Master Thesis

A Wavenet for Music Source Separation

Author: Francesc Lluís Salvadó

Advisors: Jordi Pons

Music Technology Group

Universitat Pompeu Fabra

Barcelona, August 2018

Abstract

Currently, most successful source separation techniques use magnitude spectrograms as input, and are therefore by default discarding part of the signal: the phase. In order to avoid discarding potentially useful information, we propose an end-to-end learning model based on Wavenet for music source separation. As a result, the model we propose directly operates over the waveform – enabling, in that way, to consider any information available in the raw audio signal. Provided that the original Wavenet model operates sequentialy (i.e., is not parallelisable and hence slow), in this work we make use of a discriminative non-causal adaptation of Wavenet capable to predict more than one sample at a time – thus permitting to overcome the undesirable time-complexity that the original Wavenet model has. Further, we investigate several data augmentation techniques and architectural changes to provide some insights on which are the most sensitive hyper-parameters for this family of Wavenet-like models. Our experimental results show that it is possible to approach the problem of music source separation in a end-to-end learning fashion, since our model performs on par with DeepConvSep – a state-of-the-art method based on processing magnitude spectrograms.

Acknowledgements

I would like to thank Jordi Pons for the invaluable guidance, support, patience, and sharing his passion for research provided during my stay at MTG. I specially appreciate the trust you have deposited in me since minute zero. It truly has been an honor to work with you.

I am grateful to my SMC fellows. The great environment has been a key factor in this fruitful and intense year. I would also like to thank all the academics and professionals in MTG with whom I have had the chance to discuss my work with during these months.

Last but not least, I want to thank immensely my family and friends for the support received.

Revision history and approval record

Revision	Date	Purpose
0	1/08/2018	Document creation
1	24/08/2018	Document revision
2	29/08/2018	Document approbation

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Francesc Lluís Salvadó	francesc.lluis@gmail.com
Jordi Pons	jordi.pons@upf.edu

Written by:		Reviewed and approved by:	
Date	19/08/2018	**Date**	29/08/2018
Name	Francesc Lluís Salvadó	**Name**	Jordi Pons
Position	Project Author	**Position**	Project Supervisor

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Since the digital revolution, there has been a shift in how people interact with music. Nowadays, music is more accessible than ever before and listeners have changed their preferences. Personalized ways to hear and learn about music are preferred over traditional ways of listening and discovering it [2]. Furthermore, similarly to painting or writing, musicians are now able to work directly with sound thanks to digital technologies. They don't need to know any symbolic notation (like how notes are represented in a score) to create music. To continuously satisfy music society needs, experts from music cognition, perception, engineering, musicology, and computer science have worked together to propose algorithmic and methodological solutions to music technology problems [3].

One topic of great interest within this research community is the sound source separation problem. It consists in recovering each individual source contribution from an observed mixture signal – and is motivated by the fact that sounds are generally composed of several individual sounds coming from different sources. Although audio source separation is a particularly difficult task [3], it underlies a wide range of applications such as speech denoising, content-based analysis and processing, audio restoration, or music remixing. In this work we address the problem of music source separation that, if solved, can be a truly empowering tool for those communities working in music remixing and music creation. For example, artists will have the opportunity to separate the singing voice, bass or drums from an existing mixture recording – enabling such artists to produce different versions of any original piece

by changing the pitch or rhythm of every separated track. Or, as another example, properly extracting the vocals from the mixture opens infinite opportunities within the Mashup style which, in a form of a song, consists on overlaying the vocal track of one song seamlessly over the instrumental track of another. Not to mention the commercial potential that an *universal* karaoke application has.

Since 2008, the Signal Separation Evaluation Campaign (SiSEC) [4] has become a reference (both in terms of datasets and participation) to compare the performance of music source separation systems. This year's SiSEC established a new record of participation (with over 30 different competing approaches), what confirms the increasing interest of the research community for this research area. Over these last years, the progress made by this community has been enormous – particularly after this first wave of deep learning based systems, that now define the state-of-the-art.

However, most successful deep learning algorithms participating in this year's SiSEC use magnitude spectrograms as input – and are therefore by default omitting part of the signal: the phase. Provided that via discarding potentially useful information (the phase) it exists the risk of finding a sub-optimal solution[1], in this work we aim to take advantage of the acoustic modelling capabilities of deep learning to investigate whether it is possible to approach the problem of music source separation in a end-to-end learning fashion. As a result, our investigation is centered on studying how to separate several musical sources (e.g., singing voice, bass or drums) directly from the raw music mixture – i.e., the waveform of the mixture is fed to the model without any pre-processing.

Finally, we want to remark that the idea of approaching the music source separation task directly in the waveform domain has not been actively explored throughout the years – possibly due to the elevate complexity of dealing with waveforms (that are unintuitive and high-dimensional). However, interestingly, recent works are starting to follow this research direction and are reporting promising results [4, 8, 9, 10, 11].

[1] Specially if the phase of the original mixture is used for reconstructing each of the sources [5, 6, 7].

Our work aims to keep adding knowledge on top of this (rather scarce) literature, in order to gain insights in how to approach the problem of music source separation in a end-to-end learning fashion.

1.2 Contributions

The main contribution of this thesis is to develop a competitive singing voice and multi-instrument source separation model that operates in a end-to-end learning fashion, that we call: Wavenet for Music Source Separation.

The specific contributions are:

- To review and analyze the main source separation techniques – with a particular focus on describing which techniques are capable to directly operate over waveforms.
- The adaptation of a Wavenet based state-of-the-art speech denoising model [8] for music source separation.
- An extensive evaluation showing how different data augmentation techniques, architectural changes, and hyper-parameter values affect the performance of the Wavenet for Music Source Separation.

1.3 Thesis outline

Chapter 1 motivates the problem we address throughout the thesis, and anticipates which are the main contributions of our work.

In chapter 2 we introduce the required background related to the source separation problem, and we review the state-of-the-art while we discuss the main approaches to

that task. Finally, the basics of the proposed approach are presented via discussing the merits of the model under study: Wavenet.

Chapter 3 presents our experimental results. The results of the proposed source separation model are outlined and discussed for the tasks of singing voice extraction and for multi-instrument extraction. Furthermore, we detail how some data augmentation techniques and architectural changes affect the performance of the model.

Finally, in Chapter 4, conclusions and a variety of future work are discussed.

Chapter 2

Background

2.1 Problem formulation

When two or more sounds exist at the same place and time, they interfere with each other resulting in a novel waveform signal where sounds are superposed (and, sometimes, masked). Source separation tackles the problem of recovering each individual source contribution from the observed mixture signal.

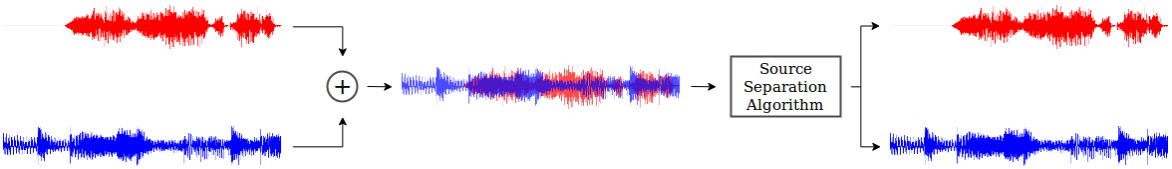

Figure 2.1: Monaural source separation diagram for two source signals

This problem inherently contains huge difficulties because there are more unknown variables than observed signals (what defines an ill-posed problem), and we estimate from a set of observations the causal factors that produced them (inverse problem).

Mathematically speaking, the problem was initially formulated as a linear mixing model where the mixture signal $x(t)$ (recorded by a microphone i) can be expressed as the weighted sum (via some gains $a_{i,j}$) of several source's $s_j(t)$:

$$x_i(t) = \sum_{j=1}^{J} a_{i,j} s_j(t) \tag{2.1}$$

As we will see in the following section, this formalism has marked the endeavour of the field – and many approaches are based on the previous formulation.

2.2 General approaches to source separation

Source Separation problem has been approached in several ways which include clustering, matrix decomposition, deep learning, or probabilistic methods.

Every method, depending on the amount of information used to achieve the separation, can be classified from *Blind Source Separation* (BSS) to *Informed Source Separation* (ISS) [12]. BSS techniques do not take into account any information about the sources nor the mixing process. But, on the other hand, ISS techniques assume that some aspects of the audio sources and the mixture process are known in advance. The progressive transition between BSS and ISS is determined by the amount of side information incorporated in the models. For example: a weakly guided ISS could be based on knowing how many sources are present in the mixture, or a strongly guided ISS system can rely on knowing the score of the musical piece to separate.

2.2.1 Computational auditory scene analysis (CASA)

Originally, in order to tackle the source separation problem, it has been essential to understand how humans separate the individual sounds in natural-world situations. For example, the Gestalt principles define how humans *group* sensory data – like audio [13]. These principles are based on the observation that humans naturally perceive objects as organized patterns and objects. Therefore, one can use these principles to define rules for dividing (via grouping) an audio stream into sources. The approaches that follow this idea are mainly based on grouping data that arrives at the same time – assuming that these are likely to be parts of the same sound stream.

There have been several attempts to built source separation systems based on these perceptual principles [14, 15] – a field known as Computational Auditory Scene Analysis (CASA). However, in music source separation there is a high correlation (both in time and frequency) between the sources [16], which is unfortunate for CASA models

since these sources become even less *groupable* when they share harmonics, onsets and have comparable timbres. For this reason, the vast majority of systems are not solely based on perceptual models but also use side information.

2.2.2 Matrix decomposition methods

Historically speaking, matrix decomposition methods (relying on the previously introduced formulation) have shown very promising results. For example, independent component analysis (ICA) [17], principal component analysis (PCA) [18], and non-negative matrix factorization (NMF) [19] methods have been widely used throughout the years. ICA exploits a statistical discriminant to differentiate the sources and decompose the input into bases. PCA uses an orthogonal transformation to factorize the data into bases that best explain the data variance. And NMF approximates the input (in most cases a time-frequency representation) as a linear product of non-negative basis and some gains. Although NMF has been able to deliver better results than ICA and PCA [19, 7], it is interesting to note that only ICA and PCA are appropriate to operate over waveform signals (which is the scope of our work) – since their basis and gains are not restricted to be non-negative (like waveforms, ranging from -1 to 1).

These methods manly rely on the formulation we introduced in equation 2.1, which describes the mixture as a weighted sum of basis functions. The goal is to find the vector basis that better approximates (linearly) the sound generated by each source. The way to do so is to estimate both the gains and the basis with an iterative algorithm that is based on maximizing the quality of the approximation via minimizing a cost function.

For the NMF case, the most successful matrix decomposition method [19, 7], it finds the optimized vector of basis that linearly approximates the input data $V \in R^{\geqslant 0, MxN}$ (generally being a time-frequency representation of the audio) with a set of

non-negative basis w_i (the columns of $W \in R^{\geqslant 0, MxK}$) and a set of non-negative gains h_i (the rows of $H \in R^{\geqslant 0, KxN}$):

$$V \approx \sum_{i=1}^{K} w_i h_i^T = WH$$

In order to illustrate previous formulation, see Figure 2.2 where we have M frequency bins, N time samples and K decomposed components.

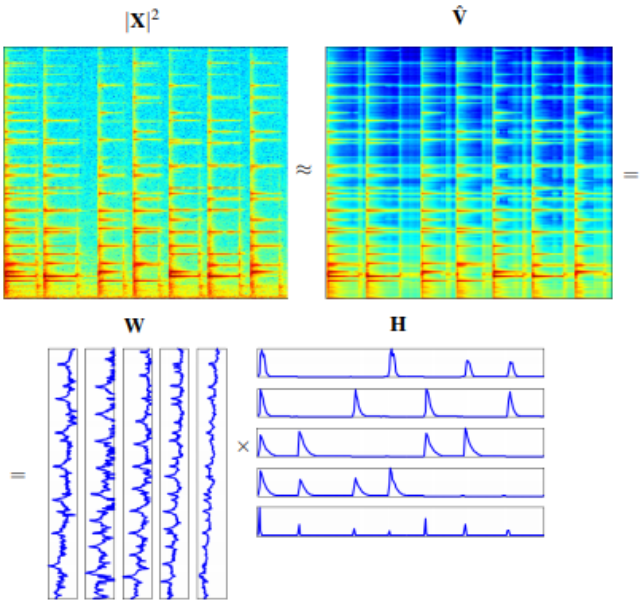

Figure 2.2: NMF applied to the spectrogram of a short piano sequence composed of four notes [1].

2.2.2.1 Signal representation for matrix decomposition methods

The music representation used as input for these matrix decomposition methodologies has a significant effect on the final performance. The most popular choices include the use of the raw waveform signal [20], spectrograms [5] or power spectrograms [7].

Waveform-based representations are straightforward to compute (basically, they do not require any pre-processing) and preserve all the information available in the raw signal. However, given the unpredictable behaviour of the phase in real-life sounds, it

is rare two find identical waveforms produced by the same sound source. As a result of this variability, a single basis cannot represent a sound source and therefore, an important amount of components are needed to obtain an accurate approximation [21]. Although several approaches have considered to directly approach the waveform with matrix decomposition methods (like ICA or PCA) [22, 23, 24, 25, 26], these techniques have never worked as well as spectrogram or power spectrogram based aproaches – possibly due to the aforementioned issues. Finally, it is worth mentioning that Abdallah and Plumbley [25, 26] found that the independent components they analysed, were similar to wavelet or short-time DFT basis.

Phase related problems disappear when the sound is represented in the time-frequency domain. Further, different realizations of the same sound are almost identical in the time-frequency domain – what allows to overcome the variability problem we found when operating with waveforms.

Many times, after applying the STFT to an audio fragment, the phase of the complex time-frequency representation is discarded – assuming that the magnitude (or power) spectrogram already carries meaningful information about the sound sources to separate. However, if the phases are not taked into consideration, is not equivalent to add several signals in the time-domain (waveforms) to add two signals represented in the spectral (or power spectral) domain – only in expectation[21]:

$$E\{|X(k)|^2\} = |Y_1(k)|^2 + |Y_2(k)|^2$$

Where $X(k) = DFT\{x(t)\}$. For this reason, most approaches make use of the power spectrograms as input to their models. However, one can observe that many works utilize magnitude spectrograms [7]. Although magnitude spectrograms work well in practice, it does not exist a similar theoretical justification. Most successful matrix decomposition methods utilize time-frequency respresentations as input – and are mostly based on NMF, that is very suitable for this signal because magnitude (or

power) spectrograms are non-negative signals.

Furthermore, when synthesizing the time domain signals after estimating the spectrograms of the sources, the phase has to be generated. Even though some approaches try to estimate the phase [27, 28], the main practice is to use the phase of the mixture for reconstruction purposes.

2.2.3 Deep learning methods

In the recent past, deep learning has emerged as a technique to solve complex problems that were previously unreachable. Source separation is one such problems where deep learning has had a very strong impact. This technique offers, mainly, two advantages when compared to matrix decomosition methods: i) provided that the underlying linear model defining matrix decomposition methods seems not expressive enough, deep learning models (that are highly expressive due to their capacity to model non-linearities) are an interesting opportunity to address the challenging task as music source separation; and ii) provided that during inference time deep learning models do not require any iterative algorithm to solve the task, deep learning methods are significantly faster than matrix decomposition ones – that are based on an iterative algorithm during inference.

Deep learning methods for source separation are generally formulated as a supervised regression problem (either via estimating a mask [29] or via directly predicting an output [8]) – where training data are used to optimize the parameters of the network given a cost function.

As reported in *The 2018 Signal Separation Evaluation Campaign* (SiSEC) paper [4], the community's methodology has shifted towards using deep learning. This technique being, by far, the most used during the last SiSEC edition – actually, during that edition, data-driven methods have clearly outperformed model-based approaches by a large margin for most targets and metrics [4].

2.2.3.1 Background

Neural Networks were first proposed by Warren McCullough and Walter Pitts in 1944 with a paper entitled "A Logical Calculus of Ideas Immanent in Nervous Activity" [30]. It was an attempt to find a mathematical representations of information processing in biological systems. Since then, and after a lot of research, neural networks have become the technique utilized by the best-performing artificial-intelligence systems using an approach named deep learning.

A neural network can have from a few dozen to millions artificial neurons arranged in a series of layers classified as input, hidden, and output. The input layer's function is to receive information from the outside and to process it. The hidden layers main job is to transform inputs into something that output layer can use. Finally, the output layer contains signals that represent how the network responds to the information it has learned.

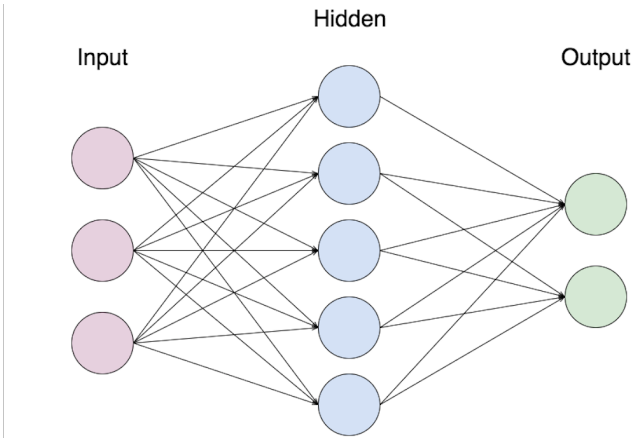

Figure 2.3: Example of a neural network. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.

The artificial neuron, which is the basis of a neural network, is inspired on a biological neuron. It contains input variables (x_i) that represents external stimulus or outputs from other neurons. After adding a bias term (b), the mentioned inputs are multiplied by weights (w_i) which are understood to perform as synapses. Then, ($x_i w_i$) are passed through an aggregation function and a nonlinear activation function ($h()$)

which is presumed to be the cellular body. Finally, it yield an output variable (y) assumed as the axon.

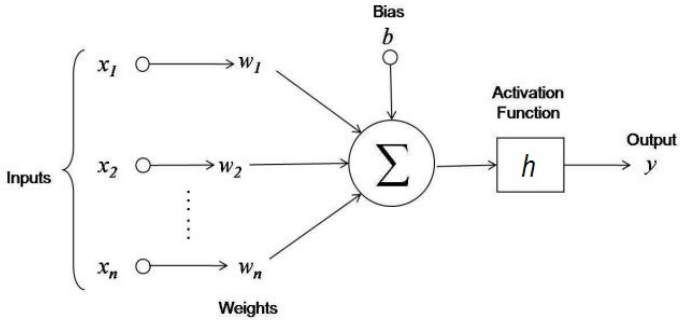

Figure 2.4: Artificial neuron model

The mathematical expression of the model takes the following form:

$$y = h(\sum_{i=0}^{n} w_i x_i) \tag{2.2}$$

Note that in equation 2.2 the bias parameter (b) is absorbed into the set of weight parameters by defining an additional input variable x_0 whose value is clamped at $x_0 = 1$.

To make a neural network useful for a specific task, it has to be trained which means that weights from layers are transformed until the network's outputs are close enough to the desired outputs. Initially, all the weights are set to random values. Then, the input layer is fed with training data and it passes through all layers. After being multiplied and added in complex ways it arrives transformed at the output layer and the difference between the target outputs and the actual outputs is calculated. During the learning process, this difference is back-propagated to the previous layer and the weights are normally adjusted using the delta rule. It finishes when the initial layer is reached.

$$\Delta w_{ij} = \alpha(t_j - y_j)h'(z_j)x_i \tag{2.3}$$

Equation 2.3 shows the delta rule for weight w_{ji} of a neuron j with activation

function $h(x)$. Where α is the learning rate, $h(x)$ is the neuron's activation function, t_j is the target output, $z_j = \sum x_i w_{ji}$ is the weighted sum of the neuron's inputs, $y_j = h(z_j)$ the actual output, and x_i is the i^{th} input.

2.2.3.2 Signal representation for deep learning methods

Nowadays, the main practice is to use time-frequency representations, like magnitude spectrograms, as input to the model. As previously explained, most people tend to use magnitude spectrograms because it allows to reduce the variance and high-dimensionality present in raw audio signals.

The 2D nature of time-frequency representations makes easy to derive inspiration from the deep learning architectures coming from the computer vision field – which are very popular among deep learning practitioners. For example, Takahashi et al. [31] proposes using the DenseNet architecture, which had shown excellent result on image classification, for music source separation. Similarly, Jansson et al. [32] adapted the U-Net architecture, which was initially developed for medical imaging tasks, for singing voice separation.

Furthermore, music signals do not necessarily need to be mono. Instead, they can have two channels (stereo) or even more (like 7.1 surround sound). In line with that, for example, Nugraha et al. [33] proposed using a deep neural network capable to process time-frequency representations combined with spatial covariance matrices (that encode the spatial characteristics of the sources).

Chandna et al. [5] proposed DeepConvSep[1], an open source model based on processing spectrograms that has been the state-of-the-art during these recent years. Their model is depicted in Figure 2.5 and can be devided into two parts: a convolutional encoding stage, and its inverse operation, the deconvolutional decoding stage. By using vertical and horizontal convolutions, they estimate time-frequency soft masks

[1] https://github.com/MTG/DeepConvSep

that are after used for separating the sources. In chapter 3, we set the DeepConvSep model as baseline for comparing our results against a state-of-the-art model processing spectrograms.

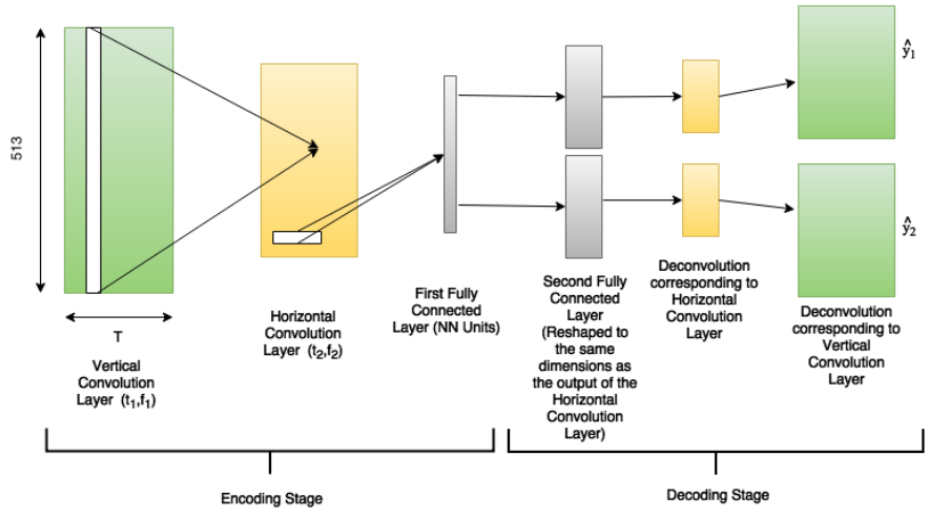

Figure 2.5: DeepConvSep Network Architecture

However, using magnitude spectograms as front-end comes with the drawbacks we explained in 2.2.2.1. In addition, there are reasons to think that instead of processing spectrograms it might be interesting to approach the task of source sepatation directly in the waveform domain. This would enable to, instead of utilizing generic feature representations like the STFT, learn a specific feature representation for a given task and data distribution. Or, as another example, it could permit to overcome the problem of not having a properly defined phase during synthesis (remember that in many cases the phase of the mixture is used to synthesize each source from the estimated spectrograms). Interestingly enough, the literature shows that deep learning models directly working with raw audio waveforms in a end-to-end fashion can achieve satisfactory results for generative tasks [34, 35], discriminate tasks [8], and classification tasks [36]. Moreover, recent results also show that it is possible to approch the problem of source separation directly in the waveform domain. Grais et al. [20] uses multi-resolution convolutional auto-encoders to determine appropriate multiresolution features for multi-channel singing-voice separation. Wave-U-Net [37], the only raw waveform system submitted to SiSEC 2018, uses an adaptation for the U-

Net architecture to the one-dimensional waveform signal to perform end-to-end audio source separation (using additional training data than the one provided by the SiSEC community). Venkataramani et al. [11] propose to learn optimal, real-valued basis functions directly from the raw waveform for the task of source separation. And Luo et al. [38] propose the TasNet, an end-to-end model for speech separation that utilizes an encoder-decoder framework to perform the separation on nonnegative encoder outputs.

As seen, the research community has a great for tackling the problem of source separation in a end-to-end learning fashion – what would help overcoming some historical challenges the field had to face. In this work, we propose using a Wavenet-like architecture for music source separation.

2.3 Proposed approach

Throughout our work, we investigate the use of an end-to-end deep learning model based on Wavenet [39] for the task of music source separation. It works in time domain (directly over the waveform) for the reasons we exposed above, and it uses a non-causal discriminative adaptation of Wavenet that (by learning in a supervised fashion via minimizing a regression loss) is able to overcome the original time-complexity of Wavenet – that is slow due to its causal generation of sounds. The proposed model for music source separation preserves Wavenet's acoustic capabilities, while being capable to run in real time [8].

The following lines introduce the original Wavenet (section 2.3.1), and the non-causal adaptation of Wavenet that we use throughout our experiments (section 2.3.1), that we call a Wavenet for music source separation.

2.3.1 Wavenet

Wavenet is a deep neural network capable of generating raw audio waveforms – which was able to outperform the best existing text-to-speech systems [39]. This model is able to produce individual audio samples at 16kH in 8 bit resolution and 24kHz in 16 bit resolution in its latest version [35]. Some of the Wavenet's main characteristics are explained below:

- **Sigmoidal gates for Tanh units**: Each convolutional block (Figure 2.6.a) of the model contains a gate that controls the contribution of each activation:

$$z = \tanh(W_{f,k} * x) \odot \sigma(W_{g,k} * x)$$

  where $\odot$ and $*$ operators denote element-wise and convolution multiplication, respectively; k is the layer index, f denote filter, g stand for gate, and W is a convolution filter.

- **Dilated convolutions**: convolutional blocks make use of dilated convolutions that allow the model to have a better global view of the input by exponentially increasing dilation factors – what increases the receptive field. Each dilated convolution is contained in a residual layer, controlled by a sigmoidal gate with an additional 1x1 convolution and a residual connection – see Figure 2.6.

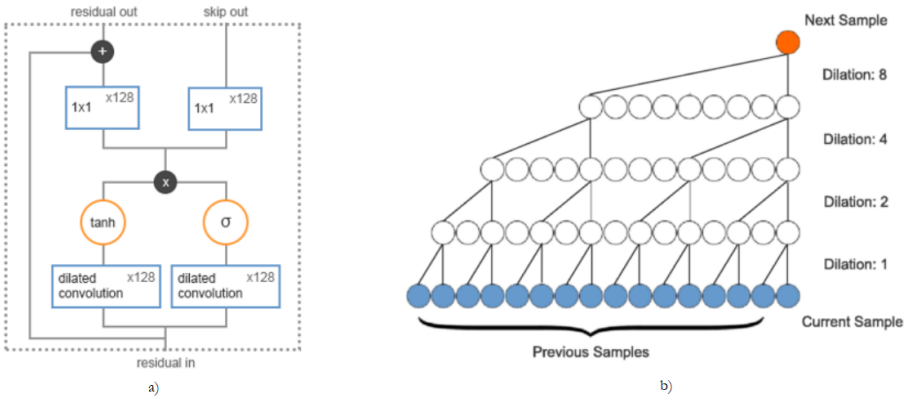

Figure 2.6: a) Residual layer. b) Dilated convolutions

- **Skip Connections**: The actual output is fed from skip connections from all layers. This allows the final prediction to contain different hierarchical level of extracted features coming from each layer.

- **Output of distributions**: Wavenet does not make real-valued predictions for each time step. Instead, it outputs a probability distribution which is sampled to produce the value of the next sample.

- **Computational burden**: Original version of Wavenet needs to feed the network with the previous generated samples in order to create the new one. This is slow during inference, and that is the main reason why we do not adopt the original Wavenet for our approach to music source separation.

2.3.2 A Wavenet for music source separation

As in Rethage et al. [8], our work makes use of an adaptation of the Wavenet model that aims to transform the original causal Wavenet model (that is generative but slow), into a non-causal model (that is discrminiative but parallelizable). The specificities of the adopted model are described below:

- **Non-causality**: Some samples from the future are used by the model to predict the present one. This results in having information about imminent sound events which are likely to enhance the current sample prediction. Also, in the source separation field, some milliseconds of latency of the model is affordable in real time applications.

- **Real-valued predictions**: The discrete softmax output used by Wavenet results in output distributions with high variance. For the source separation problem, real-valued predictions seem to be more suitable.

- **Target field prediction** Removing the autoregressive causal nature of the original Wavenet allows to predict more than one sample at a time. Due to this

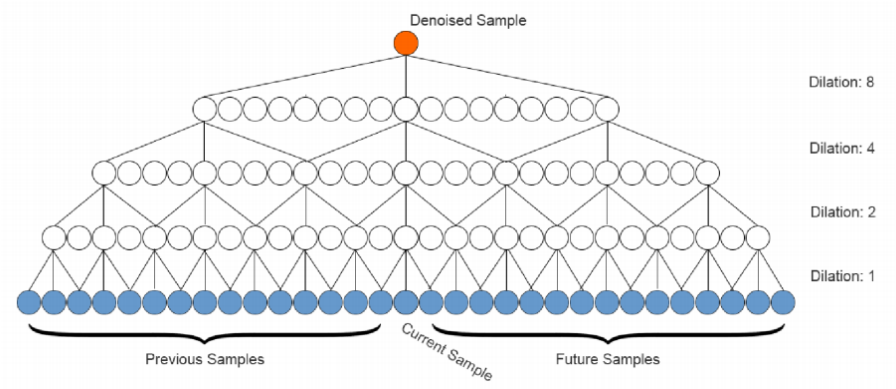

Figure 2.7: Visualization of a stack of non-causal convolutional layers

parallelization, it is possible to overcome Wavenet's time-complexity and memory constrains. In order to preserve that each target field has its corresponding receptive field, the length of the fragment that feeds the model must be equal to: $rf_1 + (tf - 1)$. Where rf_1 corresponds to the receptive field required to separate a single sample and tf is the length of the target field prediction.

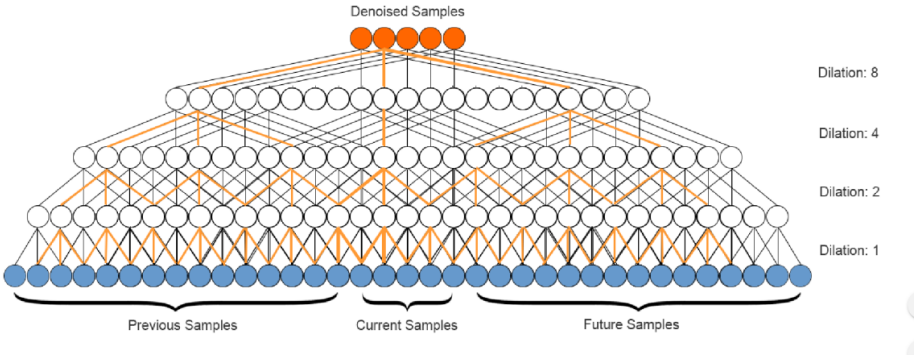

Figure 2.8: Target field prediction

Chapter 3

Experiments

This work develops a study in the deep learning field in how to adapt Wavenet for the task of music source separation. By discussing the obtained results for both singing voice and multi-instrument source separation tasks, this work investigates the possibilities of end-to-end learning for the task of music source separation. In short, the following experiments were considered:

- **Experiment 1**: evaluate a Wavenet for monaural **singing voice separation**.
- **Experiment 2**: evaluate a Wavenet for monaural **music source separation** – basically, we reformulate te task for multi-instrument separation (singing voice, bass, and drums).

The model's performance has been assessed using the MUSDB18 dataset considering *BSS Eval* evaluation metrics [40] – both explained in the following sections. Further, in order we compare our model with previous work, our results are compared against *DeepConvSep*: a model proposed by Chandna et al. [5] that we introduced in section 2.2.3.2.

3.0.1 MUSDB18 Dataset

The SiSEC 2018 Challenge dataset MUSDB18 [4] is used for this work. MUSDB18 is a dataset of 150 full-length music tracks with about 10 hours of duration containing different styles – along with their isolated drums, bass, vocals and *others* tracks. All signals are stereophonic and encoded at 44.1kHz. However, for the purposes of this work only one channel is considered for each source and it is subsampled to 16kHz.

During the singing voice separation experiment, training data is arranged as follows: drums, bass, and *others* are merged into a single audio stream to constitute the accompaniment signal. We set our training data into two parts: the train set (100 songs) and the validation set (which is conformed by 25 songs from the test set). No preprocessing to the audio, such as μ-law quantization [39], is used – allowing the pipeline to be end-to-end in the strictest sense.

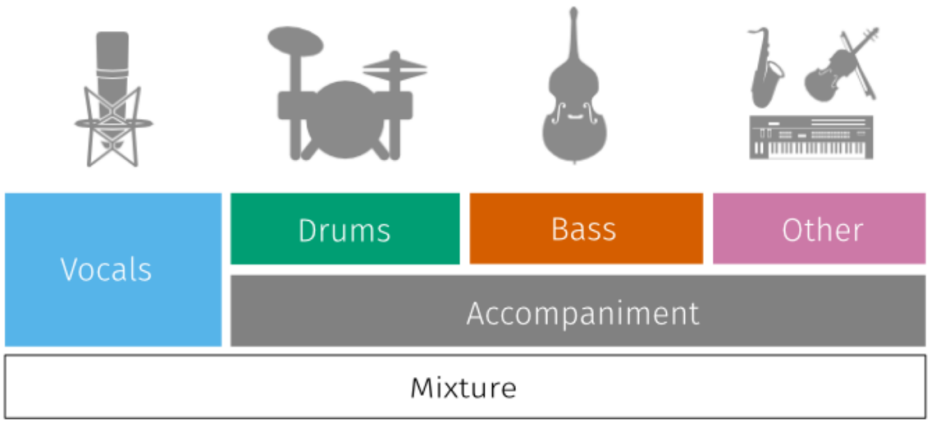

Figure 3.1: Musdb18 dataset multitrack format

3.0.2 Evaluation

Evaluating the results of a music source separation model is a difficult task. Mainly, due to the subjective hearing of humans. However, several objective metrics were developed to facilitate evaluating Blind Audio Source Separation (BASS) algorithms [41]. To this end, the estimated sources $\hat{s_j}$ (with $j = 1...J$) are decomposed as follows:

$$\hat{s_j} = s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{3.1}$$

Where $s_{target} = f(s_j)$ is a version of the original source s_j modified by an allowed distortion f. e_{interf} stands for the interference coming from unwanted sources found in the original mixture, e_{noise} denote the sensor noise, and e_{artif} refers to the burbling artifacts (musical noise) which are self-generated by the separation algorithm. Then, the objective metrics are computed as energy ratios in decibels (dB) – and higher

values are considered to be better:

Source to Distortion Ratio (SDR) is defined as:

$$SDR := 10 \log_{10} \frac{\|s_{target}\|^2}{\|e_{interf} + e_{noise} + e_{artif}\|^2} \tag{3.2}$$

Source to Interference Ratio (SIR) is defined as:

$$SIR := 10 \log_{10} \frac{\|s_{target}\|^2}{\|e_{interf}\|^2} \tag{3.3}$$

Source to Artifacts Ratio (SAR) is defined as:

$$SAR := 10 \log_{10} \frac{\|s_{target} + e_{interf} + e_{noise}\|^2}{\|e_{artif}\|^2} \tag{3.4}$$

Depending on the final application, these metrics can be useful in order to compare several source separation algorithms. For example, SIR values are related to the leakage coming from other sources, and SAR values give information about how the signal is deteriorated only due to the separation algorithm.

It is worth mentioning that perceptual evaluation is always preferred over objective evaluation – and, for example, can be carried out by asking to subjects questions like: "rate the sound quality of the examples below relative to the reference above" or "rate how well the instruments are isolated in the examples below relative to the full mixture above". The main drawback of perceptual experiments is that it is expensive to run them. Although throughout this manuscript we do not report the results of any perceptual experiment, we leave it for future work and provide several audio examples together with the thesis – so that any interested reader can perceptually assess the performance of our model.

3.0.3 Baseline setup

Our baseline model contains 30 residual layers, and is based on the model proposed by Rethage et al. [8]. The dilation factor in each layer ranges from 1 to 512 – in steps of powers of 2. This pattern is repeated 3 times (3 stacks). Preceding the first dilated convolution, the 1-channel input is linearly projected to 128 channels by a 3x1 convolution to satisfy the number of filters in each residual layer. The skip connections are 1x1 convolutions also featuring 128 filters – a RELU is applied after summing all skip connections. The final two 3x1 convolutional layers are not dilated, contain 2048 and 256 filters respectively, and are separated by a RELU. The output layer linearly projects the feature map into a single-channel temporal signal by using a 1x1 filter. This parameterization results in a receptive field of 6,139 samples ($\approx$ 384ms) and a target field of 1601 samples ($\approx$ 100ms).

3.0.4 Training procedure

During training, audio fragments are sampled randomly to fit the input length required by the model. As loss, it is used the mean absolute error (MAE) which is computed sample-wise and averaged for each source output samples. ADAM optimizer is adopted with a learning rate of 0.001 which is reduced to a factor of ten after five epochs with no improvement. We set the batch size to 10, and the model is trained during 250 epochs[1] and best validation loss weights are selected.

[1]Provided that we randomly sample the training data, is hard to define an epoch. In our case, we define an epoch to be 100 parameter updates.

3.1 Experiment 1. Singing voice separation.

The baseline setup of the Wavenet for singing voice separation shows promising results – see Table 3.1, where the results are compared with DeepConvSep (a state-of-the-art model based on processing magnitde spectrograms [5]).

		Wavenet			**DeepConvSep**		
		SDR	**SIR**	**SAR**	**SDR**	**SIR**	**SAR**
Vocals	Med	1.24	7.90	3.60	**1.95**	4.39	7.74
	Mean	0.34	6.98	3.09	1.37	3.88	7.35
Accompaniment	Med	8.36	10.50	13.60	**10.37**	14.95	13.25
	Mean	8.99	11.05	14.36	10.91	15.46	13.30

Table 3.1: Singing Voice Separation models comparison

After informal listening, we observe that its performance is really selective only generating singing voice when it is prominent in the mixture signal. When it is not, the model produces silence. However, the model struggles when generating singing voice in a continuous manner thus introducing artifacts during the separation process.

In the following sections, we study: i) several data augmentation strategies, ii) which is the best architectural setup, and iii) how different cost functions, losses, affect the performance of the model – in order to investigate if it is possible to improve the source separation capacity of the baseline model.

3.1.1 Data augmentation: circular shifting + forcing singing voice

Deep learning models are data-driven – which means that the data presented to the model can dramatically affect the final performance of it. For this reason, it is a common to practice to augment the data to cover a wider range of examples when training the network – with the idea of increasing the generalization capability of the model.

Circular shifting – Miron et al. [42] and Huang et al. [29] reported better results when circularly shifting the singing voice signals and mix them with the background music as a form of data augmentation for source separation.

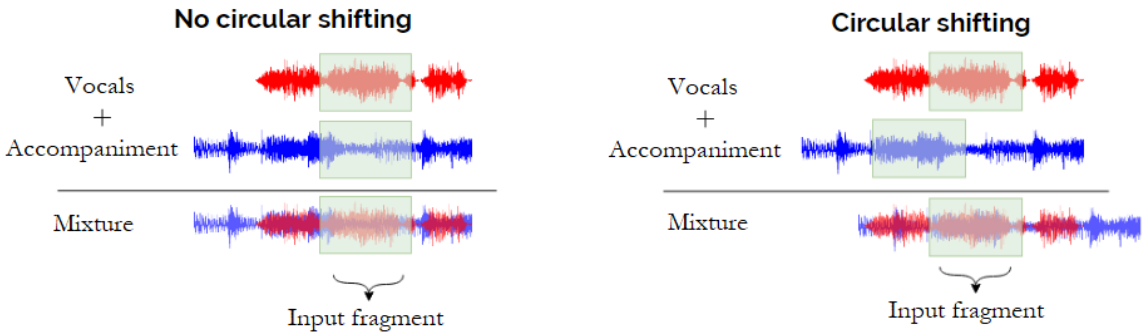

Figure 3.2: Circular shifting diagram

In this work, we study the applicability of the circular shifting data augmentation technique for training end-to-end models. Specifically, three different configurations are considered regarding the percentage of shifted samples used during training: 0%, 50%, and 100%. Thus meaning that for 0% no circular shifting is used (only the original mixings are used); for 50% half of the training examples are based on the original mixing and the other half training examples are based on circular shifting (an artificial *novel* mixing); and for 100% means that all training examples are based on the artificial circular shifting mixing.

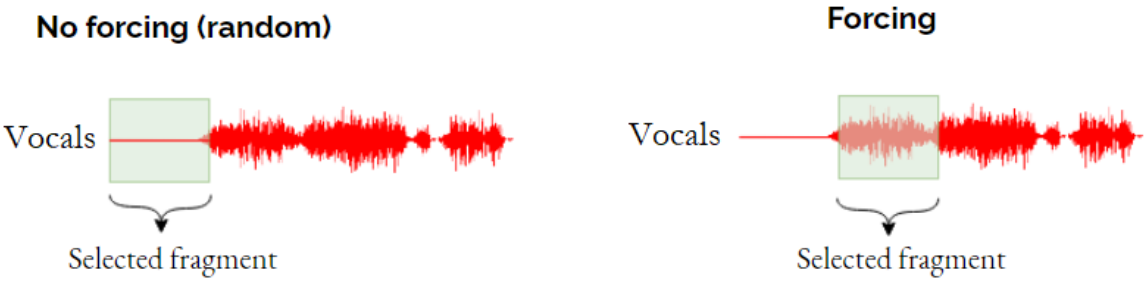

Figure 3.3: Forcing singing voice diagram

Forcing singing voice – after observing the difficulties of the model to produce singing voice in a continuous way, we decided to control which proportion of the data

contains singing voice (no silence). The percentage of forced fragments containing singing voice ranges from 0% to 90% in steps of 10% – 0% meaning that it is left at random to select singing voice segments, and 90% meaning that our sampling strategy ensures that nine out of ten examples contain singing voice.

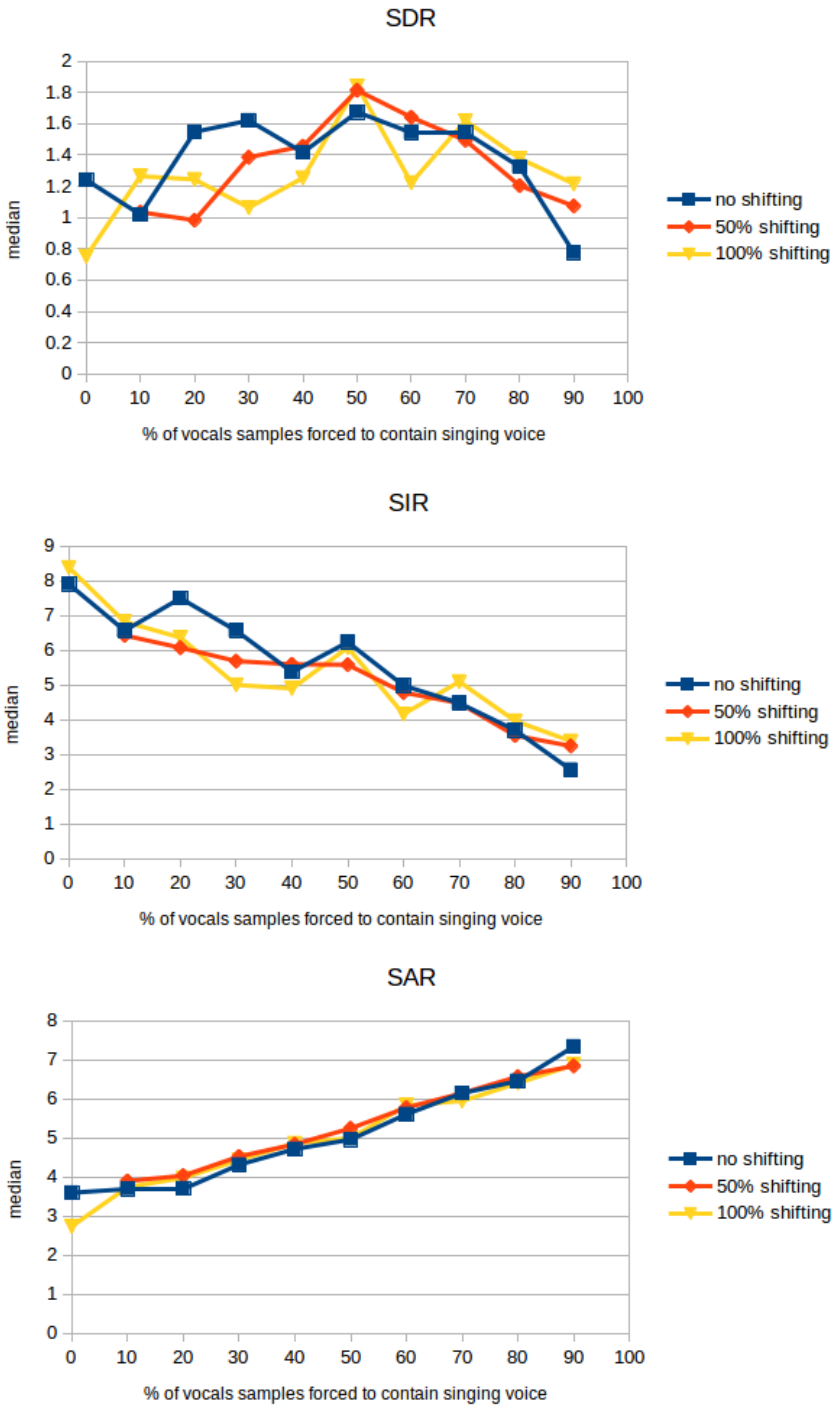

Figure 3.4: Vocals results by applying circular shifting and forcing singing voice

Results – are summarized in Figure 3.4, best SDR result is obtained when balanced results of SIR and SAR are achieved. This happens when 50% of the sampled vocals fragments are forced to contain no-silence. SIR metrics decreases as the vocals samples are forced to contain singing voice. Regarding SAR metrics, they have the opposite behaviour. A good analogy of the results obtained is to imagine a tap which controls the amount of singing voice it let pass. If the data relationship between silence and singing voice feeds the network with its the real proportions found in the musdb18 dataset, the network generates singing voice without almost leakage. However, it generates singing voice in chunks and with lots of artificial artifacts. The more we force the network to be fed without silence in the singing voice, the more the network will generate singing voice in a continuous manner but also add leaking from other sources. Regarding the shifting, there is not a clear influence of it in the final result. For this reason, input fragments in the following experiments are generated by applying shifting every time in the vocals stream – in order to virtually increase our training set to have as many different datatpoints as possible shiftings in the dataset.

3.1.2 Data augmentation: drums reinforcement

A large proportion of the leaking comes from the drum source – e.g., when the model confuses the snare drum sound with consonants. For this reason, we reinforce the drums *concept* by superposing a shifted version of it to the original accompaniment.

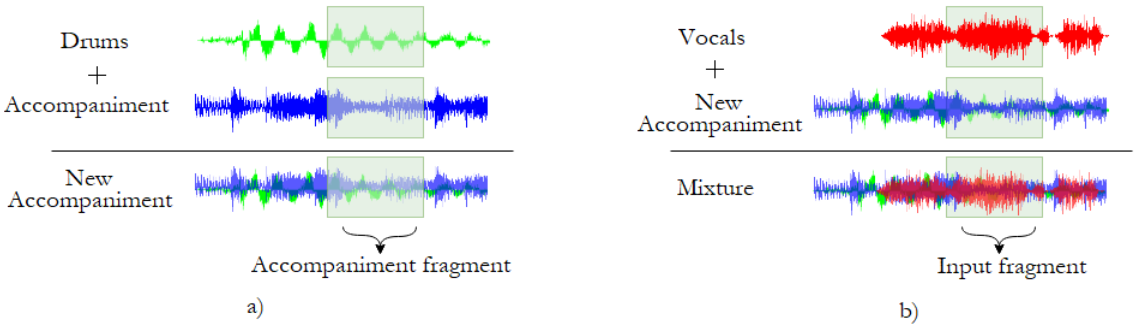

Figure 3.5: Drums data augmentation process: a) Creating new accompaniment fragment. b) Adding it to the vocals stream

	Vocals			Accompaniment		
	SDR	**SIR**	**SAR**	**SDR**	**SIR**	**SAR**
Forcing 50% singing voice	**1.84**	6.08	4.99	**9.48**	11.82	13.38
Forcing 50% singing voice + drums reinforcement	1.46	5.36	4.96	8.88	10.97	13.78
Forcing 20% singing voice	1.24	6.37	3.98	8.56	10.88	13.21
Forcing 20% singing voice + drums reinforcement	1.61	8.32	3.58	8.65	10.75	13.95

Table 3.2: Median values of the evaluation metrics

The main goal of using this type of data augmentation was to achieve less leakage from the drums source (thus increasing SIR metric) while maintaining the same or less amount of artifacts produced by the algorithm (thus maintaining or increasing the SAR metric). A seen in table 3.2, using the best model from the previous configuration plus drums reinforcement it is not achieved the expected result. On the contrary, SIR decreases while SAR remains constant. The desired effect of applying drums reinforcement is accomplished in the model where 20% of the fragments are forced to contain singing voice. However, the overall performance was not better than the one achieved before. For that reason, this type of data augmentation was no longer taken into consideration.

3.1.3 Architecture study: deeper? wider?

In this set of experiments, we aim to investigate if it is more important to have a deeper model (having less receptive field), or it is more important to have a larger receptive field (at the cost of having a shallower model) – provided that, for both cases, the GPU's memory is limited. In short, we try to address the following question: how wider/deeper should Wavenet-like models be?

We explore the tradeoff defined between the number of learnable parameters (the GPU's memory is occupied by a wide model that has many parameters per layer) and the receptive field length of the network (the GPU's memory is occupied with the stored feature maps). The amount of learnable parameters is determined by the number of convolutional filters in every layer – i.e., how wide the layer is. And the

receptive field length of the network is given by how "deep" the model is – which, for the Wavenet model, is determined by the number of dilations and stacks. In this case, the number of dilations remains constant (as described in our baseline architecture) and only the number of stacks is changed. The following models are under study:

	# stacks	**# filters**	**# params**	**receptive field length (ms)**
model 1	1	512	≈ 25.7M	2047 (128)
model 2	2	256	≈ 13.6M	4093 (256)
model 3	3	128	≈ 6.3M	6139 (384)
model 4	4	64	≈ 3.3M	8185 (512)
model 5	5	32	≈ 2.2M	10231 (639)

Table 3.3: Description of the models under study. **# filters** – number of filters per layer. **# params** – number of learnable parameters.

Results – The number of filters per layer has more importance than the number of stacks in the final performance. This means that is more important for the model to have a certain amount of learnable parameters than to have a bigger view of the input fragment. Interestingly, we have observed that larger receptive fields do not contribute into predicting a more continuous singing voice having fewer artifacts

	Vocals			**Accompaniment**		
	SDR	**SIR**	**SAR**	**SDR**	**SIR**	**SAR**
model 1	1.43	5.23	5.01	**9.75**	12.48	13.75
model 2	1.75	5.97	5.00	9.30	11.79	13.21
model 3	**1.84**	6.08	5.00	9.48	11.82	13.38
model 4	1.10	4.74	5.06	8.84	11.11	12.81
model 5	0.87	4.43	5.00	8.77	10.88	13.54

Table 3.4: Architecture deepness median results

On the other hand, a certain amount of learnable parameters helps the model to be more selective when separating sources. In fact, no more than 6.3 M parameters seem to be needed for achieving the best performance. This results in a very parameter efficient architecture compared to other state-of-the-art architectures [37, 32] that have almost 20M parameters. Interestingly enough, the model that results in with the best SDR for vocals is the original baseline system we proposed in section 3.0.3. Finally, note that the models are trained considering the previously described data

augmentation techniques: 100% circular shifting, and 50% forcing singing voice.

3.1.4 Changing the cost function

The settings proposed for the baseline architecture consider a single term loss – which only cares about the quality of the extracted singing voice: $\mathcal{L}_1 = \mathcal{L}(\hat{s}_t)$. Where $\mathcal{L}(\hat{s}_t) = |s_t - \hat{s}_t|$. Given that the model predicts a single output, the accompaniment is given by substracting the estimated singing voice from the mixture: $\hat{a}_t = m_t - \hat{s}_t$

In order to see how accompaniment prediction could affect the final performance of the singing voice prediction, we have adapted the model to predict both singing voice and accompainement (hence the model predicts two outputs), and we reformulate the loss as follows: $\mathcal{L}_2 = \mathcal{L}(\hat{s}_t) + \mathcal{L}(\hat{a}_t)$ where $\mathcal{L}(\hat{a}_t) = |a_t - \hat{a}_t|$.

To enforce energy conservation, another term that forces the sum of predicted sources to be equal to the mixture signal is introduced. Loss becomes: $\mathcal{L}_3 = \mathcal{L}(\hat{s}_t) + \mathcal{L}(\hat{a}_t) + \mathcal{L}(\hat{m}_t)$ where $\mathcal{L}(\hat{m}_t) = |m_t - \hat{m}_t|$. Note that $\hat{m}_t = \hat{s}_t + \hat{a}_t$. Energy conserving loss has been found useful in source separation tasks as [5, 43].

Aforementioned losses only care about the similarity between the prediction and the target. However, in source separation tasks is important to avoid leaking from other sources – which means a higher signal to interference ratio (SIR). For this reason, dissimilarity losses have been found successful in some approaches [6, 5]. Therefore, we propose to use a loss term that also takes into account the dissimilarity between the prediction and other sources: $\mathcal{L}_{diss} = \sum_{j=1}^{J} |\hat{y}_j - y_{\hat{j} \neq j}|$. Where $\hat{y}$ are the predicted sources.

Results – The best performing model from previous analysis (50% of the sampled vocals fragments are forced to contain no-silence and 100% circular shifting) is used to see how the proposed losses affect the final performance.

Results show that the accompaniment prediction ($\mathcal{L}_2$) does not help to have a better

	Vocals			Accompaniment		
	SDR	**SIR**	**SAR**	**SDR**	**SIR**	**SAR**
$\mathcal{L}_1$	**1.84**	6.08	4.99	**9.48**	11.82	13.38
$\mathcal{L}_2$	1.45	4.88	5.16	9.23	11.52	13.03
$\mathcal{L}_3$	1.35	4.95	5.19	9.14	11.38	12.98
$\mathcal{L}_1 - \gamma\mathcal{L}_{diss}$	1.13	4.85	5.02	9.00	11.61	12.74
$\mathcal{L}_2 - \gamma\mathcal{L}_{diss}$	1.17	4.24	5.29	8.92	11.15	13.17

Table 3.5: Changing singing voice separation losses median results

singing voice separation – and the same behaviour is observed when enforcing energy conservation by adding the mixture loss term ($\mathcal{L}_3$). A tentative explanation for this fenomena can be that the model is to small ($\approx$ 6.3M parameters) for being able to properly model how to separate the two sources. Furthermore, the dissimilarity losses do not help the network to be more selective. On the contrary, adding this loss term decreases the SIR value – the value γ is set to 0.002. Finally, it is worth mentioning that early experiments using a standard L2 loss proved to be disadvantageous.

3.1.5 Comparison with the state-of-the-art

For comparison with previous work, the time-frequency approach DeepConvSep [5] is adopted under comparable conditions.

		Wavenet			DeepConvSep		
		SDR	**SIR**	**SAR**	**SDR**	**SIR**	**SAR**
Vocals	Med	1.84	**6.08**	5.00	**1.95**	4.39	**7.74**
	Mean	1.14	5.71	4.92	1.37	3.88	7.35
Accompaniment	Med	9.48	11.82	13.38	**10.37**	**14.95**	**13.25**
	Mean	9.86	12.61	13.91	10.91	15.46	13.30

Table 3.6: Singing Voice Separation models comparison

Objective results show that both systems achieve a similar overall performance for singing voice source separation. The main differences are related to the SIR and SAR metrics. Wavenet achieves a better SIR – which means that there is less distortion coming from other sources in the predicted vocals. For example, Wavenet would be useful when planning to use the extracted vocal track in a new context. On the contrary, DeepConvSep offers less degradation on the signal caused by the separation algorithm – a better SAR. Informal listening tests show that the proposed Wavenet model struggles when predicting long vocal notes, where more artifacts are introduced.

3.2 Experiment 2. Multi-Instrument Separation.

In this section we experiment with a model capable to handle the separation of many instruments at the same time: vocals, drums and bass. For doing so, we make use the architecture that performed the best for the task of singing-voice separation. Regarding which data augmentation techniques to use, the technique is only practiced to the vocals stream using the best configuration found in experiment 1: 100% circular shifting, and 50% forcing singing voice.

Throughout this section, we mainly experiment with several loss functions – and then we compare our best performing model with a state-of-the-art approach based on processing spectrograms, the DeepConvSep model [5].

3.2.1 Multi-instrument architecture

The main difference between the multi-instrument and the singing voice separation architecture is found in the output layer which, in this case, linearly projects the last feature map into a three-channel temporal signal by using a 1x3 filter. Then, each of this channels is associated with an instrument category: vocals, drums or bass.

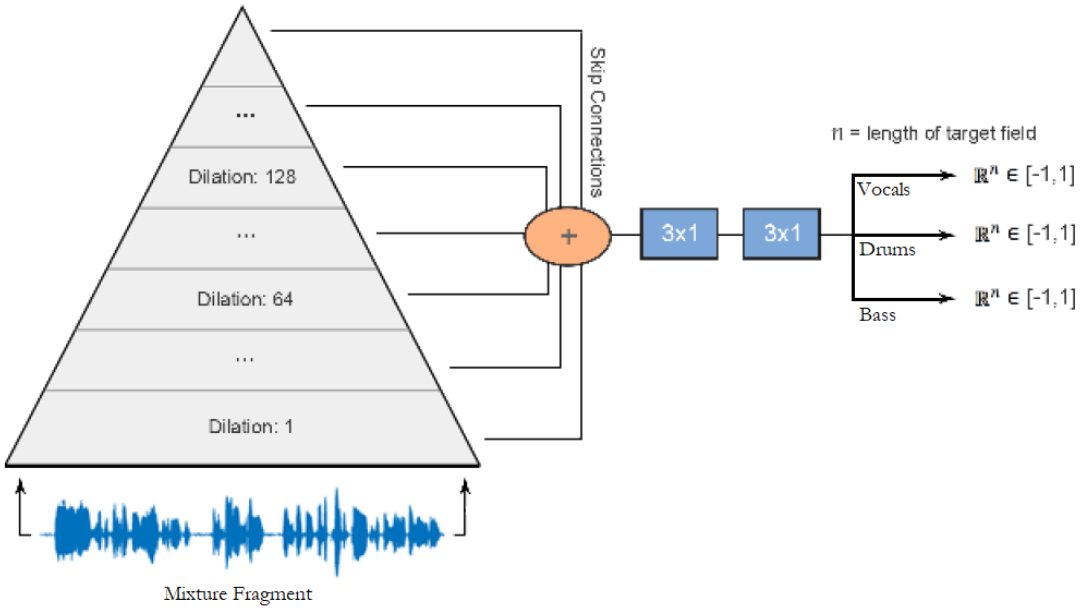

Figure 3.6: Ensuring singing voice diagram

3.2.2 Changing the cost function

The baseline model we propose is based on a three term loss which cares only about the quality of the extracted singing voice, drums and bass: $\mathcal{L}_3 = \mathcal{L}(\hat{s}_t) + \mathcal{L}(\hat{d}_t) + \mathcal{L}(\hat{b}_t)$, where $\mathcal{L}(\hat{s}_t) = |s_t - \hat{s}_t|$, $\mathcal{L}(\hat{d}_t) = |d_t - \hat{d}_t|$, and $\mathcal{L}(\hat{b}_t) = |b_t - \hat{b}_t|$

Following the same idea as in section 3.1.4, we study if the prediction of another source helps to predict the desired ones. The loss is reformulated by taking into account the prediction of the *others* (the rest of the accompaniment carried in the *other* audio stream): $\mathcal{L}_4 = \mathcal{L}(\hat{s}_t) + \mathcal{L}(\hat{d}_t) + \mathcal{L}(\hat{b}_t) + \mathcal{L}(\hat{o}_t)$ where $\mathcal{L}(\hat{o}_t) = |o_t - \hat{o}_t|$.

In order to enforce energy conservation, another term that forces the sum of predicted sources to be equal to the mixture signal is introduced. Then, the loss becomes: $\mathcal{L}_5 = \mathcal{L}(\hat{s}_t) + \mathcal{L}(\hat{d}_t) + \mathcal{L}(\hat{b}_t) + \mathcal{L}(\hat{o}_t) + \mathcal{L}(\hat{m}_t)$ where $\mathcal{L}(\hat{m}_t) = m_t - \hat{m}_t$. Note that $\hat{m}_t = \hat{s}_t + \hat{d}_t + \hat{b}_t + \hat{o}_t$.

Finally, it is also proposed a loss term that takes into account the dissimilarity between the prediction and the other sources. $\mathcal{L}_{diss} = \sum_{j=1}^{J} |\hat{y}_j - y_{\hat{j} \neq j}|$. Where $\hat{y}$ are the predicted sources – see section 3.1.4 for more information.

3.2.3 Results

	Vocals			Drums			Bass		
	SDR	**SIR**	**SAR**	**SDR**	**SIR**	**SAR**	**SDR**	**SIR**	**SAR**
$\mathcal{L}_3$	**0.99**	12.46	1.68	**2.44**	10.51	4.35	**1.13**	8.57	3.91
$\mathcal{L}_4$	0.94	3.60	5.61	2.06	8.38	5.01	0.86	3.73	6.19
$\mathcal{L}_5$	0.67	2.24	7.62	**2.51**	5.72	6.57	-0.047	2.76	5.97
$\mathcal{L}_3 - \gamma\mathcal{L}_{diss}$	0.53	3.28	5.51	2.03	7.35	5.09	0.40	2.69	6.84

Table 3.7: Changing multi-instrument losses median results

Similarly to the results in singing voice separation, predicting more sources than the targeted ones does not help to improve objective metrics. This is clearly observed when analyzing SIR values which drop drastically when predicting also the *other* source. The system is confused and it is not able to learn properly the timber of each source. In addition, enforcing energy conservation only helps for drums prediction but significantly deteriorate vocals and bass. As happened with singing voice separation, dissimilarity losses do not help the network being discriminatory. On the contrary: by adding this loss term, the SIR drops. The value γ is set to be 0.001. In conclusion: better results are achieved when the loss only cares about the quality of the targeted sources.

3.2.4 Comparison with the state-of-the-art

The objective results we got show that the performance of the model varies depending on the source. DeepConvSep predicts better the vocals than the Wavenet for Music Source Separation. However, drums and especially the bass are better separated by the Wavenet for Music Source Separation. Furthermore, we observe that all predicted sources the Wavenet for Music Source Separation achieves better SIR scores than DeepConvSep while DeepConvSep achieves better SAR than the Wavenet for Music Source Separation.

Interestingly, when we compare these results with the ones obtained with singing voice separation Wavenet (in section 3.1.5), we observe that SIR and SAR values

improved significantly when predicting many instruments at the same time – for the Wavenet for Music Source Separation and DeepConvSep, respectively.

		Wavenet			**DeepConvSep**		
		SDR	**SIR**	**SAR**	**SDR**	**SIR**	**SAR**
Vocals	Median	0.99	12.46	1.68	**1.68**	3.64	8.77
	Mean	0.47	11.11	1.86	**1.29**	3.30	8.29
	SD	4.08	6.14	2.83	4.29	4.60	2.71
Drums	Median	**2.44**	10.51	4.35	2.27	5.98	6.07
	Mean	**2.70**	10.29	4.58	2.53	5.95	6.71
	SD	4.10	5.83	3.38	3.80	4.38	2.69
Bass	Median	**1.13**	8.57	3.91	-0.13	1.15	7.54
	Mean	**0.69**	7.66	3.35	-0.57	1.43	7.36
	SD	4.12	5.63	3.22	3.52	4.45	1.85

Table 3.8: Music Source Separation models comparison

Chapter 4

Conclusions and future work

We have proposed an end-to-end learning method for music source separation to assist recording engineers, musicians, and similar end-users in the task of music remixing and content creation. More specifically, we investigated if it is possible to approach the problem of music source separation in an end-to-end learning fashion with a Wavenet model.

The non-causal discriminative adaptation of Wavenet that we use for music source separation learns in a supervised manner – via optimizing the parameters of the network, following the guidance of a regression loss. By removing the autoregressive nature of Wavenet, we are able to predict target fields (instead of a single sample at a time) and thus overcome the time-complexity of the original Wavenet.

Initially, it was challenging for the model to extract a singing voice track that had smooth onset transitions. For this reason, various forms of data augmentation, architectural changes, and learning conditions were explored. In our experiments we observed that it exists a trade-off between how selective the model is when separating a source, and how smooth is the onset transition. If we tailor the model towards having smoother transitions (with data augmentation, for example), the model also introduces leakage from other sources. However, if the model is very selective (there is not that much interference coming from other sources) then the onset transitions are very abrupt.

After this initial investigation, we propose model that is very parameter-efficient. The fact that is conformed by a reduced number of learnable parameters, helps increasing the generalization capability of the model – which is capable to effectively separate songs under conditions it has never been exposed to. In fact, our results sup-

port that our end-to-end learning model is able to achieve a similar performance than a spectrogram-based architecture – the DeepConvSep model [5]. This result confirms that it is possible to directly approach the problem of source separation from the raw audio instead of using magnitude spectrograms.

The implementation of the model will be soon publicly available online: `github.com/francesclluis/music-source-separation-wavenet` – and several audio examples are provided to perceptually asses the performance of the model.

As future work, it would be interesting to train the proposed model with additional data – since it has been proven to be the key for achieve better performance in data-driven models [37]. Moreover, it might be also interesting to investigate how this Wavenet for music source separation helps to improve other MIR tasks – which can perform better if the separated sources are used for their analysis.

Bibliography

[1] Cédric Févotte, Emmanuel Vincent, and Alexey Ozerov. Single-channel audio source separation with nmf: divergences, constraints and algorithms. In *Audio Source Separation*, pages 1–24. Springer, 2018.

[2] Michael A Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[3] Xavier Serra, Michela Magas, Emmanouil Benetos, Magdalena Chudy, Simon Dixon, Arthur Flexer, Emilia Gómez, Fabien Gouyon, Perfecto Herrera, Sergi Jorda, et al. Roadmap for music information research, 2013.

[4] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 signal separation evaluation campaign. *arXiv preprint arXiv:1804.06267*, 2018.

[5] Pritish Chandna, Marius Miron, Jordi Janer, and Emilia Gómez. Monoaural audio source separation using deep convolutional neural networks. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 258–266. Springer, 2017.

[6] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Deep learning for monaural speech separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 1562–1566. IEEE, 2014.

[7] Axel Roebel, Jordi Pons, Marco Liuni, and Mathieu Lagrangey. On automatic drum transcription using non-negative matrix deconvolution and itakura saito divergence. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 414–418. IEEE, 2015.

[8] Dario Rethage, Jordi Pons, and Xavier Serra. A wavenet for speech denoising. *arXiv preprint arXiv:1706.07162*, 2017.

[9] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. Segan: Speech enhancement generative adversarial network. *Proc. Interspeech 2017*, pages 3642–3646, 2017.

[10] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, Dinei Florêncio, and Mark Hasegawa-Johnson. Speech enhancement using bayesian wavenet. In *Proc. Interspeech*, pages 2013–2017, 2017.

[11] Shrikant Venkataramani, Jonah Casebeer, and Paris Smaragdis. End-to-end source separation with adaptive front-ends. *arXiv preprint arXiv:1705.02514*, 2017.

[12] Emmanuel Vincent, Nancy Bertin, Rémi Gribonval, and Frédéric Bimbot. From blind to guided audio source separation: How models and side information can improve the separation of sound. *IEEE Signal Processing Magazine*, 31(3):107–115, 2014.

[13] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[14] Guy J Brown and Martin Cooke. Computational auditory scene analysis. *Computer Speech & Language*, 8(4):297–336, 1994.

[15] Simon Haykin and Zhe Chen. The cocktail party problem. *Neural computation*, 17(9):1875–1902, 2005.

[16] Emmanuel Vincent, Shoko Araki, Fabian Theis, Guido Nolte, Pau Bofill, Hiroshi Sawada, Alexey Ozerov, Vikrham Gowreesunker, Dominik Lutter, and Ngoc QK Duong. The signal separation evaluation campaign (2007–2010): Achievements and remaining challenges. *Signal Processing*, 92(8):1928–1936, 2012.

[17] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[18] Paris Smaragdis and Michael Casey. Audio/visual independent components. In *Proc. ICA*, pages 709–714, 2003.

[19] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

[20] Emad M Grais, Dominic Ward, and Mark D Plumbley. Raw multi-channel audio source separation using multi-resolution convolutional auto-encoders. *arXiv preprint arXiv:1803.00702*, 2018.

[21] Tuomas Virtanen. Unsupervised learning methods for source separation in monaural music signals. In *Signal Processing Methods for Music Transcription*, pages 267–296. Springer, 2006.

[22] Shlomo Dubnov. Extracting sound objects by independent subspace analysis. In *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*. Audio Engineering Society, 2002.

[23] Gil-Jin Jang and Te-Won Lee. A maximum likelihood approach to single-channel source separation. *Journal of Machine Learning Research*, 4(Dec):1365–1392, 2003.

[24] Thomas Blumensath and M Davies. Unsupervised learning of sparse and shift-invariant decompositions of polyphonic music. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 5, pages V–497. IEEE, 2004.

[25] Samer A Abdallah and Mark D Plumbley. If the independent components of natural images are edges, what are the independent components of natural sounds. In *Proceedings of International Conference on Independent Component Analysis and Signal Separation (ICA2001)*, pages 534–539, 2001.

[26] Samer A Abdallah and Mark D Plumbley. An independent component analysis approach to automatic music transcription. *Preprints-Audio Engineering Society*, 2003.

[27] Mohit Dubey, Garrett Kenyon, Nils Carlson, and Austin Thresher. Does phase matter for monaural source separation? *arXiv preprint arXiv:1711.00913*, 2017.

[28] Nathanaël Perraudin, Peter Balazs, and Peter L Søndergaard. A fast griffin-lim algorithm. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, pages 1–4. IEEE, 2013.

[29] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *ISMIR*, pages 477–482, 2014.

[30] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[31] Naoya Takahashi and Yuki Mitsufuji. Multi-scale multi-band densenets for audio source separation. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017 IEEE Workshop on*, pages 21–25. IEEE, 2017.

[32] Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. 2017.

[33] Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent. Multichannel music separation with deep neural networks. In *Signal Processing Conference (EUSIPCO), 2016 24th European*, pages 1748–1752. IEEE, 2016.

[34] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*, 2017.

[35] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*, 2017.

[36] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. Crepe: A convolutional representation for pitch estimation. *arXiv preprint arXiv:1802.06182*, 2018.

[37] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185*, 2018.

[38] Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. *arXiv preprint arXiv:1711.00541*, 2017.

[39] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[40] BSS Eval. A toolbox for performance measurement in (blind) source separation. version 3.0.

[41] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

[42] Marius Miron, Jordi Janer Mestres, and Emilia Gómez Gutiérrez. Generating data to train convolutional neural networks for classical music source separation. In *Lokki T, Pätynen J, Välimäki V, editors. Proceedings of the 14th Sound and Music Computing Conference; 2017 Jul 5-8; Espoo, Finland. Aalto: Aalto University; 2017. p. 227-33.* Aalto University, 2017.

[43] Jordi Pons, Jordi Janer, Thilo Rode, and Waldo Nogueira. Remixing music using source separation algorithms to improve the musical experience of cochlear implant users. *The Journal of the Acoustical Society of America*, 140(6):4338–4349, 2016.