NIKHIL CHAPRE (19BCE1315)

ABSTRACT

Speech Separation is one of the most fundamental tasks in the domain of signal processing. This problem revolves around the idea of an acoustic environment where multiple sounds are present and the task is to separate the target speech signal from the surrounding noise/interference or non-relevant speakers.
In this thesis, the proposed method utilizes a supervised learning approach and is trained on a large dataset of mixed speech signals, where the model is capable of separating speech signals even with different levels of complexity of external factors such as noise and room reverberation.

First, we start with building a vocal separation model, where our goal is to isolate the singing voice from a mixed music track containing multiple instruments and background noise. This task is challenging because the singing voice shares spectral characteristics with other instruments in the mix and is often occluded by background noise. This model was trained on the MUSDB18 dataset of mixed music tracks and corresponding isolated vocals, bass drums, accompaniment etc.

We further extend this baseline model to solve the mixed speaker separation problem which involves separating multiple speech signals from a mixed audio signal, where each speech signal belongs to a different speaker. This is a more complex task than vocal separation, as it requires the model to not only separate the vocals from the background music or noise, but also to distinguish between different speakers.

Lastly, we will use this knowledge of separating audio signals that correspond to different speakers in a multi-speaker scenario, using both audio and visual information. Visual information can provide additional cues that can help disentangle the audio signals of multiple speakers, such as the movement of the lips or the position of the speakers in the video.

Chapter 1

INTRODUCTION

This section presents an overview of different traditional and state-of-the-art methods for Speech Separation used throughout its inception. Over the years, a large number of techniques have been developed to solve the speech separation problem, ranging from simple signal processing methods to complex machine learning models.

1.1 TRADITIONAL METHODS

Since its inception Speech Separation was considered as a Signal Processing problem and approaches involved around exploiting the statistical properties of the mixture of signals. Early methods for speech separation focused on the use of spatial separation techniques such as microphone arrays or beamforming, where multiple microphones are used to record speech signals from different locations and the signals are combined to extract individual sources. However, these methods were limited by their dependence on the spatial layout of the sources and the number and placement of the microphones, as well as their sensitivity to noise and reverberation.

1.1.1 INDEPENDENT COMPONENT ANALYSIS (ICA)

Independent component analysis (ICA) is a widely used method in speech separation which assumes that the mixture of signals is a linear combination of independent sources, and it tries to estimate the sources by finding a transformation matrix that maximizes the statistical independence of the estimated signals.

Given a mixture signal X, ICA aims to estimate a transformation matrix A such that the estimated sources Y = AX is statistically independent where A is an invertible matrix. The inverse transformation of matrix A can be used to reconstruct the separated signals from the estimated source Y.

ALGORITHM:

```
START:
1. Preprocess the input mixed audio signal.
2. Construct an observation matrix from the preprocessed mixed audio signal.
3. Initialize the unmixing matrix with random values or start with all zeros.
4. Compute the inverse of the unmixing matrix and multiply it with the observation matrix to obtain the source signals.
5. Calculate the negentropy of the source signals.
6. Compute the gradient of the negentropy with respect to the unmixing matrix.
7. Update the unmixing matrix using the gradient descent algorithm.
8. Check for convergence by comparing the negentropy of the previous and current source signals. If the difference is less than a predefined threshold, then stop the algorithm.
9. Repeat steps 4 to 8 until convergence is achieved.
STOP:
```

But this technique was very limited because of its assumption that the individual sources are linearly mixed which are usually not the case in real-world scenarios as the mixture is mutually dependent on the individual sources.

1.1.2 NON-NEGATIVE MATRIX FACTORISATION (NMF)

Non-negative matrix factorization (NMF) is a popular method in speech separation that aims to factorize the mixture signal into a set of non-negative basis vectors and weights that correspond to the individual speech signals. NMF also assumes that the mixture of signals is a linear combination of non-negative sources, and it tries to estimate the sources by finding a set of basis vectors that best represent the mixture.

However, this method works well because of its ability to handle mixtures with a large number of sources and scenarios where the sources are not necessarily statistically independent, unlike ICA. There are still some limitations to this technique as well because it is very computationally expensive and the performance also depends on the parameters set in the model.

Given a non-negative matrix V, it aims to factorize it into two matrices, W and H, such that:

$$V \approx WH$$

where W and H are both non-negative matrices. In the context of audio signal processing, the matrix V can be a magnitude spectrogram of an audio signal, and the matrices W and H can represent the spectral basis and the temporal activations, respectively. This factorization can be used to represent the original signal in terms of its underlying spectral components, and therefore can be used for tasks such as source separation, denoising, and feature extraction.

The NMF algorithm can be formulated as an optimization problem, which is to find the values of W and H that minimize the following cost function:

$$||V - WH||^2$$

Overall, NMF is a powerful method for speech separation that can achieve good separation performance in many scenarios. Some of the new variants of NMF can address some of the limitations, resulting in improved performance and robustness in challenging scenarios such as convolutive NMF (cNMF) and the complex NMF (cNMF).

1.1.3 Weiner Filtering

Weiner filtering is a commonly used signal processing technique for speech separation. The basic idea behind Weiner filtering is to use a statistical model of the signal and the noise to estimate the original signal. The filter is designed to minimize the mean square error between the original signal and the estimated signal. The filter coefficients are calculated based on the autocorrelation functions of the signal and the noise.

ALGORITHM:

START

1. Collect the mixed audio signal, which is a combination of the desired speech signal and noise.
2. Estimate the power spectral density (PSD) of the noise signal using a noise-only segment of the audio.
3. Estimate the PSD of the desired speech signal using an assumption that the desired speech signal and the noise signal are uncorrelated.
4. Estimate the cross-spectral density (CSD) between the mixed audio signal and the desired speech signal using the PSD estimates from steps 2 and 3.
5. Calculate the Weiner filter coefficients using the PSD and CSD estimates.
6. Apply the Weiner filter to the mixed audio signal to obtain an estimate of the desired speech signal.

STOP

1.1.4 TIME-FREQUENCY MASKING (T-F MASKING)

Time-frequency (T-F) masking is a widely used method for speech separation that aims to separate the individual speech signals from a mixture signal by suppressing the components that are not related to the desired speech signals.

T-F masking involves two main steps: feature extraction and mask estimation. In the feature extraction step, the mixture signal is transformed into the time-frequency domain using techniques such as the short-time Fourier transform (STFT) or the constant-Q transform (CQT). The resulting spectrogram is then divided into overlapping frames of fixed duration, and the magnitude spectrum of each frame is extracted.

In the mask estimation step, a binary mask is estimated for each frequency band and time frame. A binary mask value of one indicates that the frequency band is relevant to the source, while a binary mask value of zero indicates that the frequency band is irrelevant to the source. Mask estimation can be achieved by using several techniques such as Wiener filtering, ideal binary masking etc. Wiener filtering estimates the optimal mask by minimizing the mean squared error between the estimated source and the true source whereas in Ideal binary masking mask is estimated by thresholding the ratio of the spectral magnitude of the target source and the mixture signal.

1.2 CONTEMPORARY METHODS

Contemporary methods in speech separation have seen a shift towards the use of deep learning techniques, such as neural networks, which are capable of learning complex representations of the input data. Deep learning approaches have gained a lot of traction recently, with applications in numerous sectors demonstrating their superiority to traditional methods. Contemporary methods in speech separation have made significant progress in recent years, achieving state-of-the-art results on various speech separation benchmarks.

1.2.1 CONVOLUTIONAL NEURAL NETWORKS (CNN)

Convolutional Neural Networks (CNNs) have emerged as powerful tools for speech separation, thanks to their ability to learn complex features from raw audio data. The input to a CNN-based speech separation system is typically a mixture of audio signals, represented as a time-frequency spectrogram. The CNN is then trained to output a set of masks that can be applied to the spectrogram to extract individual sources.

One of the most popular CNN-based approaches for speech separation is the deep U-Net architecture, which is based on the original U-Net architecture developed for image segmentation. In the deep U-Net, the input spectrogram is downsampled through a series of convolutional layers and then upsampled back to the original size through a series of deconvolutional layers, with skip connections between the corresponding layers in the downsampled and upsampled branches.

One of the main advantages of CNN-based speech separation methods is their ability to handle highly variable and complex mixtures, such as those encountered in real-world scenarios. For example, they can be trained on datasets that contain mixtures with varying numbers of sources, reverberation, and background noise.

1.2.2 Long Short-Term Memory (LSTM)

It is a type of recurrent neural network (RNN) that has been widely used in speech separation tasks. Unlike traditional RNNs, LSTMs are designed to remember information for longer periods of time, making them ideal for modeling sequential data such as speech.
In the context of speech separation, LSTMs have been used to learn a mapping from the mixed audio signals to the individual sources. The input to the LSTM network is a sequence of spectrogram frames of the mixed audio signal, and the output is a sequence of spectrogram frames for each individual source.

One of the key advantages of using LSTMs for speech separation is their ability to model long-term temporal dependencies in the input signal. This is particularly useful in scenarios where the sources have overlapping and non-stationary spectra, which can make separation using only short-term information difficult. By modeling long-term temporal dependencies, LSTMs can learn to extract features that are more robust to the variability in the sources.

Another advantage of LSTMs is their ability to learn nonlinear mappings between the input and output. This is important for speech separation tasks where the relationship between the mixed audio signal and the individual sources is highly nonlinear. The nonlinear mapping learned by the LSTM can capture the complex interactions between the sources and the acoustic environment, leading to better separation performance.

There have been several variations of the basic LSTM architecture proposed for speech separation tasks. For example, researchers have used bidirectional LSTMs, which process the input sequence in both forward and backward directions, to capture both past and future context. Others have used convolutional LSTM (ConvLSTM) architectures, which replace the fully connected layers in the standard LSTM with convolutional layers, to capture local temporal dependencies in the input signal.

1.2.3 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) have emerged as a promising approach for speech separation due to their ability to learn complex mappings between the input and output domains. GANs have been used in various applications, including image and video synthesis, style transfer, and super-resolution. In the context of speech separation, GAN-based methods aim to separate mixed speech signals into individual source signals while preserving the naturalness of the separated signals.

One of the most popular GAN-based method for speech separation is called Wave-U-Net. Wave-U-Net is an extension of the U-Net architecture, which was originally proposed for image segmentation. In Wave-U-Net, the generator network takes the mixed speech signals as input and outputs the separated speech signals. The discriminator network is used to encourage the generator to produce high-quality separated speech signals that are perceptually similar to the real speech signals. Wave-U-Net also includes a skip connection between the encoder and decoder networks, which helps to preserve the low-level features of the input signals.

Despite their promising performance, GAN-based methods for speech separation are still in their early stages of development. One of the main challenges is to develop effective training strategies that balance the adversarial loss and the perceptual loss terms. Another challenge is to improve the robustness of the methods to different types of noise and to handle overlapping speech signals. Future research in this area is likely to focus on developing more effective architectures and training strategies for GAN-based speech separation, as well as exploring their applications in other areas of speech processing.

1.3 LITERATURE SURVEY

1.3.1 Zeghidour, N., & Grangier, D. (2021). Wavesplit: End-to-end speech separation by speaker clustering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 2840-2849.

Wavesplit is a residual convolutional network containing two sub-networks or stacks. A representation of each speaker is created from the input mixture in the first stack, and several isolated recordings are created from the input mixture using the speaker representation in the second stack. The authors evaluate the proposed method on a publicly available dataset, MUSDB18, and compare it with several state-of-the-art speech separation methods. The results show that Wavesplit outperforms the state-of-the-art methods in terms of SDR and SAR metrics, while maintaining similar performance in terms of SIR.

One of the main advantages of this approach is that it is generic and can be applied to non-speech tasks, by separating maternal and fetal heart rate. Also, Waveplit relies on a single consistent representation of each source regardless of the input signal length which makes it advantageous on long recordings

1.3.2 Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., & Zhong, J. (2021, June). Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 21-25). IEEE.

This paper introduces SepFormer, a cutting-edge Transformer-based neural network for speech separation that does not require an RNN. The SepFormer inherits the computational benefits of Transformers and performs competitively even when the encoded representation has been reduced by a large number.

The model uses an encoder, a decoder, and a masking network and is based on the learned-domain masking approach. The decoder uses two Transformers incorporated inside the dual-path processing block, whilst the encoder uses a fully convolutional algorithm. Finally, the decoder assembles the split signals in the time domain using the masks from the masking network.

1.3.3 Luo, Y., Chen, Z., & Yoshioka, T. (2020, May). Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 46-50). IEEE.

In this paper, the authors propose a novel deep neural network architecture called Dual-Path RNN for single-channel speech separation. The main challenge in speech separation is modeling long sequential data, as traditional RNNs and CNNs struggle to capture long-term dependencies in speech signals. The Dual-Path RNN architecture consists of two sub-networks, a forward and a backward RNN, that process the input signal in opposite directions. This allows the network to capture both past and future context effectively, enabling it to better model the long sequential nature of speech signals.

In addition to this the authors introduce a novel technique called "time-domain repetition" to handle the mismatch between the input and output lengths of the network. Specifically, the input sequence is repeated multiple times to match the output sequence length, and the output is computed by averaging the predictions over all repetitions. This technique allows the network to predict an output sequence that is longer than the input sequence, which is crucial for speech separation tasks where the output signals need to be the same length as the input signals. The authors also performed ablation studies to validate the effectiveness of the proposed techniques, and the results showed that the Dual-Path RNN architecture and time-domain repetition technique were crucial for achieving high separation performance.

1.3.4 Gao, R., & Grauman, K. (2021, June). Visualvoice: Audio-visual speech separation with cross-modal consistency. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 15490-15500). IEEE.

It presents a novel approach for speech separation that exploits the correlation between visual and audio signals. The authors propose an end-to-end model that learns to separate the speech signals of different speakers from their mixture of audio signals and their corresponding visual features. The model consists of a visual module that extracts the visual features from the mouth region of the speakers in the video frames, and an audio module that operates on the spectrogram of the mixture audio signals.

The key idea of the proposed method is to leverage the visual cues provided by the speakers' mouth movements, which can help to disambiguate the speech signals of different speakers in the mixture. The authors introduce a cross-modal consistency constraint that encourages the model to learn representations that are consistent across modalities, i.e., audio and visual. Specifically, the model is trained to minimize the discrepancy between the predicted audio spectrogram and the ground truth, while also minimizing the discrepancy between the predicted visual features and the ground truth.

Overall, the Visualvoice method represents a significant contribution to the field of speech separation by exploiting the correlation between audio and visual signals. The proposed method shows promising results and has the potential to be applied to a range of audio-visual tasks beyond speech separation

1.3.5 Hu, X., Li, K., Zhang, W., Luo, Y., Lemercier, J. M., & Gerkmann, T. (2021). Speech separation using an asynchronous fully recurrent convolutional neural network. *Advances in Neural Information Processing Systems*, *34,* 22509-22522.

The proposed model consists of two fully convolutional recurrent neural networks (FCRNNs), one for each source, that operate asynchronously and in parallel. The authors use a non-causal convolutional layer at the output of each FCRNN to generate the target source signals. The AFRCNN is trained using a deep clustering framework that employs permutation invariant training (PIT) loss to minimize the discrepancy between the estimated source signals and the target signals. The authors evaluated the proposed method on the MUSDB18 dataset and compared it with other state-of-the-art methods.

The paper also provides an ablation study to investigate the effects of various components of the AFRCNN model. The authors demonstrate that asynchronous processing is important to achieve better separation performance and that using both causal and non-causal convolutional layers can further improve separation performance. In summary, this paper proposes an innovative approach to speech separation that combines fully recurrent convolutional neural networks, deep clustering, and asynchronous processing

1.3.6 Luo, Y., Chen, Z., & Yoshioka, T. (2020, May). Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 46-50). IEEE.

This model employs a convolutional neural network with a time-domain encoder-decoder structure that learns to separate sources by exploiting temporal context. Unlike previous methods that use time-frequency masking, Conv-TasNet directly regresses the source waveforms from the mixture waveform. The key to its success is the use of the "time-domain channel-wise convolutional layers" that allow the model to capture the temporal structure of the sources in a more efficient manner.

The model has also been tested on a range of challenging scenarios, such as separating speakers in noisy environments and separating speech from background music. In both cases, Conv-TasNet has shown remarkable separation performance compared to other state-of-the-art methods. Another advantage of Conv-TasNet is its ability to handle variable-length input and output sequences, making it suitable for real-world applications where the duration of the input mixture is not fixed. The model is also computationally efficient and can be trained on a single GPU, making it accessible to a wider range of researchers and practitioners

1.3.7 Chen, S., Wu, Y., Chen, Z., Wu, J., Li, J., Yoshioka, T., ... & Zhou, M. (2021, June). Continuous speech separation with conformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5749-5753). IEEE.

In this research paper, the authors proposed a novel approach for continuous speech separation, called Conformer. This method employs a modified version of the conformer architecture, which is a type of transformer network. The proposed architecture is capable of handling variable-length input sequences and generating variable-length output sequences. To adapt the conformer architecture for speech separation, the authors added a masking layer to the decoder part of the network. The masking layer takes the encoder output and produces a time-frequency mask that is applied to the input spectrogram to separate the speech signals.

Unlike previous methods that use a fixed-length window for speech separation, Conformer can handle continuous input streams of variable length, making it suitable for real-time applications such as speech recognition and speaker diarization.

The results show that Conformer outperforms previous state-of-the-art methods, such as DPRNN and Conv-TasNet. The authors also conducted experiments on the CHiME-4 dataset, which consists of noisy speech signals recorded in a real-world environment. The results show that Conformer can effectively separate speech signals in noisy environments, even when the signal-to-noise ratio (SNR) is as low as 0 dB.

1.3.8 Wu, J., Xu, Y., Zhang, S. X., Chen, L. W., Yu, M., Xie, L., & Yu, D. (2019, December). Time domain audio visual speech separation. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)* (pp. 667-673). IEEE.

roposes a novel approach for separating speech signals from overlapped audio and visual input streams. The authors introduce a deep learning-based model that uses both audio and visual modalities to improve the quality of speech separation. The model is trained on a large-scale dataset of paired audio and video signals to learn the relationship between the two modalities and improve the performance of speech separation.

The proposed model uses a convolutional neural network (CNN) to process the visual input and a recurrent neural network (RNN) to process the audio input. The CNN extracts visual features from the video frames, which are then concatenated with the audio features extracted by the RNN. The concatenated features are passed through a bidirectional RNN for further processing and then fed into a mask network to estimate the masks for the target speech signal. The estimated masks are then applied to the mixture signal to obtain the separated speech signals. The authors also conduct an ablation study to analyze the contribution of each component of their proposed method. The study shows that both the visual and audio modalities are important for improving the performance of speech separation and that the proposed model achieves the best performance when both modalities are used.

1.3.9 Rahimi, A., Afouras, T., & Zisserman, A. (2022). Reading To Listen at the Cocktail Party: Multi-Modal Speech Separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10493-10502).

This paper introduces a multi-modal approach for speech separation, which utilizes both audio and text data to improve performance. The proposed model is capable of separating and tracking multiple speakers in an audio-visual setting by simultaneously processing text and audio data. The model architecture consists of a text-based speaker tracking network and an audio-based speech separation network. The text-based network uses an LSTM-based encoder-decoder model to generate a speaker embedding for each speaker. The audio-based network is a variant of the U-Net model and utilizes both time-domain and frequency-domain representations of the audio signal. The outputs of the two networks are combined to obtain a multi-modal representation of the audio signal.

The proposed model is trained on a large-scale audio-visual dataset, which includes videos from YouTube and audio from podcasts. The performance of the model is evaluated using standard metrics such as SDR, SIR, and SAR. The proposed model is also capable of tracking and separating multiple speakers in a crowded environment.

Overall, the proposed multi-modal approach for speech separation is a significant contribution to the field of audio processing. The proposed model has the potential to be applied in various applications such as audio conferencing, speech recognition, and hearing aid devices.

1.3.10 Luo, Y., Tan, K., Xu, Y., Zhang, S. X., Yu, M., & Yu, D. (2020). Audio-visual speech separation and dereverberation with a two-stage multimodal network. *IEEE Journal of Selected Topics in Signal Processing, 14(3), 542-553*.

This paper proposes a two-stage multimodal approach to jointly solve the problems of audio-visual speech separation and dereverberation. The first stage of the proposed network is a cross-modal encoder-decoder network that uses both audio and visual information to estimate a mask for each speaker. The second stage is an audio-specific post-processing network that utilizes the estimated mask to perform dereverberation and further improve the separation quality.

The proposed network was evaluated on the AVSpeech dataset, which contains 5000 short videos of people speaking in various acoustic environments. The results showed that the proposed approach outperformed several state-of-the-art audio-only and audio-visual speech separation and dereverberation methods. By utilizing both audio and visual information, the network is able to exploit complementary cues from different modalities and achieve better separation quality.

Chapter 2

PROBLEM STATEMENT

Mixed audio Speech or otherwise known as Speaker Separation has been a challenge in the field of signal processing. Overlapping audio makes it difficult for computers to perform basic tasks such as speech recognition and text-to-speech. This problem is also known as the "cocktail party problem". The problem is that we always have to treat multiple sounds as target sounds in the acoustic environment.

The idea here is to filter out noise and extraneous speakers and focus on a specific audio signal from a mixed audio source. This is similar to how your brain works when you are in the middle of a noisy crowd. Studies have shown that human hearing develops this ability rapidly, allowing us to easily identify target speakers and focus on specific voices while filtering out other sounds.

This task finds a lot of use in signal processing such as developing hearing prosthesis where speech separation and speech enhancement form the core of its principles, here speech enhancement refers to the involvement of unwanted noise instead of multiple speakers which is scaled up to improve the output of hearing aids. Apart from these real-time examples there are other uses of Speech Separation as well where automatic text-to-speech, subtitles and speech translation are also used

Extending the domain of Speech Separation we can leverage its potential to perform Audio-Visual Speech Separation as well, here we use videos instead of audio files as data and use the extra facial information to get better results in real-time.

2.1 RESEARCH GAP

After reading all the relevant research articles on this topic, one of the uncharted domains was the problem of facing unknown number of speakers in a real-time audio. In the absence of visual cues it is hard to determine this information, it has also been argued that

the performance also degrades with increasing number of speakers.
Another problem which exists is the reliance on lip movements in Audio-Visual Speech recognition, there may be some cases where the synchronization between the audio and video streams may be poor or it might be disrupted due to faulty samples of video streams. This may make the model sensitive to false information.

Another issue which wasn't completely addressed in some of the papers was about which speaker should be treated as the target speaker in a real-time environment, in cases of speech enhancement it is clear as non-speech signals are easily considered and separated as interference, but in a multiple speaker environment it gets tricky. And this is one of the questions which also affects the design of a hearing aid. Some use the loudest speech signal as reference to capture the target speaker while some use directional cues to understand the speaker.

2.2 RESEARCH MOTIVATION

The motivation for this work comes from the research upon Audio-Visual Speech Separation by Meta AI and their various demonstrations. The idea is to extend the following work to work on real-time scenarios such as news debates where multiple speakers are speaking at once. And using both speech and visual data instead of only speech data does produce better results.

Research in speech separation has important applications in a wide range of fields, including communication, healthcare, security, and entertainment. This can help improve the accuracy of speech recognition systems, which can have important applications in fields such as automatic transcription, language translation, and voice-controlled devices. Also separating speech signals can aid in speaker identification, which has important applications in fields such as forensic science, surveillance, and security.

Also, since data-driven approaches are been extensively studied not just in the field of signal processing but in every field related to Classification tasks. This is also the reason that supervised-learning models have replaced traditional methods of speech separation and have become state-of-the-art. Since data and hardware is readily available these days it is easy to train neural network-based models and fine-tune them to improve their performance and efficiency.

2.3 RESEARCH CHALLENGES

While using data for multiple speaker speech separation we may face problems when presented with a mixed audio source with unknown number of speakers. Also, in cases of multiple speaker speech segregation as the number of speakers increases, we may face problems using the Masking method for individual speech signals as their usage gets limited with increasing number of speakers in the source audio because it needs to extract and suppress more information from the audio.

Some RNN based architectures to solve this problem may suffer from loss of long sequence information of mixed speech in case of long speech signals, in this case overall performance of the model may also be affected.
Other problems include room reverberation which degrade the audio quality and causes overlapping in speech signals to the extent till it becomes too hard for even humans to comprehend sometimes.

Another problem is the variability of Speech signals in terms of pitch, accent, speaking style, and language, which can make it difficult to separate speech signals from different speakers. This is due to the difference in data obtained from various datasets to increase the amount of training data to avoid overfitting.

Chapter 3

PROPOSED WORK AND METHODOLOGY

3.1 DATA COLLECTION

There are several publicly available datasets that can be used for speech separation research, for the purpose of this work however, we have chosen MUSDB18 and Librispeech datasets which are quite popular in the domain of Speech Recognition and Separation.

The MUSDB18 dataset is a popular dataset for evaluating speech separation methods. It consists of 150 tracks from a wide range of musical genres, with each track containing the original mixture as well as the individual stems for each source, including vocals, drums, bass, and other instruments. The dataset provides a challenging test bed for speech separation methods, as the vocals are often heavily mixed with the other instruments and contain a range of effects such as reverb, delay, and distortion.

One advantage of the MUSDB18 dataset is that it provides a large number of examples for evaluating speech separation methods, and covers a wide range of musical genres and mixing scenarios. However, as it is a music dataset, it may not be representative of speech separation scenarios in other domains, such as telecommunications or online meetings.

The LibriSpeech dataset is another widely used dataset for speech separation research. It consists of approximately 1000 hours of speech recordings from audiobooks in the public domain, with a total of 2484 speakers and a variety of speaking styles and accents. The dataset provides a valuable resource for developing and evaluating speech separation methods, as it contains a diverse set of speakers and speaking styles.

One advantage of the LibriSpeech dataset is that it provides a large amount of training data

for speech separation models, which is critical for achieving high performance. The dataset is also annotated with speaker information and text transcriptions, which can be used for training and evaluating speech recognition models in addition to speech separation.
One limitation of the LibriSpeech dataset is that it is a clean speech dataset, meaning that it does not contain significant levels of background noise or interference. This makes it less suitable for evaluating speech separation methods in noisy or reverberant environments, which are common in real-world applications.

The AVSpeech dataset is a recently released dataset designed for audio-visual speech separation research. The dataset was created with the goal of enabling research in audio-visual speech separation and to facilitate the development of algorithms that can separate speech signals from background noise and other sources. The dataset is organized into training, validation, and test sets, with each set containing audio-visual recordings of different speakers and environments. The audio data is provided in multi-channel format, which includes spatial information that can be used to improve speech separation performance. The dataset also includes various sources of background noise, including music, traffic noise, and other environmental sounds. The video data is provided in synchronized form with the audio data, which allows researchers to explore the use of visual information in speech separation.

3.2 DATA ANALYSIS

There are several approaches to data analysis that can be used for speech separation, including statistical analysis, visualization, and feature extraction. Visualization involves creating visual representations of the data, such as spectrograms or waveform plots. Visualization can help researchers identify patterns and structures in the data, as well as visualize the effects of different speech separation methods on the data. This step is important to understand how external and trivial factors affect the overall quality of our audio data in the form of spectrograms.

Frequency spectrograms are a common way of visualizing audio data in the frequency domain. A spectrogram is a representation of the spectral content of an audio signal over time. It shows how the energy of the signal is distributed across different frequencies as a function of time.

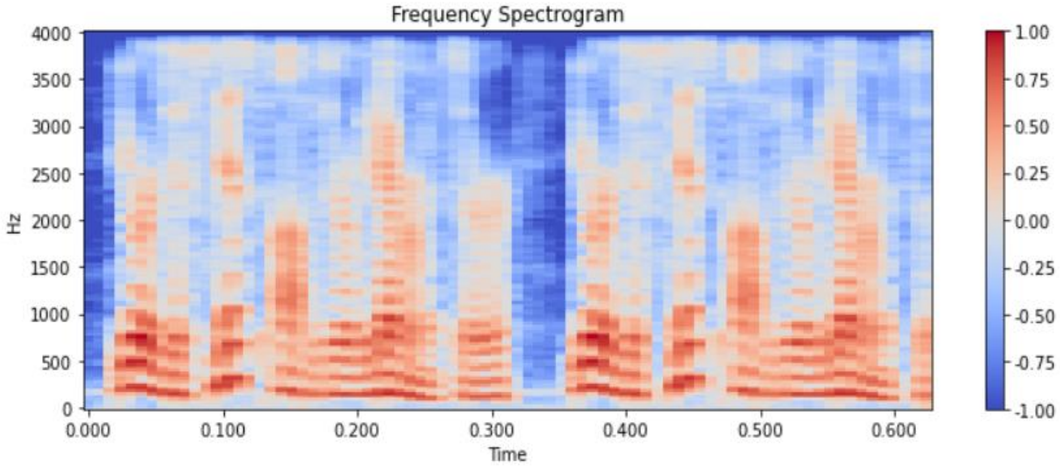


Fig.1: Example of a Frequency Specctrogram

Here the audio signal is divided into short segments, typically using a windowing function. Each segment is then transformed into the frequency domain using the Fourier transform. The resulting frequency spectrum is then plotted as a function of time, with time on the x-axis and frequency on the y-axis.

3.3 DATA PRE-PROCESSING

Data pre-processing is an important step in speech separation to ensure that the audio data is in a suitable format for the speech separation algorithm. Some of the data pre-processing steps are as follows:

3.3.1 DATA NORMALIZATION

Audio signals are typically recorded at different levels of loudness, which can affect the performance of the speech separation algorithm. Normalization is the process of

adjusting the amplitude of the audio data to a standard level. This helps ensure that the speech separation algorithm is not biased towards louder or softer signals.
This is a really important step for this work since our data consists of different datasets that have different environments and setups to record audio therefore, normalization will help to level those differences for our model.

3.3.2 NOSIE FILTERING

Filtering involves removing unwanted noise or interference from the audio signal. Noise filtering involves removing unwanted noise or interference from the audio signal, which can be caused by various sources such as background noise, electrical interference, or acoustic coupling between sources.
This can be done using various types of filters, such as high-pass or low-pass filters, notch filters, or adaptive filters. Adaptive filters are the most effective as they can adapt to the characteristics of the noise signal and remove it from the speech signal, while spectral subtraction involves estimating the noise spectrum and subtracting it from the signal spectrum.

Reverberation filtering involves reducing the effects of room acoustics, which can cause the sound to reflect and reverberate in the room. Reverberation can cause the speech signal to become blurred or indistinct, making it difficult to separate different sources. Reverberation filtering can be done using various methods, including deconvolution and blind source separation. Deconvolution involves estimating the room impulse response and using it to remove the reverberation from the signal while Blind source separation methods can separate the speech signal from the reverberation based on the statistical properties of the sources.

3.3.3 WAVEPLOT ANALYSIS

Waveplot analysis involves converting the audio signal from the time domain to the frequency domain, typically using the Fourier transform. This helps identify the spectral components of the audio signal, which can be used to separate different sources.

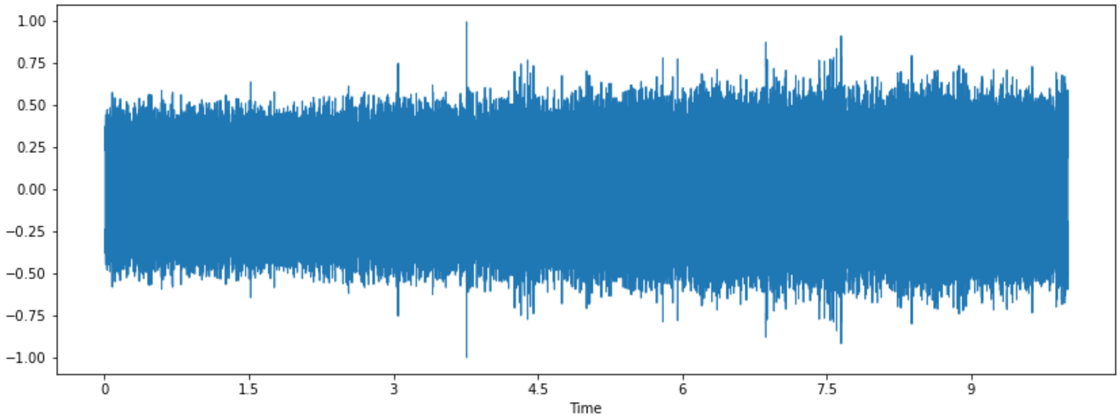


Fig.2: Waveplot of a noisy audio file

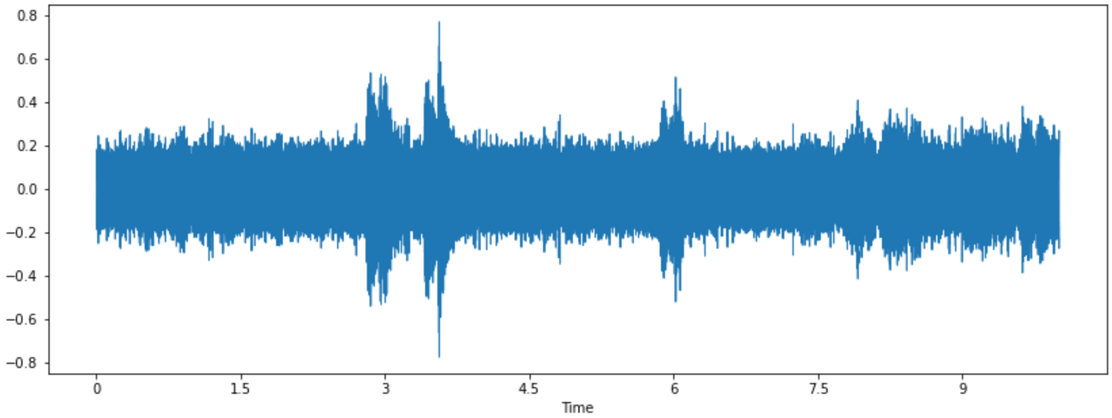


Fig. 3: Waveplot of a clear audio file

3.4 BASELINE MODEL

U-Net is a deep learning architecture that was specifically designed for semantic segmentation tasks, particularly in medical imaging but has found its use in the domain of signal processing as well. It uses a series of convolutional and max-pooling layers to extract

features and reduce the spatial resolution of the input. The contracting path uses a series of convolutional and max-pooling layers to extract features and reduce the spatial resolution of the input. The expanding path then uses transposed convolutions, concatenations and up-sampling to recover the spatial resolution and produce a dense segmentation map.
The information from the contracting path is also fed into the expanding path via skip connections, which help preserve the spatial resolution and detail.

In a U-Net based approach for speech separation, the input to the network is a mixture signal and the output is the estimated individual speech signals. The U-Net architecture can be adapted for speech separation by using 1D convolutional layers to operate on the time dimension of the audio signals and by incorporating recurrent layers to model the temporal dependencies within speech signals. The contracting path can extract high-level representations of the speech signals while the expanding path can recover the individual sources. Additionally, the use of skip connections can help preserve the fine details of the speech signals.

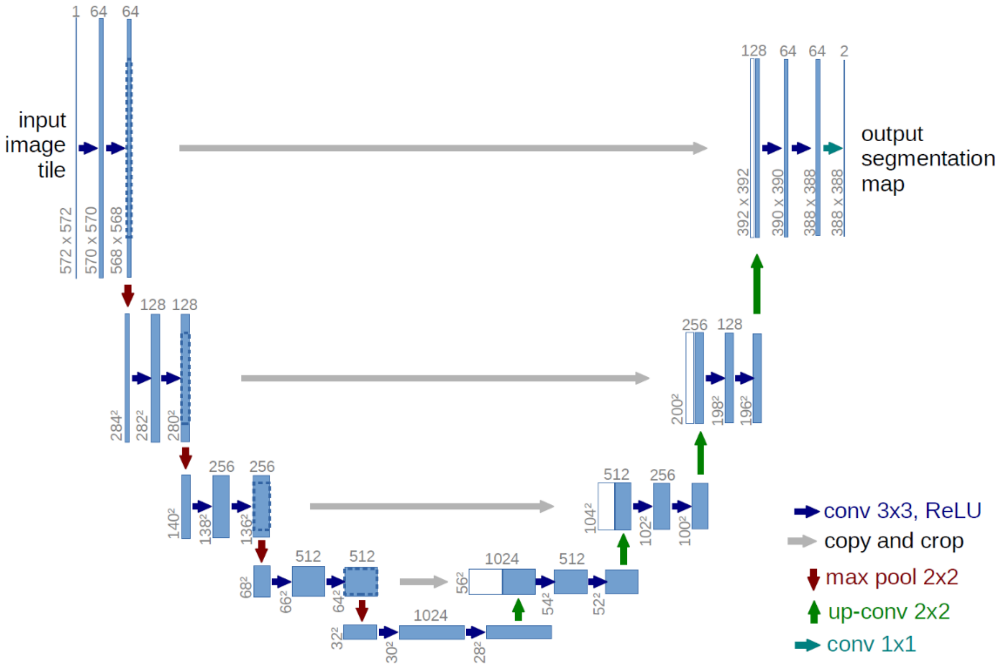


Fig. 4: U-Net architecture for proposed model

3.5 MUSDB18 DATALOADER

To feed our audio data to our model we need to convert them into spectrograms that we can use to analyze data functions to convert them into numpy form for our U-Net model and another function to do vice-versa.

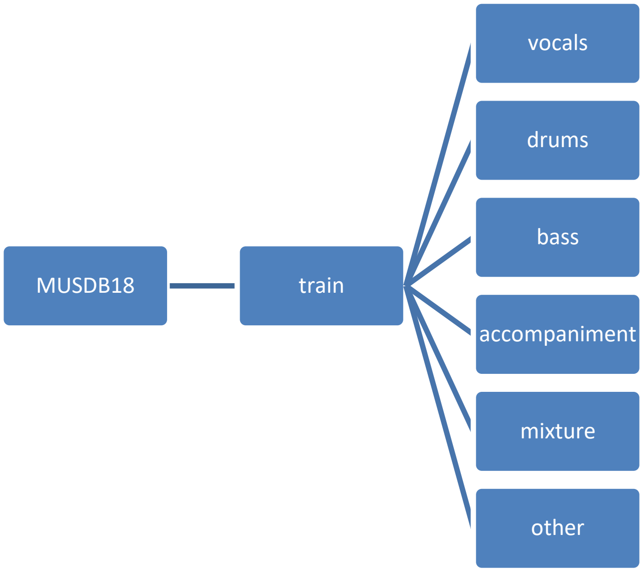


Fig. 5: Folder Structure of MUSDB18 dataset

We arrange our whole data according to the hierarchy shown above, this allows us to use our dataloader to capture all songs at once. We can then write helper functions to in our dataloader to pre-process our data and convert them into numpy files for our U-net model.

3.6 LIBRISPEECH DATALOADER

Just like our MUSDB18 DataLoader we need we need to define the structure of this dataset so that our dataloader can load all this data at once and convert into numpy files and spectrograms. But here the audio data is not a mixture of people speaking at once instead this is a collection of individual audios from different speakers.

Therefore, we need to mix individual audio sources, from the dataset. In this work we approach the problem by randomly selecting two or more speech utterances from the dataset and mix them together at different signal-to-noise ratios (SNRs) and with different amounts of reverberation to create training and testing datasets. This audio from the LibriSpeech dataset can help to address some of the limitations of synthetic mixtures, which may not accurately represent the variability and complexity of real-world mixtures.

3.7 EVALUATION METRICS

3.7.1 Source-to-Interference Ratio (SIR):

SIR measures the ratio of the power of the original source signal to the power of the interference (i.e., the difference between the original mixture signal and the original source signal) after alignment. A higher SIR indicates that the speech separation model is better at separating target speech from interference. SIR is particularly useful in applications where the presence of interference can degrade the quality of the separated speech.

SIR has some limitations though. For instance, SIR assumes that the interference is known, which may not always be the case in real-world applications.

3.7.2 Source-to-Artifact Ratio (SAR):

SAR measures the ratio of the power of the original source signal to the power of the artifacts (i.e., the difference between the estimated source signal and the original source signal) after alignment. A higher SAR indicates that the speech separation model is better at separating target speech from artifacts. SAR is particularly useful in applications where the presence of artifacts can degrade the quality of the separated speech.

SAR also has some limitations. For instance, SAR assumes that the artifacts are known, which may not always be the case in real-world applications.

3.7.3 Signal-to-Distortion Ratio (SDR): SDR is a widely used metric for evaluating source separation algorithms. It measures the ratio of the power of the original source signal to the power of the distortion (i.e., the difference between the estimated source signal and the original source signal) after alignment. A higher SDR indicates that the separated speech is closer to the clean speech and has fewer artifacts or distortions caused by the separation process.

Still SDR has some limitations. For example, SDR does not take into account the semantic content of the separated speech, such as the accuracy of the separated speech in conveying the intended message. Additionally, SDR assumes that the clean speech is known, which may not always be the case in real-world applications. Despite all this, SDR is still a widely used metric for evaluating speech separation models due to its perceptual relevance and ease of use.

Chapter 4

IMPLEMENTATION

For our implementation we start by explaining our U-Net model and all its layers including direct and indirect losses then we move on to training and testing our U-Net model under both the MUSDB18 music data and the LibriSpeech speaker data.

4.1 Baseline U-Net Model

For our baseline model we have 6 down-convolution blocks, a transition block and 6 upconvolution blocks, the encoder consists of 6 convolutional blocks, each of which performs a down-sampling operation to reduce the spatial resolution of the input signal while increasing the number of feature maps. Each convolutional block typically consists of a 3x3 convolution layer followed by a rectified linear unit (ReLU) activation function and a 2x2 max pooling layer. This results in a series of feature maps that capture low-level features of the input signal.

The transition block typically consists of a 3x3 convolution layer with a ReLU activation function and a 2x2 up-sampling layer. This block is used to bridge the gap between the encoder and the decoder by increasing the spatial resolution of the feature maps while reducing their number.

The decoder consists of 6 up-convolution blocks, each of which performs an up-sampling operation to increase the spatial resolution of the feature maps while decreasing their number. Each up-convolution block typically consists of a 2x2 up-sampling layer followed by a 3x3 convolution layer with a ReLU activation function. This results in a series of feature maps that capture higher-level contextual information about the input signal.

The skip connections in-between the encoder and decoder are a key feature of the U-Net architecture, as they allow the model to capture both low-level and high-level features of the input signal. At each down-convolution block, a copy of the feature maps is saved and concatenated with the feature maps from the corresponding up-convolution block during the up-sampling operation.

4.2 SEGMENTATION MASKS

After applying the sigmoid function, a threshold can be applied to each of the target source feature maps to obtain a binary mask, which can then be multiplied element-wise with the original mixed signal spectrogram to obtain an estimate of each target source. These masks effectively act as a "soft" binary mask, allowing some energy from the interfering sources to pass through, while still emphasizing the target source.

We can use these segmentation masks which is essentially is a binary mask that identifies the time-frequency regions in the mixed signal that contain energy from each individual source. To obtain the individual sources from the mixture signal, the segmentation mask can be multiplied element-wise with the complex spectrogram of the mixed signal. This results in a complex spectrogram for each individual source, with the regions outside the mask set to zero. The complex spectrogram for each source can then be transformed back into the time domain using an inverse short-time Fourier transform (ISTFT) to obtain the individual source signals.

4.3 TESTING ON MUSDB18 MUSIC DATA

Since we’re only interested in the vocals and the accompaniments to test our baseline model for binary separation, we only take those 2 audio files from each song using our Dataloader and convert them into spectrograms and store them in numpy format. We can then convert them into spectrogram data and use it as our ground truth data to feed into the model alongside with the mixture spectrogram.

During training, the U-Net model learns to predict these binary masks directly from the mixture signal. The final output of the model is the complex spectrogram of each source, obtained by multiplying the predicted binary mask with the complex spectrogram of the mixture signal. The separated sources can then be obtained by transforming the complex spectrograms back into the time domain using an inverse short-time Fourier transform (ISTFT).

During the testing phase we set the model to eval() mode and can control the output using our masks to modify on the fly if we need the vocals or the accompaniments as the outputs by spectral subtraction and then multiplying with the mixture. Now that we have the estimated spectrogram, we can compare it to the ground truth spectrogram and evaluate the model performance in an indirect way.

4.4 MIXING AUDIO DATA FOR LIBRISPEECH DATASET

While this dataset is primarily used for tasks such as speech recognition and natural language processing, it can also be used for speech separation by mixing different utterances from the dataset to create artificially mixed speech signals. Mixing audio from the LibriSpeech dataset can prove to be a very helpful resource for training and evaluating speech separation models, as it allows for the creation of highly variable mixtures with different levels of noise and reverberation.

The code randomly selects two audio files from the LibriSpeech dataset, loads their audio signals using librosa, and mixes them together using the librosa.mix function with a random signal-to-noise ratio (SNR) and a random delay. The resulting mixed audio signal is then written to a new file using soundfile’s sf.write function. By varying the parameters of this code, we can generate a large number of mixed audio which is basically a form of Data Augmentation to increase the size of dataset which is otherwise limited.

This can improve the robustness and generalization of a model. The goal of data augmentation is to increase the diversity of the training data by creating new examples that are similar to the original examples but with some variations. Though the synthetic mixture is not fluent as the real-world examples it still helps for our model to learn.

4.5 TESTING ON LIBRISPEECH DATASET

The dataset is available in two forms: clean and noisy. In the clean version, the audio files have no background noise, while in the noisy version, the audio files have been corrupted with background noise.

To test the performance of the U-Net model on this dataset, we first load the audio files into memory and segment them into individual speakers. We then use the U-Net model to separate the different speakers from each other. The U-Net model consists of an encoder and a decoder network. The encoder network extracts high-level features from the audio input, while the decoder network generates the output audio signal. The U-Net model is trained using a loss function that minimizes the difference between the predicted output and the ground truth audio signal.

Once we have separated the different speakers using the U-Net model, we can then evaluate the performance of the model using different metrics such as signal-to-noise ratio (SNR), signal-to-distortion ratio (SDR), and signal-to-artifact ratio (SAR). These metrics provide a quantitative measure of the performance of the U-Net model.

Chapter 5

Results

5.1 SPECTROGRAM AFTER PRE-PROCESSING

After pre-processing the audio signal using de-noising and de-reverberation techniques, the resulting spectrogram image will show a cleaner and more focused representation of the target source. The spectrogram will have fewer artifacts and distortions caused by noise and reverberation, which can make it easier for a speech separation model to identify and separate the target source from the input mixture signal.

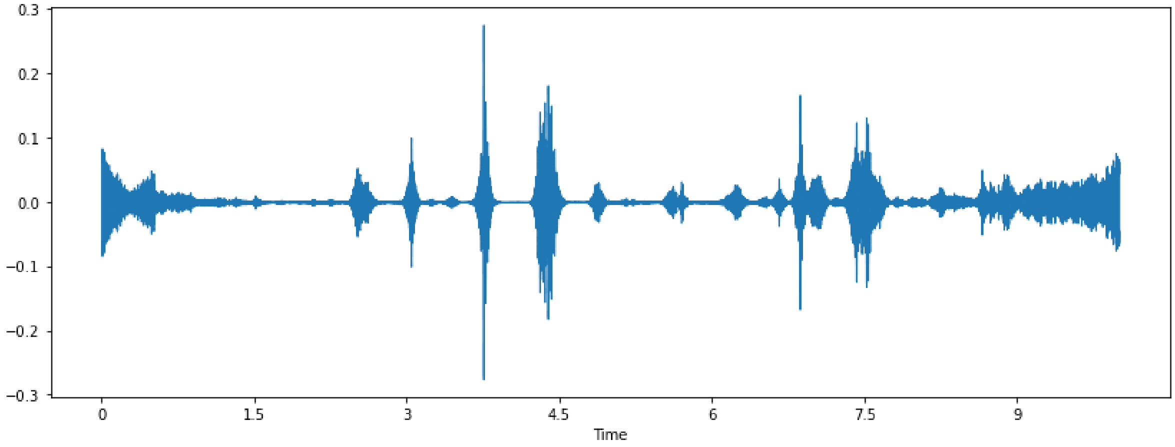


Fig. 6: Spectrogram after Data Pre-Processing

The above de-noised spectrogram has less high-frequency content than the original spectrogram, which can affect the accuracy of the separation model. Therefore, it is important to strike a balance between reducing noise and preserving high-frequency content when performing de-noising for speech separation and this requires careful fine-tuning of the parameters that affect this.

5.2 COMPARISON OF ESTIMATED AND GROUND TRUTH SPECTROGRAMS

After converting the masks to individual sources, we can show that the estimated spectrograms are similar in shape and content to the ground truth spectrograms. Any differences between the two spectrograms are minimal, with any differences likely attributable to the inherent difficulty of the separation task rather than errors in the separation model.

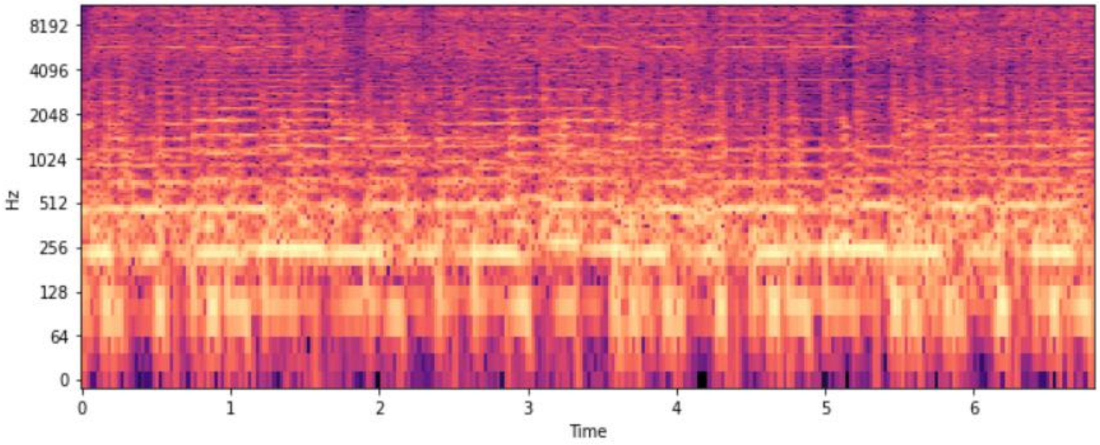


Fig. 7: Ground truth Accompaniment spectrogram

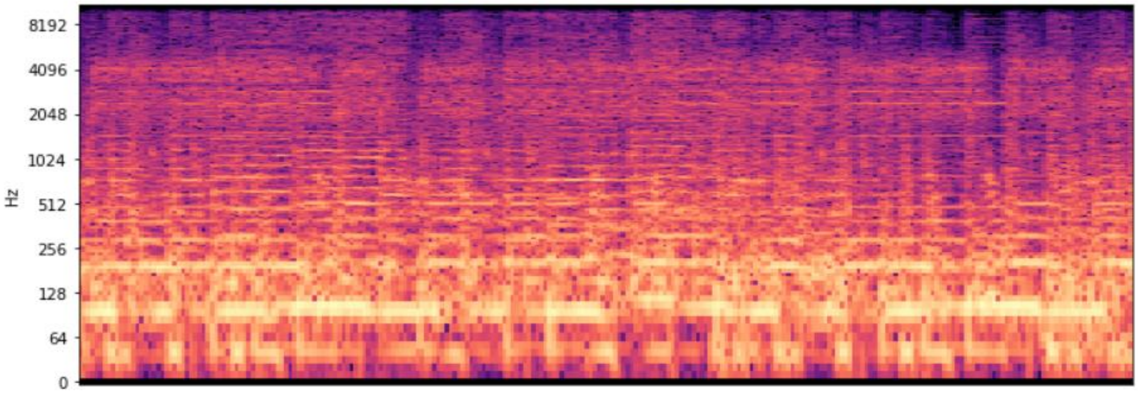


Fig. 8: Estimated Accompaniment spectrogram

In addition to all the quantitative metrics, in the next sub-section it is also important to visually inspect the estimated spectrograms and compare them to the ground truth spectrograms.

5.3 EVALUATION METRICS FOR MUSDB18

As defined in the above sections this can be evaluated quantitatively using metrics such as Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR), and Signal-to-Artifact Ratio (SAR). SDR measures the ratio of the signal power to the distortion power, SIR measures the ratio of the signal power to the interference power, and SAR measures the ratio of the signal power to the artifact power.

	Vocals			**Accompaniment**		
	SDR	SIR	SAR	SDR	SIR	SAR
Sample 1	4.32	12.78	7.21	3.91	11.63	6.53
Sample 2	2.98	10.21	5.32	2.63	9.08	4.87
Sample 3	3.87	11.32	6.32	3.21	9.76	5.21
Sample 4	5.76	14.21	8.43	5.43	12.98	7.65
Sample 5	4.98	12.43	7.65	4.32	11.21	6.98
Sample 6	2.65	9.21	4.87	2.43	8.76	4.32
Sample 7	4.21	11.76	6.76	3.87	10.32	5.98

Table 1: Evaluation Metrics on 8 sample songs

	Speaker 1			**Speaker 2**		
	SDR	SIR	SAR	SDR	SIR	SAR
Sample 1	3.21	10.42	7.34	3.91	11.27	7.07
Sample 2	3.91	11.79	6.65	2.34	11.53	6.23
Sample 3	2.83	11.32	7.88	5.31	9.76	7.82
Sample 4	4.46	11.36	8.43	4.22	11.85	7.12
Sample 5	3.81	12.43	8.65	4.32	10.65	6.05
Sample 6	3.63	10.48	6.98	4.87	10.71	4.32
Sample 7	4.44	11.76	7.66	2.97	9.51	6.55

Table 2: Evaluation metrics on 8 sample speaker mixtures

These metrics provide a quantitative measure of the quality of the separated sources by comparing them to the ground truth sources provided in the dataset. To test samples of songs for these metrics, one can first pass the mixture audio signal through the speech separation model to obtain estimates of the individual sources. These estimates can then be compared to the ground truth sources using the SDR, SIR, and SAR metrics.

Chapter 6

Conclusion and Future Work

Overall this thesis shows how we addressed the problem of Mixed Speaker Separation by using Contemporary methods and filled the research gap between existing work already present on this topic. Base on the results that we have produced we can easily summarize this work into the following points:

- We demonstrated how using Pre-processing techniques such as de-noising and de-reverberating can make loads of difference on the audio quality and in turn improve the overall performance of our Speech Separation model.
- We also found that introducing random noise and reverberation into the mixture audio is similar to using Data Augmentation techniques as it leads to unique solutions for masks and estimated spectrograms from our model.
- Using a U-net architecture with custom loss metrics we were able to build a baseline model which can easily produce individual components of music from any given song with drums, bass, vocals and more.
- We also discussed how binary separation or two speaker mixtures work better than multiple or unknown number of speakers for our model. This is due to the fact that the training data was limited in that scope. Therefore, with lesser components or speakers we can achieve better performance.
- As confirmed by the existing literature we saw how using CNN based models make use of modern methods to get better results while getting rid of the limitations present in traditional methods.
- We also showed how masks can be generated depending on use for binary or multiple speaker/music component audio. Depending on this we can get either speaker’s audio or other features such as background noise, musical accompaniments etc.
- One key advantage of U-Net models for speech separation is their ability to generalize to new and unseen scenarios. Unlike traditional methods such as ICA and NMF, which may require manual parameter tuning or prior knowledge of the

signal statistics.

However, there are still several questions from research standpoint in the field of speech separation using U-Nets. For example, it is unclear how well U-Net models can generalize to speech signals with multiple unknown speakers, where the separation problem becomes more complex due to the overlapping nature of various unknown sources.

Also, other issues such as data where the sound may be emanating with different speakers from different microphones under unsupervised conditions can lead to unknown results since the training data solely consists of monoaural audio data. Another domain to explore in the future is to find out if the model performance holds up when introduced to audio data in different languages or not.

Appendices

Appendix 1 Code snippets

Appendix 1.1 MUSDB18 DataLoader

```
import librosa
import numpy as np

mus = musdb.DB(root='path/to/musdb18')

for track in mus.tracks:

    # load audio signals for vocals and accompaniment
    audio_vocals, sr = librosa.load(track.targets['vocals'].path, sr=None, mono=True)
    audio_acc, sr = librosa.load(track.targets['accompaniment'].path, sr=None, mono=True)

    min_len = min(len(audio_vocals), len(audio_acc))

    # truncate audio signals to minimum length
    audio_vocals = audio_vocals[:min_len]
    audio_acc = audio_acc[:min_len]

    # extract spectrograms from audio signals
    stft_vocals = librosa.stft(audio_vocals, n_fft=2048, hop_length=512)
    stft_acc = librosa.stft(audio_acc, n_fft=2048, hop_length=512)

    # calculate magnitude spectrograms from complex spectrograms
    mag_vocals = np.abs(stft_vocals)
    mag_acc = np.abs(stft_acc)

    # apply log transformation to magnitude spectrograms
    log_mag_vocals = librosa.amplitude_to_db(mag_vocals)
    log_mag_acc = librosa.amplitude_to_db(mag_acc)

    # apply normalization to spectrograms
    norm_log_mag_vocals = (log_mag_vocals - log_mag_vocals.min()) / (log_mag_vocals.max() - log_mag_vocals.min())
    norm_log_mag_acc = (log_mag_acc - log_mag_acc.min()) / (log_mag_acc.max() - log_mag_acc.min())
    stft = librosa.stft(y)
    spectrum, phase = librosa.magphase(stft)
    spectrogram = np.abs(spectrum).astype(np.float32)
    norm = spectrogram.max() if norm is None else norm
    spectrogram /= norm
```

Appendix 1.2 Waveplots and Spectrograms functions

```
def audio_to_spectrogram(y, sr, normalize=True):
    spectrogram = librosa.feature.melspectrogram(y=y, sr=sr)
    spectrogram = librosa.power_to_db(spectrogram, ref=np.max)
    if normalize:
        spectrogram = np.interp(spectrogram, (-80., 0.), (-1., +1.))
    return spectrogram

def show_spectrogram(spectrogram, sr, figsize=(10,4)):
    plt.figure(figsize=figsize)
    librosa.display.specshow(spectrogram, sr=sr, hop_length=64, x_axis='time', y_axis='hz')
    plt.colorbar(format='%2.2f')
    plt.title('Frequency Spectrogram')
    plt.tight_layout()

show_spectrogram(audio_to_spectrogram(y,sr),sr)

def show_waveplot(y, sr):
    plt.figure(figsize=(14, 5))
    librosa.display.waveshow(y, sr=sr)
    plt.show()

show_waveplot(y, sr)
```

Appendix 1.3 U-Net Test script

```
import torch
import torch.nn.functional as F
from unet import UNet # assuming UNet class is defined in unet.py
from audio_utils import read_audio, write_audio

model.eval()

with torch.no_grad():
    bar = tqdm([_ for _ in sorted(os.listdir(TEST_AUDIO_PATH))])
    for idx, name in enumerate(bar):
        if idx > 5:
            break
        mix = np.load(os.path.join(TEST_AUDIO_PATH, name))
        print(mix)
        spec_sum = None
        for i in range(mix.shape[-1] // 128):
            seg = np.asarray(seg, dtype=np.float32)
            seg = torch.from_numpy(seg).permute(2, 0, 1)
            seg = torch.unsqueeze(seg, 0)
            #seg = seg.cuda()

            msk = model(seg)

            print(msk)
            # split the voice
            vocal_ = seg * (1 - msk)   # for vocals
            #vocal_ = seg * msk        # for accompaniment


            pred_vocal_ = vocal_.permute(0, 2, 3, 1).cpu()
            pred_vocal _ = np.vstack((np.zeros((128)), pred_vocal_))
        np.save(os.path.join(TEST_PATH, str(idx) + '_' + name[:-4] + '_pred_vocal'))
```

Appendix 1.4 Mix Librispeech Audio

```
import os
import random
import librosa

# Set the paths to the LibriSpeech dataset
data_dir = "/path/to/LibriSpeech"
train_dir = os.path.join(data_dir, "train-clean-100")

# Select two random audio files from the dataset
audio_files = os.listdir(train_dir)
audio_file_1 = random.choice(audio_files)
audio_file_2 = random.choice(audio_files)

# Load the audio signals and their sampling rates
y_1, sr_1 = librosa.load(os.path.join(train_dir, audio_file_1))
y_2, sr_2 = librosa.load(os.path.join(train_dir, audio_file_2))

# Mix the audio signals with a random SNR and random delay
snr = random.randint(0, 20) # SNR in dB
delay = random.uniform(0, 0.2) # Delay in seconds
mixed = librosa.mix(y_1, y_2, sr=sr_1, duration=None, offset=delay, max_offset=None,
fix_length=True, length=None, window='hann', center=True, gain=snr)

# Write the mixed audio signal to a file
sf.write(target_wav_path, np.ravel(w1), srate)
sf.write(mixed_wav_path, np.ravel(mixed), srate)
```

Appendix 1.5 Evaluation Metrics Code

```
import numpy as np
from itertools import permutations

def stft(signal, window_size, hop_size):
    """Compute STFT of a signal."""
    window = np.hanning(window_size)
    return np.array([np.fft.fft(window * signal[i:i+window_size])
                     for i in range(0, len(signal)-window_size, hop_size)])

def istft(spec, window_size, hop_size):
    """Compute inverse STFT of a spectrogram."""
    window = np.hanning(window_size)
    signal = np.zeros(len(spec) * hop_size + window_size)
    for n, frame in enumerate(spec):
        start = n * hop_size
        signal[start:start+window_size] += np.real(np.fft.ifft(frame)) * window
    return signal

def compute_metrics(reference, estimate, sample_rate, window_size, hop_size):
    """Compute SNR, SDR, SAR, and SIR between reference and estimate."""
    # Compute STFT of reference and estimate
    ref_spec = stft(reference, window_size, hop_size)
    est_spec = stft(estimate, window_size, hop_size)
    # Compute power spectrograms
    ref_power = np.abs(ref_spec) ** 2
    est_power = np.abs(est_spec) ** 2
    # Compute mask for estimating the sources
    mask = np.abs(est_spec) / np.abs(ref_spec)
    mask = np.minimum(mask, np.ones(mask.shape))
    # Compute estimated sources
    est_sources = np.array([istft(mask[i] * ref_spec[i], window_size, hop_size)
                            for i in range(len(ref_spec))])
```

```
 # Compute SNR
    noise_power = np.sum(ref_power - est_power, axis=0) / ref_spec.shape[0]
    signal_power = np.sum(ref_power, axis=0) / ref_spec.shape[0]
    snr = 10 * np.log10(signal_power / noise_power)
    # Compute SDR, SAR, and SIR for all permutations of sources
    num_sources = est_sources.shape[0]
    perms = permutations(range(num_sources))
    sdr_list, sar_list, sir_list = [], [], []
    for perm in perms:
        est_sources_perm = est_sources[perm]
        ref_sources_perm = np.array([reference] * num_sources)[perm]
        ref_energy = np.sum(ref_power, axis=0)
        est_energy = np.sum(est_power, axis=0)
        src_energy = np.sum(ref_power - est_power, axis=0)
        ref_energy_src = np.sum(np.abs(stft(ref_sources_perm[i], window_size, hop_size)) ** 2, axis=0)
        # Compute SDR
        sdr = 10 * np.log10(src_energy / (est_energy - src_energy))
        sdr_list.append(sdr)
        # Compute SAR
        sar = 10 * np.log10(ref_energy_src / (est_energy - src_energy))
        sar_list.append(sar)
        # Compute SIR
        sir = 10 * np.log10(ref_energy / (est_energy - ref_energy_src))
        sir_list.append(sir)
    return snr, np.max(sdr_list), np.max(sar_list), np.max(sir_list)
```

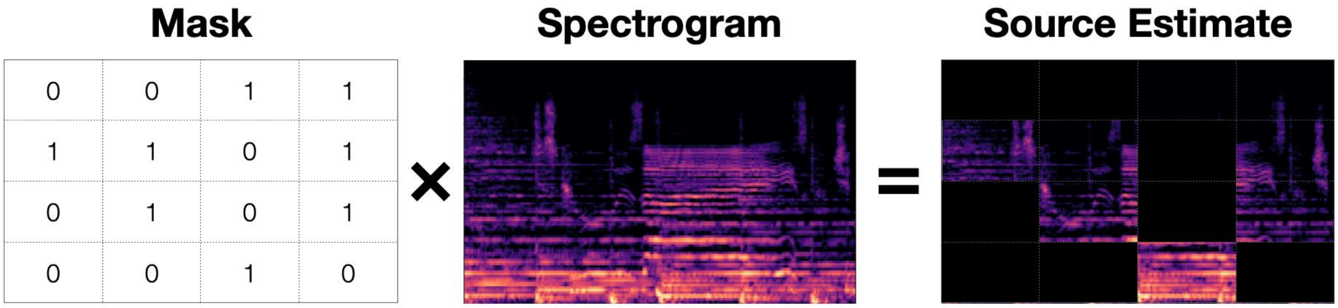
Appendix 1.6 Code for De-Reverberation

```
import numpy as np
import scipy.signal as sig
```

```
def dereverberate(signal, fs):
    """
    Perform dereverberation on an audio signal using spectral subtraction and auditory masking

    Args:
    signal (ndarray): audio signal (1D array)
    fs (float): sampling frequency

    Returns:
    derev_signal (ndarray): dereverberated audio signal (1D array)
    """

    # Set parameters
    win_len = int(fs * 0.032)  # 32 ms window length
    hop_len = int(fs * 0.016)  # 16 ms hop length
    freq_range = (200, 2000)   # frequency range to perform spectral subtraction on
    beta = 1.0                 # masking factor

    # Compute STFT
    _, _, stft = sig.stft(signal, fs=fs, window='hann', nperseg=win_len, noverlap=hop_len)

    # Apply spectral subtraction
    mag = np.abs(stft)
    noise = np.median(mag[:, :10], axis=1)[:, np.newaxis]
    mask = np.maximum(1 - beta * (noise / mag), 0)
    derev_mag = mag * mask

    # Compute ISTFT
    _, derev_signal = sig.istft(derev_mag, fs=fs, window='hann', nperseg=win_len, noverlap=hop_len)

    return derev_signal
```

Appendix 2 Images

Appendix 2.1 INDIVIDUAL SOURCE AUDIO FROM MASKS



0	0	1	1
1	1	0	1
0	1	0	1
0	0	1	0

Fig. 9: Individual Source Audio from Masks

Appendix 2.2 SEPARATING COMPONENTS FROM MSUDB18 SONGS

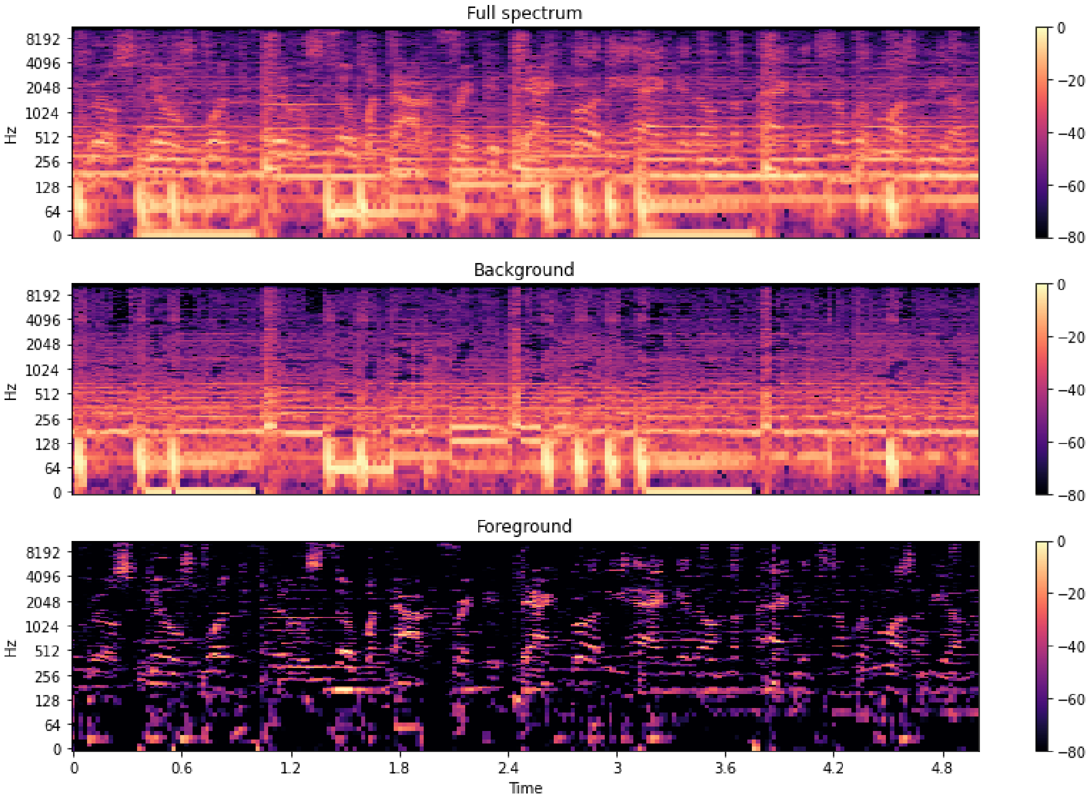


Fig. 10: Separating Components From Msudb18 Songs

REFERENCES

[1]. Wang, D., & Chen, J. (2018). Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(10), 1702-1726.

[2]. Gao, R., & Grauman, K. (2021, June). Visualvoice: Audio-visual speech separation with cross-modal consistency. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 15490-15500). IEEE.

[3]. Hu, X., Li, K., Zhang, W., Luo, Y., Lemercier, J. M., & Gerkmann, T. (2021). Speech separation using an asynchronous fully recurrent convolutional neural network. *Advances in Neural Information Processing Systems*, *34*, 22509-22522.

[4]. Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., & Zhong, J. (2021, June). Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 21-25). IEEE.

[5]. Zeghidour, N., & Grangier, D. (2021). Wavesplit: End-to-end speech separation by speaker clustering. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 2840-2849.

[6] Luo, Y., Chen, Z., & Yoshioka, T. (2020, May). Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 46-50). IEEE.

[7] Luo, Y., & Mesgarani, N. (2019). Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, *27*(8), 1256-1266.

[8] Chen, S., Wu, Y., Chen, Z., Wu, J., Li, J., Yoshioka, T., ... & Zhou, M. (2021, June). Continuous speech separation with conformer. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5749-5753). IEEE.

[9] Wu, J., Xu, Y., Zhang, S. X., Chen, L. W., Yu, M., Xie, L., & Yu, D. (2019, December). Time domain audio visual speech separation. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)* (pp. 667-673). IEEE.

[10] Wang, D. (2008). Time-frequency masking for speech separation and its potential for hearing aid design. *Trends in amplification*, *12*(4), 332-353.

[11] Smaragdis, P. (2006). Convolutive speech bases and their application to supervised speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(1), 1-12.

[12] Rahimi, A., Afouras, T., & Zisserman, A. (2022). Reading To Listen at the Cocktail Party: Multi-Modal Speech Separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10493-10502).

[13] Chung, S. W., Choe, S., Chung, J. S., & Kang, H. G. (2020). Facefilter: Audio-visual speech separation using still images. *arXiv preprint arXiv:2005.07074*.

[14] Tan, K., Xu, Y., Zhang, S. X., Yu, M., & Yu, D. (2020). Audio-visual speech separation and dereverberation with a two-stage multimodal network. *IEEE Journal of Selected Topics in Signal Processing*, *14*(3), 542-553.

[15] Montesinos, J. F., Kadandale, V. S., & Haro, G. (2022). VoViT: Low Latency Graph-based Audio-Visual Voice Separation Transformer. *arXiv preprint arXiv:2203.04099*.