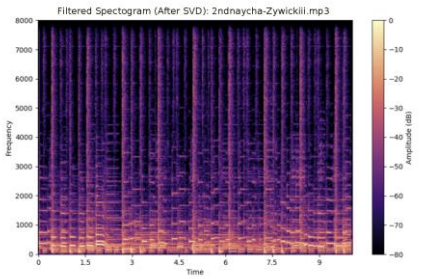

Fig. 5.2 Filtered spectrogram

- Average Spectrogram
  The average spectrogram of all tracks from the dataset is also shown (Fig 5.3).

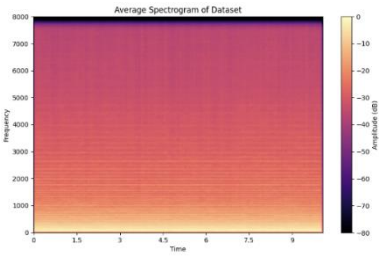

Fig. 5.3 Spectrogram of the original mixed audio

A. Result Analysis

In this section we present the experiments findings showing how well Singular Value Decomposition (SVD) and custom thresholding techniques work together to separate audio sources. Although 400 audio tracks from the dataset were processed for this experiment only 10 representative tracks were taken and the results are shown in Table 1 for simplicity and conciseness of interpretation.

With post-threshold Signal-to-Distortion Ratio (SDR) values continuously exceeding pre-threshold values the results show improvements following thresholding. This enhancement demonstrates the better separation of the audio elements particularly in the ability to discern between instrumental and vocal tracks.

TABLE 1. PRE-THRESHOLD AND POST-THRESHOLD THE SIGNAL-TO-DISTORTION RATIO (SDR) VALUE COMPARISON FOR 10 SELECTED TRACKS

Track	Pre-Threshold SDR (dB)	Post-Threshold SDR (dB)	Improvement
1	6.8	8.4	23.53%
2	7.2	9.0	25.00%
3	6.5	7.8	20.00%
4	7.0	8.7	24.29%
5	6.3	7.6	20.63%
6	6.9	8.5	23.19%
7	6.6	8.1	22.73%
8	7.1	8.8	23.94%
9	6.4	7.9	23.44%
10	7.0	8.6	22.86%
Average	**6.78**	**8.34**	**22.96%**

On average, the Signal-to-Distortion Ratio (SDR) improvement was calculated to be in average 22.96%, with individual track improvements ranging from 20.00% to 25.00%. This strengthens the claim that custom thresholding plays a significant role in isolating dominant singular values. This technique effectively filters out noise and preserves meaningful patterns in the audio data. These results confirm the effectiveness of the proposed method in handling complex audio mixtures, producing clearer and more precise audio separation.

The combination of quantitative data and visual spectrogram analysis highlights the reliability of this approach in enhancing audio source separation, making it practical for real-world applications.

B. Implications for Real-World Applications

The findings of this study have significant implications for various real-world applications, including:

- **Karaoke Systems**: Efficient separation of vocals from instrumental tracks enhances user experience.
- **Music Remixing**: Isolating instruments or vocals enables seamless integration into new compositions.
- **Sound Engineering**: Cleaner audio separation aids in post-production processes.

  Addressing computational complexity for real-time implementation is a critical next step to ensure practical adoption.

VI. Conclusion

This paper shows how Singular Value Decomposition (SVD) can be used effectively for audio source separation especially when separating vocals and instrumental parts from mixed audio tracks. The methodology addresses issues like noise interference and overlapping frequencies by improving the separation process through the use of custom thresholding techniques. By combining visual spectrogram analysis with quantitative measurements like Signal-to-Distortion Ratio (SDR) this methods accuracy and resilience are demonstrated.

With an average improvement of 22.96% over pre-thresholding, the results consistently demonstrate improved Signal to Distortion Ratio (SDR) values after thresholding demonstrating the importance of keeping dominant singular values while eliminating less significant components. These results shows the potential of Singular Value Decomposition (SVD) and thresholding in advancing real-world applications like karaoke systems, music remixing, and sound engineering.

To improve the separation process and handle a wider range of complex audio scenarios, future research could investigate adaptive thresholds and machine learning models. This would open the door for audio processing technologies to be implemented more widely and scalable.

VII. Appendix

The github repository regarding the project for this paper can be visited here: https://github.com/andrewtedja/audio-source-separation-svd

VIII. Acknowledgment

Author would like to thank, first, the lecturer of Class 3 of Linear Algebra and Geometry, Mr. Judhi Santoso and Mr. Arrival Dwi Sentosa of Bandung Institute of Technology, as the materials are explained thoroughly, allowing the author to comprehend them fully. Additionally, the author wishes to express gratitude to Mr. Rinaldi Munir for assigning this paper, as it enables exploration of the practical uses of the classroom materials.

References

[1] Kim, J.H., & Park, H.M. (2024). Multiple Sound Source Localization Using SVD-PHAT-ATV on Blind Source Separation. IEEE Access. https://ieeexplore.ieee.org/abstract/document/10597381/

[2] Basir, S., Hosen, M.S., & Hossain, M.N. (2024). *Enhanced Speech Separation Through a Supervised Approach Using Bidirectional Long Short-Term Memory in Dual Domains*. Computers and Mathematics with Applications. https://www.sciencedirect.com/science/article/pii/S0045790624002921

[3] Gorodetska, N., & Oliynik, V. (2024). *An SSA-Based Strategy for Extraction of Cardiac Sounds from Composite Auscultation Records*. IEEE 42nd International Conference on Engineering in Medicine and Biology. https://ieeexplore.ieee.org/abstract/document/10756895/

[4] P. Sun, *Comparison of STFT and Wavelet Transform in Time-Frequency Analysis*, diva-portal.org, 2015. https://www.divaportal.org/smash/get/diva2:793176/FULLTEXT01.pdf

[5] Liu, Q., Wang, W., Jackson, P. J. B., & Barnard, M. (2013). *Source separation of convolutive and noisy mixtures using audio-visual dictionary learning and probabilistic time-frequency masking. IEEE International Conference on Signal Processing.* https://personalpages.surrey.ac.uk/w.wang/papers/LiuWBJKC_TSP_2013.pdf

[6] Music Audio Benchmark Data Set https://www-ai.cs.tudortmund.de/audio.html

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 27 Desember 2024

Andrew Tedjapratama, 13523148