Electrical and Computer Engineering
University of Thessaly (UTH)

ECE439 - Ειδικό Θέμα

Fall Semester — Educational year 2024-2025

Classical Music Source Seperation and Enhancement

Vasileios Stergioulis - AEM: 03166

Spyridon Petroglou - AEM: 03185

Contents

Acknowledgement

Σε αυτό το κομμάτι θα θέλαμε να ευχαριστήσουμε την κ. Παπαδημητρίου Αικατερίνη για την συνεισφορά της σε όλη την εργασία-project ακόμα και αν δεν ευόδωσε να την στείλουμε στον διαγωνισμό του Cadenza [1]. Η βοήθειά της ήταν καθοριστική, καθώς όχι μόνο μας καθοδήγησε και μας οργάνωσε, αλλά έτρεξε και τα νευρωνικά δίκτυα και μας γλύτωσε αρκετό πολύτιμο χρόνο.

Glossary of Notations

ARUNet Attentive MultiResUNets

BiLSTM Bidirectional Long Short-Term Memory

CASA Computational Auditory Scene Analysis

cMSS Classical Music Source Separation

CPP Cocktail Party Problem

DCT Discrete Cosine Transform

DTCT Short-Time Discrete Cosine Transformation

GNF Generalized Wiener Filter

MSS Music Source Separation

NMF Non-Negative Matrix Factorization

NN Neural Networks

sota state-of-the-art

STFT Short-Time Fourier Transform

TCN Temporal Convolutional Network

TD Time Domain

TF Time-Frequency

WUN Wave-U-Net

1 Introduction

Music source separation (MSS) and in general the source separation task, is a challenge dating back to the start of signal processing problems-tasks, especially the *cocktail party problem* (CPP) [2]. CPP, first proposed by Colin Cherry, is a psychoacoustic phenomenon that refers to the remarkable human ability to selectively attend to and recognize one source of auditory input in a noisy environment. Essentially is a task of who spoke when, and in our case with Classical Instrument is played in a mixture!

1.1 Source Separation

Returning to source separation, its target is to isolate individual signals-sound sources from a mixture of signals, with a wide area of applications such as speaker identification, speech enhancement and for our task *separating-estimating the original musical sources from a mixture*, which would allow us to remix, suppress or upmix the sources or instruments [3]. The techniques of source separation [4] can be summed up to three major categories:

I **Blind Isolation of Sources.** When there is no further information about the source signal.

II **Weakly Guided or Semi-Blind Isolation of Sources.** When there is some information about the signal, e.g. the type of instruments, the environment, the recording environment (studio, live music).

III **Strongly Guided Source Separation.** When there is specific information about the signal, e.g. source activity times.

IV **Informed Music Source Separation.** When detailed metadata is provided along with the waveform about the mixing process and the characteristics of the sources.

As it can easily deducted from the following chapters, our task is focused around the isolation of musical sources, for which we know only the number and type of instruments, thus we lie on the **Weakly Guided or Semi-Blind Isolation of Sources**.

1.2 Music Signals

Music signals have distinct characteristics that clearly differentiate them from other types of audio signals such as speech or environmental sounds. All music separation problems start with the definition of the desired musical source to be separated, often referred to as the target source. In principle, a musical source refers to a particular musical instrument, such as a saxophone or a guitar, that we wish to extract from the audio mixture. The instruments we are called to separate are the following:

- Bassoon
- Cello
- Clarinet

- Flute
- Oboe
- Saxophone
- Viola
- Violin

In our mixture is an ensemble of two to five instruments, whilst some instruments can have a second voice in the same mixture. Our task is by knowing the total number of instruments, to identify them, isolate them and finally enhance them, based on our listeners needs[1].

1.3 Problem Statement

Now that the basics of MSS and cMSS, are defined, it is time to state the problem [6], in a more mathematical way. Let $x(n) \in \Re^2$ be the stereo mixture (two-channel audio) in the time domain that is known to be composed by I sources in a way that:

$$x(n) = \sum_{ieI} s_i(n) \quad , I = [Bassoon, Cello, Clarinet, Flute, Oboe, Saxophone, Viola, Violin]$$

Our goal is of course to retrieve the stereo source estimate $\hat{s}_i(n)$, that resemble the ground truth sources $s_i(n)$.

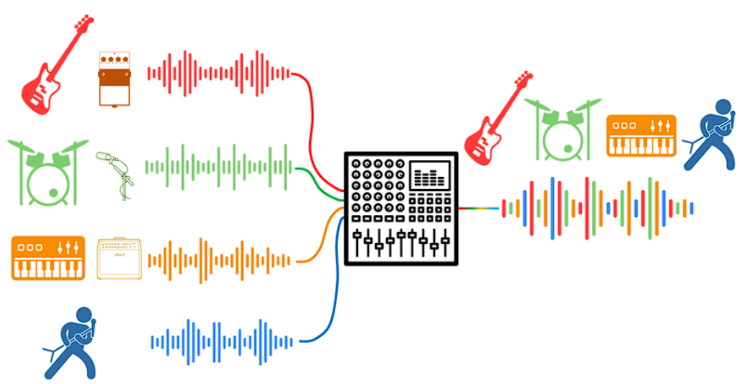

Figure 1: Mixing of music signals

[1]The project was to take part in the Cadenza Music Challenge 2024, Task 2, Separation of Classical Music and Enhancement for people with hearing loss [5]

2 Classic Methodologies

A common scheme-workflow for Musical Source Separation, is illustrated in **Figure 1**. First the input mixture signal is transformed to the time-frequency (TF) domain. The TF representation of the signal is then manipulated to obtain parameters which model the individual sources in the mixture. These are then used to create filters to yield time-frequency estimates of the sources. This is typically done in an iterative manner before the final estimated time-domain signals are recovered via an inverse TF transform. The most common *TF transform* that is used, is of course the Short-Time Fourier Transformation (STFT), but for our own purposes, we also found the Short-Time Discrete Cosine Transformation (TDCT) [7] to be an appealing alternative. In the *Source Modelling* domain, Most MSS methods focus solely on analyzing the magnitude spectrogram of the mix. The goal at this stage is to estimate either a model of the spectrogram of the target source, or a model of the location of the target source in the sound field. At the *Filtering* stage, the goal is to estimate the separated music source signals given the source models. This is typically done using a soft-masking approach, the most common form of which is the Generalized Wiener Filter (GWF) [8]. Essentially, each time-frequency point in the original mixture is weighted with the ratio of the source magnitude to the sum of the magnitudes of all sources. This can be understood as a multi-band equalizer of hundreds of bands, changed dynamically every few milliseconds to attenuate or let pass the required frequencies for the desired source. The final stage in our scheme is centered around *Inverse Transform*, that is to obtain the time domain source waveforms using the befitting inverse transform.

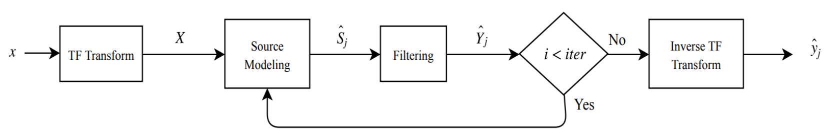

Figure 2: Typical MSS scheme

2.1 Transforms

2.1.1 Short Time Fourier Transform

The most common TF, that is used is the STFT. The Fourier Transform (FT) of a signal $f(t)$ is obtained by its correlation with the function $e^{-j\omega t}$:

$$F(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \tag{1}$$

The above transform is a representation in the frequency domain, providing information about the spectral content of the signal, but does not provide information about the change in spectral characteristics characteristics of the signal over time [9]. To achieve a joint analysis in time and frequency, the signal is multiplied by a non-zero window function $w(t)$ for a short period of time.

$$STFT[f(t)] = F(\tau, \omega) = \int_{-\infty}^{\infty} f(t)w(t-\tau)e^{-j\omega t}dt \tag{2}$$

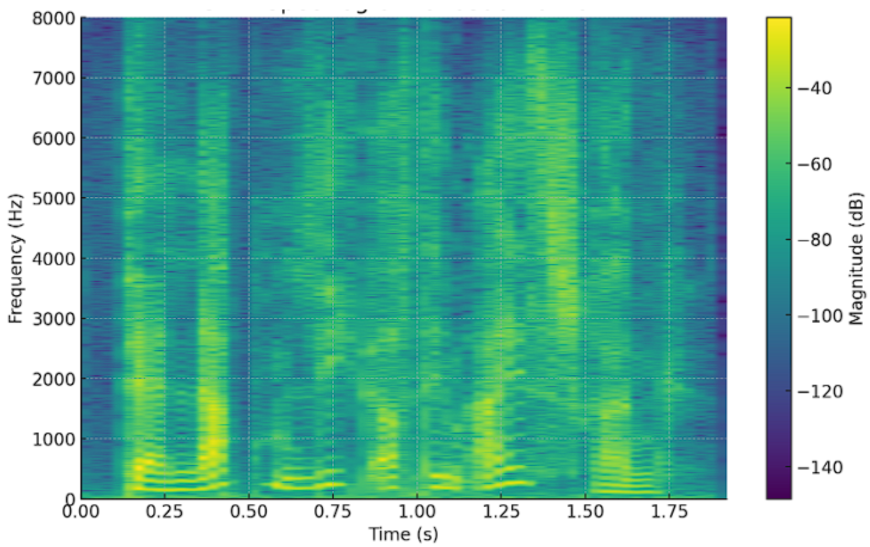

Figure 3: STFT spectrogram

By arranging the segments resulting from all the window shifts, we obtain a representation in time and frequency of the signal, containing information about the change in spectral content over time. This representation is the well-known STFT. Note that the window $w(t)$ we use can be either a Hamming, Hanning, etc, window, but for the consistency of our experiments, Hamming window is used.

An interesting fact is the product $w(t-\tau)e^{-j\omega t}$, known as Gabor Frame. If we choose gaussian as our window function, in order to obtain the maximum STFT resolution [10], then the above transform can be also named Gabor Transform.

2.1.2 Short Time Discrete Cosine Transform

A common problem while using STFT, is that the signal can be inverted back to the time domain but this process is complex, computationally expensive and the offered improvement is not always noteworthy. That's one of the reason DTCT is proposed.The main motivation behind DTCT [7], is that the transform is equally sparse and linear, but most importantly it is a real-valued transform. Thus, the transform values can be directly presented to our NN as input and the network can infer the real values of the transformed separated sources.

DTCT follows the same mechanism as the STFT, but uses the Discrete Cosine Transform, instead of the Fourier Transform. More specifically, the audio signal is segmented into short overlapping segments of equal duration. Each of these frames is windowed and the 1D-DCT (1D Discrete Cosine Transform) [9] is applied on each frame. If the signal is stereo, then we apply the transform to each channel separately. For our purposes, DCT type-II is used. Assuming again $f(t)$ as our signal, $f(n)$ as our discrete signal and N_1 as the discrete signal length:

$$F(k) = \sqrt{\frac{2}{N_1}}\beta(k)\Sigma_{n=0}^{N_1-1}f(n)cos(\frac{\pi k(2n+1)}{2N_1}), \ \ \forall k \in [0, N_1-1]$$

$$\beta(k) = \begin{cases} \frac{1}{\sqrt{2}}, k=0 \\ 1, k=1,\cdots,N_1-1 \end{cases}$$

The real-valued DCT "spectrogram" can be used in this form for training the network, without any further processing and without losing any primal information. To inverse the transformation, one must simply use DCT type-III:

$$\hat{f}(n) = \sqrt{\frac{2}{N_1}}\Sigma_{k=0}^{N_1-1}\beta(k)F(k)cos(\frac{\pi k(2n+1}{2N_1}), \ \ \forall n \in [0, N_1-1]$$

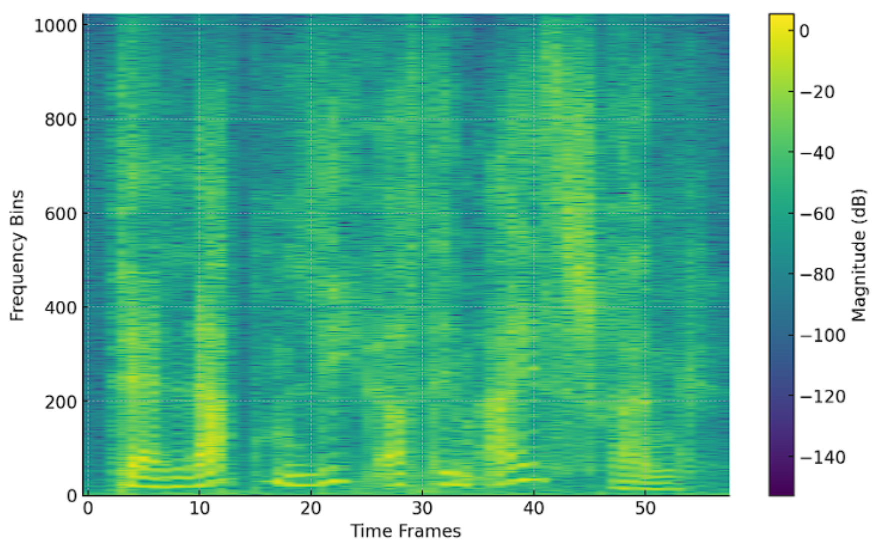

Figure 4: DTCT "spectrogram"

From the spectrograms of **Figures 2,3**, we can see that the DTCT produces a similar result to STFT; A difference that can easily spotted is that the DTCT spectrogram is more "rough" and more intense than STFT, which may or may not produce problems later on[2]

2.2 Methodologies

This chapter serves as an introduction to music source separation methods. Although in today's word they aren't used extensively (a bright exclusion would be that of Non-Negative Matrix Factorization (NMF) [11] , [12], [13]), they should serve as the "basics" - starter algorithms to help us evaluate and understand the task more.

[2]This is going to be a subtle foreshadowing, as the DTCT, didn't produce a correct- acceptable results as shown in later chapters

2.2.1 Computational Audiotory Scene Analysis (CASA)

Computational Auditory Scene Analysis Analysis (CASA) is a family of techniques that attempt to simulate the hearing human ability to isolate specific sounds; the desired audio-signal out of a mixture (coctail party problem [2]). CASA algorithms usually receive a time-frequency representation as input, through which derive the features that model the human auditory system. A clustering algorithm is then used, whose sole purpose is to separate by clustering the characteristics extracted in the previous stages. The clustering algorithms are either NNs, or they search for parameterized similarities of segmentation matrices.

The most known example of such an algorithm is [14], illustrated in **Figure 5**. The proposed algorithm by Brown and Cooke operates through two main processes: segmentation and grouping, both of which mirror human auditory perception. The first step, segmentation, involves decomposing an input sound mixture into elementary units in a spectro-temporal representation. To achieve this, the algorithm employs a gammatone filterbank [15], [16], which simulates cochlear filtering, breaking the signal into frequency bands. This is followed by hair-cell transduction, a nonlinear transformation that models how the auditory nerve processes sounds. The model then computes a running autocorrelation function for each frequency channel, producing a correlogram that captures periodicity and pitch-related information.

Once the auditory representation is formed, the next step is grouping, where these elements are organized into separate auditory streams corresponding to different sound sources. Grouping occurs in two stages: simultaneous and sequential. Simultaneous grouping operates across frequency, where the model clusters components that likely belong to the same source based on harmonicity and common onset cues. Meanwhile, sequential grouping tracks elements over time, ensuring continuity in pitch and onset/offset timing, which helps maintain a consistent stream for each source.

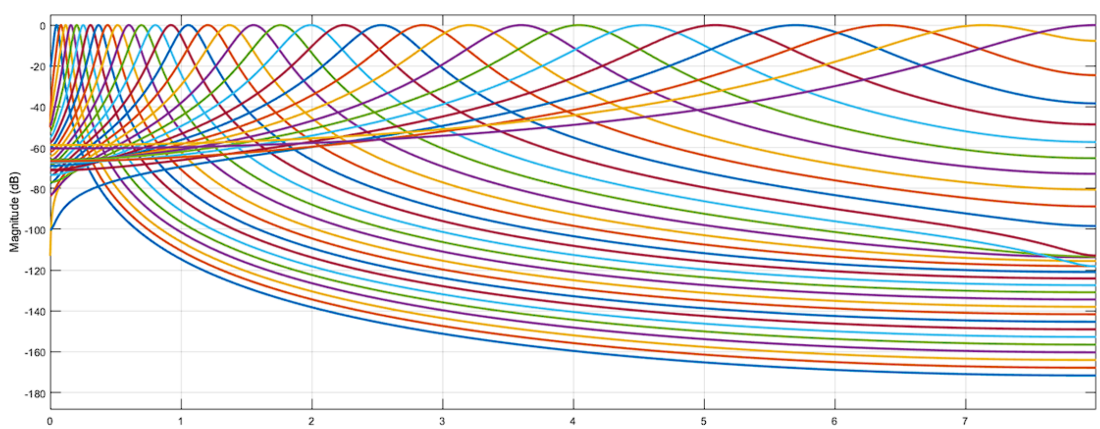

Figure 5: Gammatone Filterbank

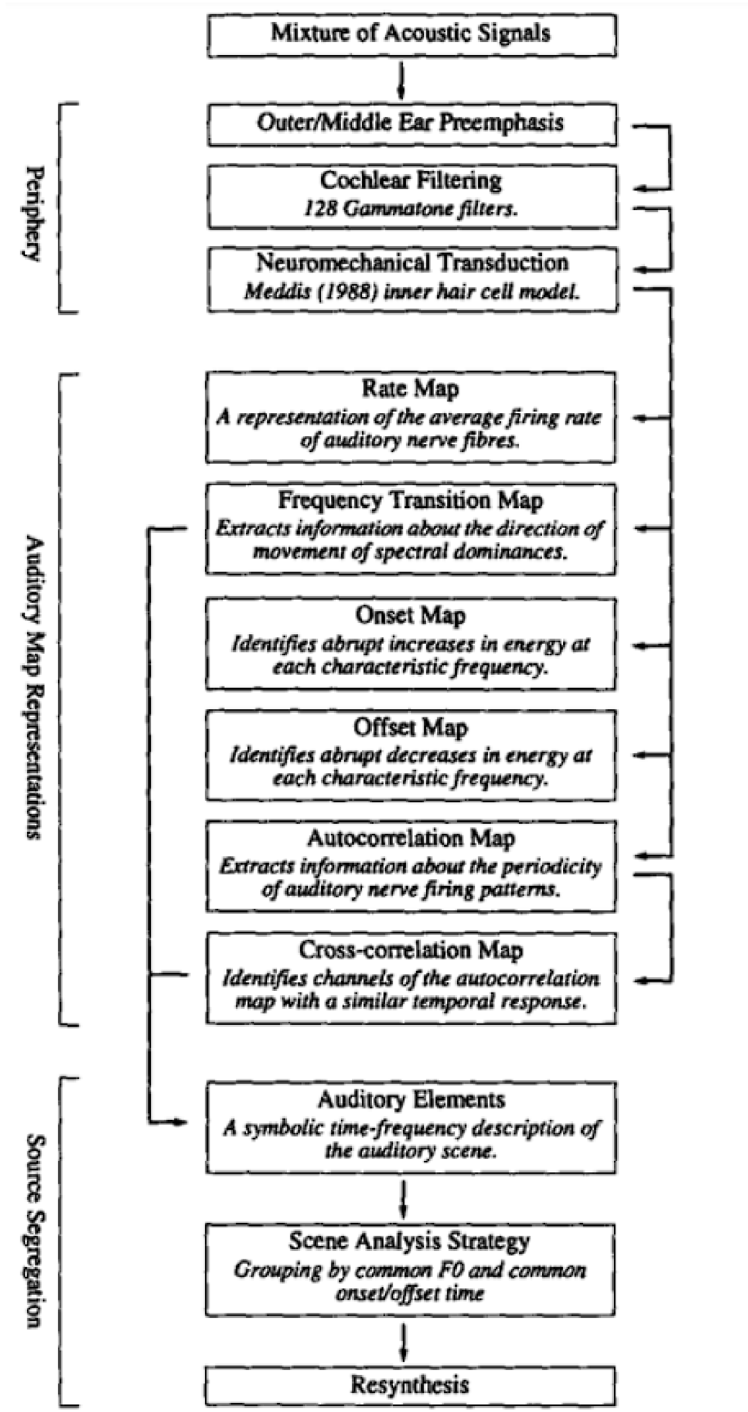

Figure 6: CASA System

2.2.2 Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF), attempts to factorize a given non-negative matrix into two nonnegative matrices [12]. In the MSS domain, NMF can be applied to the non-negative magnitude spectrogram of the mixture x (in section 1.3, we fedined our mixture as $x(n) \in \Re^2$, for stereo), which for our ease will be denoted as M [3]. The goal is to factorize M into a product $M = W \cdot H$, of a matrix of basis vectors W, which is a dictionary of spectral templates modeling spectral haracteristics of the sources, and a matrix of time activations H [11]. The factorization task is solved as an optimization problem where the divergence (or reconstruction error) between M and WH is minimized using common divergence measures D such Kullback-Leibler, or Itakura-Saito.

In [12], the separation task is executed in two ways: a *static NMF model* and a *dynamic NFM model*. The most classical example of a static model has been stated above, although probabilistic models hae been also suggested (such as Probabilistic Latent Component Analysis [17]); it is mostly a traditional method. For dynamic models, the dependencies between consecutive columns of M can be imposed either on the basis matrix W or on the activations H, a case of convolutive NMF. Here repeating patterns within data are represented with multidimensional bases which are not vectors anymore, but functions that can span an arbitrary number of dimensions (e.g., both frequency and time). Examples of dynamic models are Smooth NMF and Nonnegative state-space models

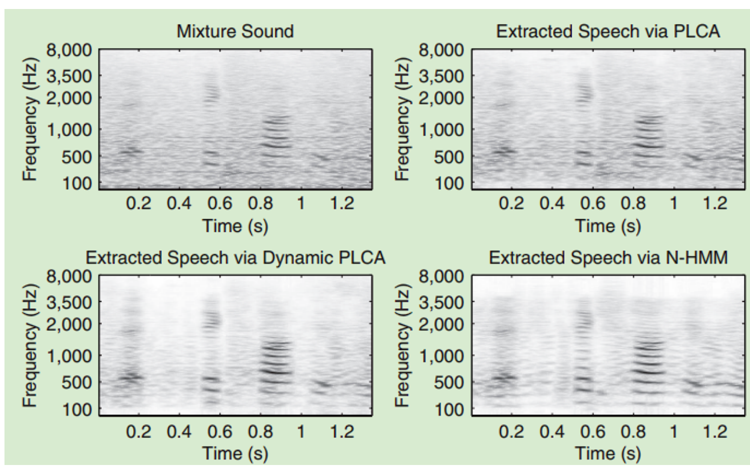

Figure 7: NMF Source Separation

3 Neural Network Models

MSS research has focused heavily on the use of model-based estimation that enforced desired properties on the source spectrograms. However, if the properties required by the models are not present, separation quality can rapidly degrade. The answer to this problem is of course the use of NN or Deep NN models. These models, are heavily inspired from the Computer Vision domain, with models such as VGG [18], ResNet [19], U-net [20], etc, used in the *Image Segmentation* task. The main idea, is that by using spectrograms as our "images", we can leverage the model properties and split them to the correct instruments.

3.1 Wave U-net

The first model used for the NN approach of the problem, is Wave-U-Net [21] or Multichannel U-net [22] (WUN), one of the first NN operating in the waveform domain. Wave-U-Net, is a computationally cheaper alternative from the other U-net family models for multi-instrument source separation, using a novel Energy Based Weighting strategy, based on the energy distribution in the ground truth sources. Altough the model was first proposed for speech de-noising and speech signal separation problems, it has notable results in the MSS domain.

As illustrated in **Figure 8**, the Wave-U-Net architecture consists of convolution layers, with an increasing number of higher-level features on coarser time scales using downsampling blocks. These features are combined with the earlier computed local, high-resolution features using upsampling blocks, yielding multi-scale features which are used for making predictions. The network has L levels in total, with each successive level operating at half the time resolution as the previous one. For K sources to be estimated, the model returns predictions in the interval $[-1, 1]$ (normalized output), one for each source audio sample.

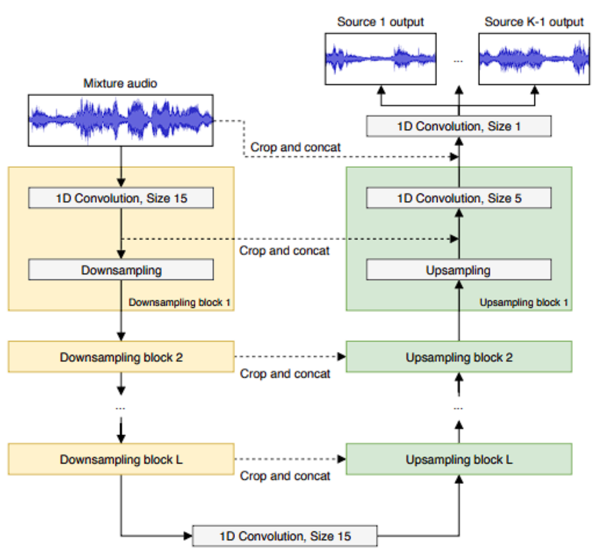

Figure 8: Stoller's proposed Wave-U-Net model

Again, the Multichannel U-net (or more correctly Multichannel Wave-U-net) generates multiple outputs, one per source in the mixture. Having multiple outputs gives rise to multiple task-specific loss terms and hence the following multi-task loss function:

$$\mathcal{L} = \Sigma_{i=1}^{K} w_i L_i \ ,$$

where L_i is the loss term corresponding to the i-th source, w_i is its corresponding weight, K is the number of sources and $\mathcal{L}$ is the overall scalar-valued loss. Note that the input here is the log-magnitude spectrogram of a mixture [23] because it achieved state-of-the-art performances. Our goal by using WUN, is to generate soft masks M_i, as our outputs; by applying the masks to a log-magnitude spectrogram we manage to "filter" the instruments of non-interest and segment our spectrogram.

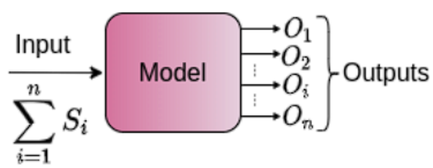

Figure 9: Multichannel Wave-U-Net model

3.2 ConvTasNet

Convolutional Time-Domain Audio Separation Network, or ConvTasNet [24] for short, was the baseline model for the competition. Although it was primarily made for Speech Separation, it was also used for MSS tasks. It operates fully in the Time Domain (TD) in order to avoid phase reconstruction issues that appear in spectogram-based methods. It uses an Encoder/Decoder architecture, where the encoder receives the mixture waveform and the decoder outputs the separated waveforms. The mixture waveform can be expressed as

$$x(t) = \sum_{i=1}^{C} s_i(t), \tag{3}$$

where $x(t)$ is the input waveform, $s_i(t)$ is the ideal waveform of the i-th target and C is the total number of sources. Our goal is to directly estimate $s_i(t)$, $i = 1, \ldots, C$, from $x(t)$. It was primarily designed to operate in mono signals (1D waveforms), but it is easily expandable to stereo signals.

There is an intermediate block between the encoder and decoder blocks, named separation block. Its purpose is to create multiplicative functions for each of the target sources. When the NN computes these masks, they will be multiplied element-wise with the waveform of the output of the encoder in order to estimate the target waveforms. A high-level diagram of the ConvTasNet architecture is illustrated in the figure below:

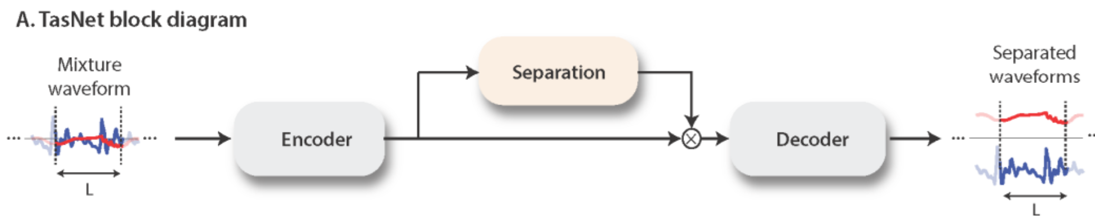

Figure 10: A High-Level ConvTasNet Architecture (Image from [24])

In the following figure we can see a more detailed diagram. Specifically, the encoder uses 1D convolutional layers (Depthwise Convolutions) to create a high-dimensional representation of the signal. This signal is then passed to the separation block which uses Temporal Convolutional Network (TCN) with dilated 1D convolutional layers to estimate the masks for each target. After the application of the masks in the high-dimensional signal, a transposed 1D convolution is used by the decoder to reconstruct the waveforms from the masked encoded representations. The final output consists of separated waveforms for each target instrument.

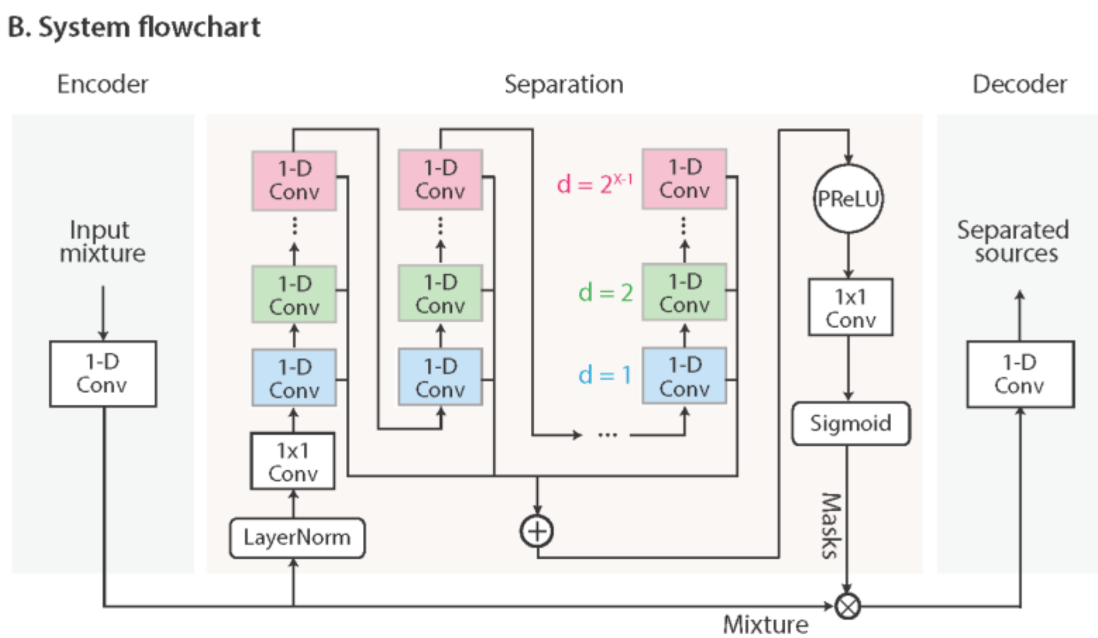

Figure 11: A More Detailed ConvTasNet Architecture (Image from [24])

In short, the described NN was a state-of-the-art model when first introduced, as it outperformed traditional spectrogram masking methods. Its innovative use of 1D convolutions in both the encoder and decoder, along with its ability to operate entirely in the waveform domain, made it a highly effective tool for MSS and, as we will see later, for cMSS as well.

3.3 Demucs Models and its Variations

The next model that is widely used for MSS tasks is named Demucs [25]. In comparison to the ConvTasNet, this NN was created to address the MSS problem. Since the release of the original

Demucs model in 2019, several newer versions have been introduced, improving its performance significantly. The first Demucs models were operating in the waveform domain, just like the ConvTasNet. However, after the third edition [26], Demucs became hybrid, which means that it was processing the input waveform in both time and frequency domains.

The novelty of this architecture is that it uses a Bidirectional Long Short-Term Memory (BiLSTM) to capture longer dependencies on the data, and U-Net-shaped convolutions [27] instead of TCN. It consists of six encoder layers and six, almost symmetric, decoder layers. The purpose of the encoder is to continuously increase the number of channels, till it reaches to the BiLSTM unit. On the other hand, the decoder performs the reverse operations, but in each layer, it concatenates the result with the output of the corresponding encoder layer.

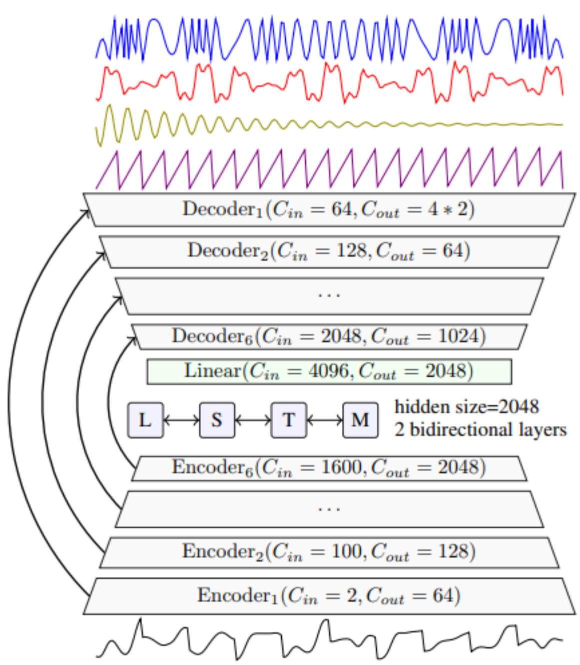

Figure 12: The Demucs v1 Architecture (Image from [25])

This model takes a stereo mixture $s = \sum_i s_i$ as input and outputs a stereo estimate $\hat{s}_i$ for each source. Thus, the encoder receives two signals in its input, whereas the decoder outputs twice as many channels as the target sources.

The Demucs architecture was extended from processing signals only in the waveform domain to processing in both the waveform and frequency domains [3]. It was the first model that was made to operate in both domains in order to maintain the benefits of both implementations. It took part in Sony's Music Demixing Challenge [28] and won it due to this unique implementation.

As you can observe in **Figure 11** above, the temporal and spectral channels are working individually until their dimensions align. Then, the outputs of the encoders are summed. There is

[3] This is the third version of Demucs. In the second version, some LSTM layers were removed to slightly improve the model's speed and efficiency.

also a summation between the output of the last temporal decoder and the Inverse STFT of the last spectral decoder. The BiLSTM unit was removed too, rendering the NN fully convolutional.

The last released model from the Demucs family [29] is illustrated in **Figure 12** below. It is the state-of-the-art model that is used to tackle the MSS problem right now. Inspired by the Transformer Neural Networks [30], self and cross attention modules were added in order to capture even longer distance dependencies than the BiLSTM of the first versions. Also, two out of six encoder and decoder layers were discarded, leading to four encoder and four decoder layers.

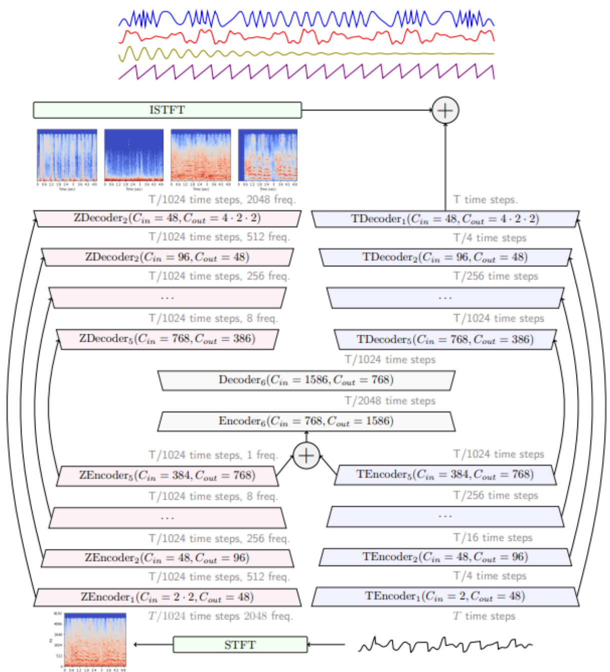

Figure 13: The Hybrid Demucs Architecture (Image from [26])

As you can observe in the figure above, the temporal and spectral channels are working individually until their dimensions align. Then, the outputs of the encoders are summed. There is also a summation between the output of the last temporal decoder and the Inverse STFT of the last spectral decoder. The BiLSTM unit was removed too, rendering the NN fully convolutional.

The last released model from the Demucs family [29] is illustrated in the figure below. It is the state-of-the-art model that is used to tackle the MSS problem right now. Inspired by the Transformer Neural Networks [30], self and cross attention modules were added in order to capture even longer distance dependencies than the BiLSTM of the first versions. Also, two out of six encoder and decoder layers were discarded, leading to four encoder and four decoder layers.

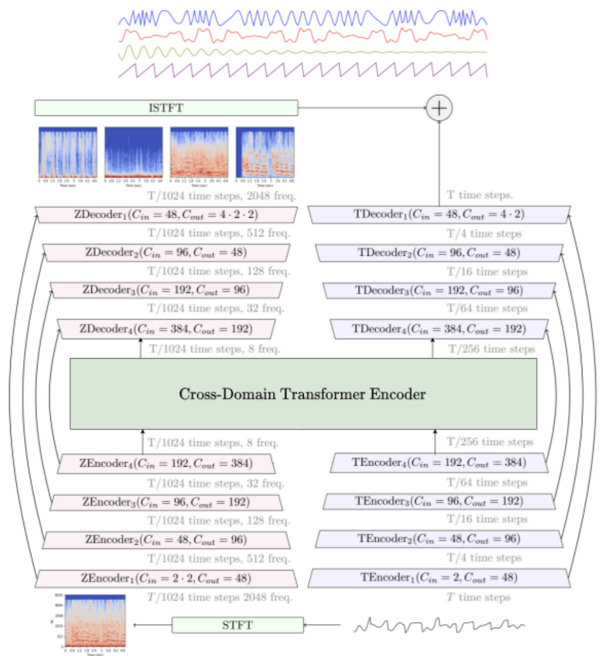

Figure 14: The Hybrid Transformer Demucs Architecture (Image from [29])

3.4 Attentive MultiResUNet

Attentive MultiResUNet [7] (ARUNet), follows the general U-Net architecture, containing an encoder and a decoder. Unlike the WUN, L separate ARUNets are trained, one for every instrument-component. The architecture is based on the original U-Net, but employs residual blocks that connect similar levels of the encoder and the decoder, making the network more robust and capable of analysing objects at different scales; an attention module is incorporated at the end of the residual skip connection path that connects the same level encoder and decoder layers. Its major advantage is the considerably decreased computational cost, compared to other sota source separation networks, while featuring performance that ranks behind only far more complicated networks. A general idea of the structure can be seen in **Figure 15**.

It should also be noted that in the NN, MultiRes blocks and Res paths are utilised. The MultiRes blocks consist of three different groups of 3×3 Convolutional blocks with a gradually increasing number of filters F. In every group, there is an increasing number of Convolutional blocks from 1 to 3 with each block containing a 3×3 Convolutional layer, followed by a Rectified Linear Unit (ReLU). The F metric [31], is used to create a connection between our model and the original U-Net. By gradually increasing the number of filters there is a compromise between heavy memory operations and the quality of feature extraction, therefore using larger size data inputs and acquire better audio quality.

The Res path, is a shortcut between the encoder and the decoder, similar to U-Net's skip connections. It is formed as a chain of Convolutional layers, which have residual connections. Using this path, the feature maps from the encoder are transferred to the decoder. There, they can be concatenated with the decoder's features, since they have the same size. The Res path assists the network in extracting improved features, as the information is more accurate, leading to better results. The incorporated self-attention mechanism at the end of the residual convolutional layers that connect each level of the encoder with the corresponding level of the decoder, has the ability to preserve the key features of the target source, while suppressing the features of the other components.

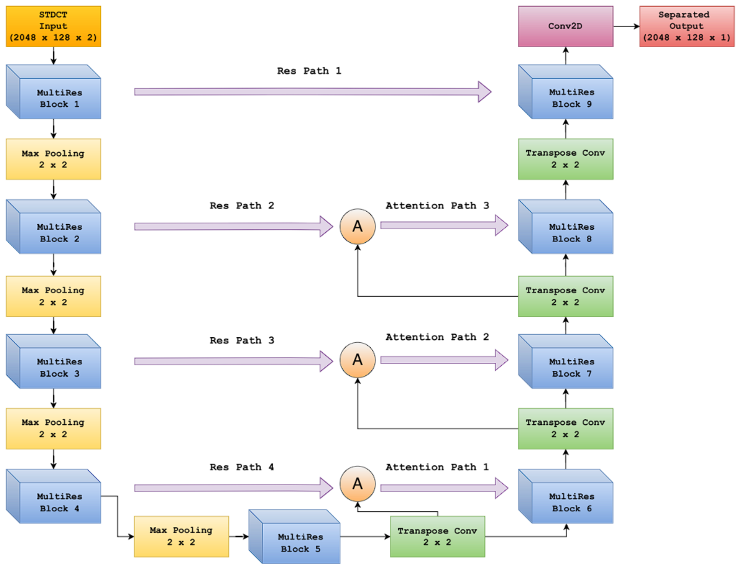

Figure 15: Attentive MultiResUNet

4 Proposed System

After a careful consideration of time and convenience, we deducted that the best results would be yielded by creating a **linear combination** of models, in a similar manner shown in [32]. Though some problems occurred during this phase, we concluded that the models to be used where: **ConvTasNet** which also served as our baseline, **Wave-U-Net** and **Attentive MultiResUnet** incorporating the DTCT. The last two NNs where tested in a subset of the original dataset (because time was not on our side), totaling six hundred (600) samples and produced the results as show in **Table 1**[4] .

Neural Network	№Testing Samples	HAAQI Accuracy
ConvTasNet	600	52.88 %
Wave-U-Net	600	50.55 %
Attentive MultiResUNet	600	22.45 %

Table 1: Individual NN performances.

Our main idea, is to create an Ensemble-Stacking of Neural Networks, in order to exploit the best results and to produce a system with the highest HAAQI accuracy. Again, due to time limitations, we choose to just produce a linear combination of the chosen NNs. Each NN produces a waveform (audio) for every instrument appearing in a mixture. Let $Conv(t)$, $Wav(t)$

[4]HAAQI is explained in a later chapter

and $Att(t)$ denote the outputs of ConvTasNet, WUN and ARUNet respectively. For every source $S \in [Bassoon, Cello, Clarinet, Flute, Oboe, Saxophone, Viola, Violin]$, the separated instrument signal $\hat{s}$:

$$\hat{s}_i(t) = w_{Conv} \cdot Cont_i(t) + w_{Wav} \cdot Wav_i(t) + w_{Att} \cdot Att_i(t), \quad i \in S,$$

where $w_{Conv} = 0.7$, $w_{Wav} = 0.2$ and $w_{Att} = 0.1$. Although this combination produced reletively good results for the selected subset, a better way of choosing weights, would be to create a *Meta-Model* that receives every instrument-model and produces a specialized combination for every instrument. That plan was unfortunately out of our reach, due to time constraints.

5 Dataset Description

To achieve our objective, we utilized a variety of datasets. The most well-known dataset for MSS tasks is musdb18 [33], along with its uncompressed version, musdb18-hq [34]. Although these datasets do not contain classical music tracks, we used them solely to adapt the models to our local hardware and verify their compatibility.

Fortunately, we were provided with datasets containing classical music tracks for the challenge. Due to time and hardware constraints, we used only two of them. The first, EnsembleSet [35], consists of ensembles featuring string, wind, and brass instruments. The second, CadenzaWoodwind [36], was created by the challenge organizers and includes ensembles of four different woodwind instruments: Oboe, Bassoon, Clarinet, and Flute.

Additionally, we had access to several datasets for fine-tuning and evaluation, including BACH10 [37] and the University of Rochester Multi-Modal Music Performance (URMP) dataset [38]. Detailed descriptions of all datasets, along with the provided metadata, can be found on the competition's Zenodo page [39].

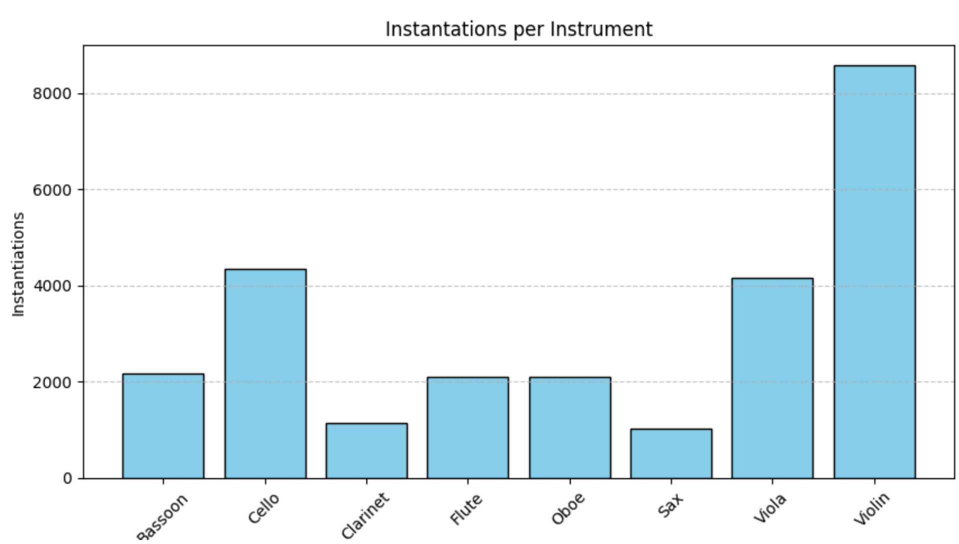

Figure 16: Variation of Instruments in the used Datasets

The facilitators provided us with a combination of the first two datasets and they split it in two subsets, the training set and the validation set. Along with that, they also gave us a metadata folder, with JSON files that describe the datasets. In short, the data are divided as follows: For each song in both train/valid sets, four different scenes are created. Each scene applies a different gain (in dB) in each target instrument. Also, in each scene, there are two different listeners, leading in eight different scene-listeners pairs. Thus, to obtain more robust results, there are created eight different instances for each single track.

6 Evaluation Metrics

In general, the most usual metrics used in MSS to evaluate a model are the Signal-to-Distortion Ratio (SDR), Signal-to-Inference Ratio (SIR) and Signal-to-Artifacts Ratio (SAR) [40]. The first one measures the overall distortion, including interference and artifacts and is given by the following formula

$$\mathrm{SDR} = 10 \cdot \log_{10}\left(\frac{||s_{\text{target}}||^2}{||e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}||^2}\right)$$

where
s_{target} is the true source signal,
e_{interf} is the error due to the interference of other sources,
e_{noise} is the error due to noise and
e_{artif} is the error due to artifacts introduced by the separation algorithm.

For the two remaining metrics, as the name suggests, SIR measures how well interference from other sources is suppressed, whereas SAR quantifies the amount of artifacts introduced by the separation process. They are given by the following formulas

$$\mathrm{SIR} = 10 \cdot \log_{10}\left(\frac{||s_{\text{target}}||^2}{||e_{\text{interf}}||^2}\right),$$

$$\mathrm{SAR} = 10 \cdot \log_{10}\left(\frac{||s_{\text{target}} + e_{\text{interf}}||^2}{||e_{\text{artif}}||^2}\right)$$

However, the facilitators of the challenge provided us with a more specialized evaluation metric, the Hearing-Aid Audio Quality Index (or HAAQI for short) [41]. Since this model is intended for people with hearing loss, the evaluation process should employ a more sophisticated method to penalize even minor distortions. HAAQI was the optimal choice because it considers perceptual factors by emulating human hearing perception. Unlike common metrics, it predicts perceived audio quality by comparing the original reference signal with the processed signal, focusing on both distortions and temporal characteristics.

The HAAQI score is computed separately for each ear. For each ear e $\in$ [left, right], we have:

$$Q_e = w_{\text{spec}} \cdot Q_{\text{spec},e} + w_{\text{temp}} \cdot Q_{\text{temp},e} + w_{\text{nonlin}} \cdot Q_{\text{nonlin},e}$$

where
$Q_{\text{spec},e}$ is the spectral similarity score for the selected ear,
$Q_{\text{temp},e}$ is the temporal envelope similarity score for the selected ear,

$Q_{\text{nonlin},e}$ is the core for non-linear distortion effects for the selected ear and $w_{\text{spec}}, w_{\text{temp}}, w_{\text{nonlin}}$ are the weighting factors that sum to 1, empirically determined to align with perceptual studies.

To compute the total HAAQI score, we just average the results that we got from each ear

$$Q_{\text{total}} = \frac{Q_{\text{left}} + Q_{\text{right}}}{2}$$

where Q_{left} and Q_{right} is the score for the left and the right ear correspondingly. The range of the score is between zero and one, where zero indicates a significant distortion (similar to having low SDR values), whereas one indicates a perfect perceptual quality, the signal under examination is the same with the reference signal.

7 Results

The results were the outcome of using the proposed system in section 4. The subset used V, consists of 1.011 samples in total; It also consists of two subsets $V_1 = 457$ & $V_2 = 554$, containing only string (V_1) and wind (V_2) instruments. The HAAQI scores of V_1, V_2 and V, are presented in **Table 2**.

Subset	Left HAAQI	Right HAAQI	Average HAAQI
V_1	0.5384	0.5115	0.5249
V_2	0.5163	0.5004	0.5084
V	0.5263	0.5054	0.5159

Table 2: System performance.

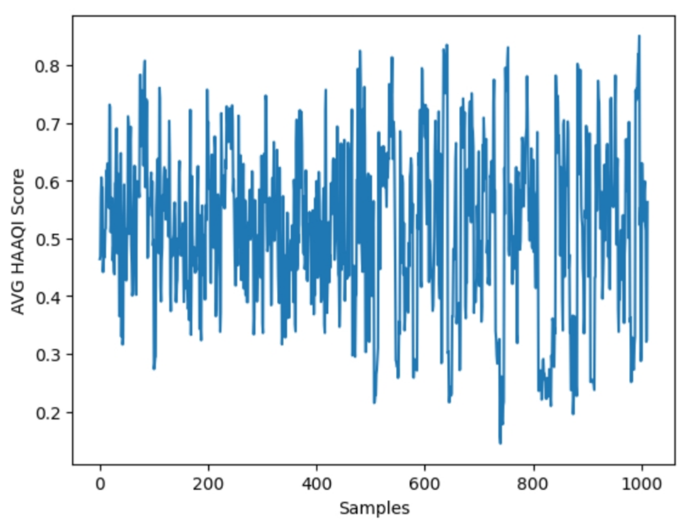

Figure 17: Average HAAQI Score across all samples

Although the idea was to surpass the ConvTasNet baseline, no such thing occured.

References

[1] G. R. Dabike, M. A. Akeroyd, S. Bannister, J. P. Barker, T. J. Cox, B. Fazenda, J. Firth, S. Graetzer, A. Greasley, R. R. Vos, and W. M. Whitmer, "The first cadenza challenges: using machine learning competitions to improve music for listeners with a hearing loss," 2024.

[2] S. Haykin and Z. Chen, "The cocktail party problem," *Neural Computation*, vol. 17, pp. 1875–1902, 09 2005.

[3] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F.-R. Stöter, "Musical source separation: An introduction," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 31–40, 2019.

[4] C. Jutten and P. Comon, "Chapter 1 - introduction," in *Handbook of Blind Source Separation* (P. Comon and C. Jutten, eds.), pp. 1–22, Oxford: Academic Press, 2010.

[5] "https://cadenzachallenge.org/docs/cadenza2/intro."

[6] M. Laurent, "Master thesis: Music source separation with neural networks," Master's thesis, Université de Liège, Liège, Belgique, 2023.

[7] T. Sgouros, A. Bousis, and N. Mitianoudis, "An efficient short-time discrete cosine transform and attentive multiresunet framework for music source separation," *IEEE Access*, vol. 10, pp. 119448–119459, 2022.

[8] N. Q. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *Trans. Audio, Speech and Lang. Proc.*, vol. 18, p. 1830–1840, Sept. 2010.

[9] A. Oppenheim and R. Schafer, *Digital Signal Processing*. MIT video course, Prentice-Hall, 1975.

[10] M. Szmajda, K. Górecki, and J. Mroczka, "Gabor transform, gabor-wigner transform and spwvd as a time-frequency analysis of power quality," in *Proceedings of 14th International Conference on Harmonics and Quality of Power - ICHQP 2010*, pp. 1–8, 2010.

[11] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *CoRR*, vol. abs/1709.05362, 2017.

[12] P. Smaragdis, C. Févotte, G. J. Mysore, N. Mohammadiha, and M. Hoffman, "Static and dynamic source separation using nonnegative factorizations: A unified view," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 66–75, 2014.

[13] P. Cabanas-Molero, A. J. Munoz-Montoro, J. Carabias-Orti, and P. Vera-Candeas, "Pretrained spatial priors on multichannel nmf for music source separation," 2023.

[14] G. J. Brown and M. Cooke, "Computational auditory scene analysis," *Computer Speech Language*, vol. 8, no. 4, pp. 297–336, 1994.

[15] M. Slaney, "An efficient implementation of the patterson-holdsworth auditory filter bank," Technical Report 35, Apple Computer, 1993.

[16] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand, "Complex sounds and auditory images."

[17] P. Smaragdis, B. Raj, and M. Shashanka, "A probabilistic latent variable model for acoustic modeling," in *Advances in Neural Information Processing Systems (NIPS)*, Dec. 2006.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[21] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," 2018.

[22] V. S. Kadandale, J. F. Montesinos, G. Haro, and E. Gómez, "Multi-channel u-net for music source separation," 2020.

[23] A. Jansson, E. J. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *International Society for Music Information Retrieval Conference*, 2017.

[24] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, p. 1256–1266, Aug. 2019.

[25] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[26] A. Défossez, "Hybrid spectrogram and waveform source separation," in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[28] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défossez, M. Kim, W. Choi, C.-Y. Yu, and K.-W. Cheuk, "Music demixing challenge 2021," *Frontiers in Signal Processing*, vol. 1, Jan. 2022.

[29] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *ICASSP 23*, 2023.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[31] N. Ibtehaz and M. S. Rahman, "Multiresunet : Rethinking the u-net architecture for multimodal biomedical image segmentation," *Neural Networks*, vol. 121, p. 74–87, Jan. 2020.

[32] M. Daly, “Remixing music for hearing aids using ensemble of fine-tuned source separators,” 2024.

[33] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “The MUSDB18 corpus for music separation,” Dec. 2017.

[34] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “Musdb18-hq - an uncompressed version of musdb18,” Aug. 2019.

[35] S. Sarkar, E. Benetos, and M. Sandler, “EnsembleSet: A New High Quality Synthesised Dataset for Chamber Ensemble Separation,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, (Bengaluru, India), pp. 856–863, December 2022.

[36] G. Roa Dabike, T. J. Cox, A. J. Miller, B. M. Fazenda, S. Graetzer, R. R. Vos, M. A. Akeroyd, J. Firth, W. M. Whitmer, S. Bannister, A. Greasley, and J. P. Barker, “The cadenza woodwind dataset: Synthesised quartets for music information retrieval and machine learning,” *Data in Brief*, vol. 57, p. 111199, 2024.

[37] Z. Duan and B. Pardo, “Soundprism: An online system for score-informed source separation of music audio,” *J. Sel. Topics Signal Processing*, vol. 5, pp. 1205–1215, 10 2011.

[38] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, “Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications,” *IEEE Transactions on Multimedia*, vol. 21, p. 522–535, Feb. 2019.

[39] T. J. Cox and G. Roa Dabike, “Cadenza challenge (cad2): databases for rebalancing classical music task,” Dec. 2024.

[40] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[41] J. M. Kates and K. H. Arehart, “The hearing-aid audio quality index (haaqi),” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 354–365, 2016.