Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

# PodcastMix: A dataset for separating music and speech in podcasts

Nicolás Schmidt

**Supervisor:** Marius Miron

**Co-Supervisor:** Jordi Pons

August 2021

Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

# PodcastMix: A dataset for separating music and speech in podcasts

Nicolás Schmidt

**Supervisor:** Marius Miron

**Co-Supervisor:** Jordi Pons

August 2021

# Contents

## Acknowledgement

I would like to thank Marius for his tremendous willingness, patience and guidance throughout the development of this thesis. To Jordi to accept being the co-supervisor, for his hours, patience, follow up, good vibes and attention to detail. To Martín Haro and Pedro Muñoz for providing me with his podcasts sessions. Also to Caro, who was with me when I was finishing the thesis while she was organizing our trip by her own. To my family for always being supportive. Finally to Tamarites for making this year a community experience like no other.

## Abstract

Over the last few years, the popularity of podcast shows in streaming services has increased considerably. Licensed music in these shows is frequently used, but the precision of song identification services could be affected by the speakers voice in the mix. This presents a major problem both for the musicians, who do not receive their respective royalty payments, and for the broadcasters, who may be exposed to legal problems for non-compliance with international copyright laws. In this Master Thesis, a benchmark between two state of the art models for music source separation, the ConvTasNet and the UNet, was performed against a novel Podcast-like audio dataset called PodcastMix with the objective of separating both the voice of the speakers and the background music from a podcast. In this way, the background music and foreground speech source separation task was formalized. This new dataset is compound by music from the Jamendo free music streaming service, mixed with the VCTK speech dataset. The models were trained on this dataset and evaluated both in the test partition and on a dataset of real podcasts. The results show that UNet performs better than ConvTasNet in separating speakers and music from podcasts. The benchmark was performed using the Asteroid toolkit and the evaluation metrics were computed using BSSEval tool in order to measure the quality of the separations.



Keywords: PodcastMix, Music Speech Source Separation, Deep Learning, Dataset, Podcast, Radioshow, Background music foreground speech.

# Chapter 1

# Introduction

During the last few years, the production and consumption of podcasts has massively increased. According to the 2020 research conducted by Edison Research and Triton titled The Infinite Dial, 75% of Americans aged 12 and older are familiar with the concept Podcasts and 37% listen to some type of podcast on a monthly basis, up from 70% and 32% respectively identified in 2019 [1].

This upward trend in podcast consumption is directly related with an increasing variety of podcast offer. Today people can find Podcasts on any topic; fashion, vehicles, lifestyle, music, work, entrepreneurship, among many other categories. This increase in the diversity of offerings has also allowed consumer interest to diversify. For example, in 2010, only 30% of the U.S. audience that consumed podcasts were in an ethnic demographic segment. By 2016, that percentage had increased to 36% [2]. In 2020, according to [1], the percentage of people belonging to the ethnic demographic segment in the U.S. who listen to podcasts on a monthly basis reached 37%, getting closer to the 42% of U.S. residents over the age of 12 who belong to this group.

The emergence of Podcasts as the new on-demand Radio Shows, brings with it technical and legal challenges typical of any broadcasting network. Among the most important issues is compliance with copyright regulations in force in each country where the Podcast is consumed. The use of music and other sound resources

that may be subjected to country-specific licenses represents a great challenge that involves a large number of players in the industry: musicians, broadcasters, labels, radio stations, streaming platforms, listeners and even governments.

This problem has only been partially solved by some streaming platforms such as Spotify. The company run by Daniel Ek allow podcast creators to upload their content to the streaming platform. However, they were not allowed to include commercially licensed music on the recordings. Just a couple of months ago Spotify allowed podcast creators to add commercially licensed music to their creations, as long as the podcasts were created using Spotify's Anchor software, a tool to help content creators to address the recording and post-production tasks of their Podcasts. Anchor allows creators to select any song from Spotify's catalog and use it as background music or as a music block inside the Podcast.

With this tool, Spotify made music sharing available to podcast creators, allowing rightholders to get what they are legally entitled to for the playback of their licensed songs. However, for podcasts that have not been created using Anchor, as well as for other podcast distribution services that do not have an integrated music catalog, the issue of song identification and therefore royalty distribution for commercial music remains a big problem.

With the massification of technologies and new advances in the field of deep learning, new possibilities are opening up to solve this problem. One of the most widely exploited methods for automatic song recognition is the use of acoustic fingerprinting algorithms. This technique consist on the creation of an acoustic summary of a signal, computed from the audio features contained in the signal itself.

Perhaps the best known mass user-facing application that uses this technology is the Shazam. Shazam is a mobile application that allows users to identify songs using their phone's microphone. The app performs impressively well, accurately identifying songs, artists and even specific versions of a song with just a couple of seconds of audio. Shazam's fingerprinting algorithm is based on the calculation of a series of acoustic features from the audio. Moreover, the speed with which the

application can recognize a given song is based on a robust data structure, which allows for effective indexing of the audio descriptors.

Fingerprinting algorithms work the same way that humans recognize sounds: they are based on acoustic features. Thus, any music fingerprinting algorithm should work well even in contexts where the song may have background noise, been modified in pitch or even in tempo. However, in podcast contexts, where the background music generally sounds fainter than the speaker's voice and where there are conversations going on at the same time, the effectiveness of song identification algorithms is affected.

To allow better song identification, source separation techniques can be used. Source separation is a Music Information Retrieval (MIR) task consisting on the implementation of algorithms and systems that, starting from a mixed signal, are able to separate the source signals, isolating them from one another.

Source separation in the context of music information retrieval is often considered the holy grail task. The reason for this is that the extraction of clean sources can be used to then feed other systems that perform other MIR tasks, such as pitch detection, beat estimation, key estimation, lyrics/music alignment, speech enhancement, among others.

In order to contribute to the development of new and better models based on supervised learning for source separation, the Podcastmix dataset is presented in this Master Thesis. This dataset consist of a mix between music licensed under creative commons uploaded to the Jamendo music streaming service and the VCTK dataset, which consists of recordings of small phrases recorded by two different microphones at high quality. The music on the dataset is a subset of the most popular songs from the creative commons music platform Jamendo to ensure the musical quality of the songs.

The dataset is presented along with a benchmark of the state-of-the-art methods in source separation tasks, in order to have a baseline of comparison between new models to be developed and evaluated using this new dataset. The models evaluated

on the dataset correspond to both time-frequency and waveform-based models. The implementations of these models are based on the Pytorch toolkit Asteroid. The evaluation of the models was performed using the ps_bss_eval module, which allows the computation of the most commonly used metrics in the source separation field.

This document is structured as follows. First, the state of the art regarding source separation models and available datasets is presented. In this section, both analytical methods and data-based models are reviewed. Then, the methodology for the elaboration of the dataset and the pipeline to evaluate the different models implemented in Asteroid is described. Later, the final dataset and its most important features are presented. The results obtained by computing the evaluation metrics of the different models trained with the presented dataset are also showed. Finally, the conclusions and the contributions that the PodcastMix brings to the Music Information Retrieval community are described.

# Chapter 2

# State of the Art

In the present chapter, the most important techniques developed in the field of source separation are presented. The chapter is divided into five sections. In the first section, the problem of audio source separation is defined and discussed. Later on, a list of the most important analytical approaches to this relevant field are presented. Then, the state-of-the-art deep learning approaches to accomplish this task are discussed. In this section the most important architectures for the audio source separation task are presented. Following, the evaluation methods to measure the effectiveness of the source separation strategies are presented and discussed, along with a benchmark between the state of the art architectures. Finally, the main open source tool-kits to perform source separation are listed.

## 2.1 Audio Source Separation

Source Separation is the process of isolating individual sounds in an auditory mixture of multiple sounds [3]. In particular, audio source separation is a task which main goal is to blindly separate the audio sources that compounds the input audio mix. In other words, the task of audio source separation aims is to create a model or system where a mixed signal $x(t)$ is input to the system. The system assumes that $x(t)$ is defined by the following formula:

$$x(t) = \sum_{i=1}^{N} g_i x_i(t)$$

Where $x(t)$ is the mixture signal that is compound by $N$ sources $x_i(t)$, multiplied by a weight (gain) factor of $g_i$. The main of the source separation task is to obtain one or several $x_i$ by only processing $x(t)$ .

Source separation in the context of music is a particularly difficult task for several reasons. First, the sources in a musical context usually suffer changes at the same time, for example a chord change. This means that the sources are highly correlated. Secondly, the tools developed over the bases of source separation methods tend to have a higher bar quality exigences, since the end-users are usually musicians, music producers or labels.

Source separation is considered the holy grail of the Music Technology field, since any tool that allows to clean out a source of a mix could help to improve the performance of all the other fields in Music Information Retrieval. Having cleaner sources improves the tasks of beat estimation, chord recognition, key estimation, music transcription, lyrics alignment, fundamental frequency analysis, among others [3].

### 2.1.1 Low-rank approximation methods

Historically, audio source separation was first approached from an analytical perspective. This means that the first methods developed to perform signal source separation were adapted from electrical engineering. The first audio source separation method was introduced by Jeanny Herault and Christian Jutten in 1985 in the context of a statistical framework [4]. In this section, the main analytical approaches for source separation are presented.

#### Principal Component Analysis

Principal Component Analysis or PCA is a statistical method to identify the dynamics and underlying behaviour of a set of data, signal or system. PCA works by

decomposing the data into N different modes concurrently and non-recursevely. The generated modes are a linear decomposition of the original signal. The attributes of the data are normalized and its mean are centered by the PCA, right before passing it to a Variational Mode Decomposition (VMD) [5].

VMD Concurrently and non-recursively decomposes a signal into a variety of modes such that the combination of all modes will recreate the original signal. It also looks for the center frequency of the modes, and this center frequency is band-limited for each mode. Thus, it simultaneously decomposes a real valued signal into a finite number of modes. Such modes have unique sparsity properties, and along with the modes, the center frequency is also calculated [6].

From a random set of data, PCA describes the dynamics and behavior of a system. Low level reductions of the signal can be achieved, even without any previous knowledge of the underlying data distributions. PCA can be achieved by decomposing a data covariance matrix by its own value or by decomposing a data matrix by a singular value (SVD), usually after the mean centering and normalization of the data matrix for each attribute.

Over a mixed signal $x(t)$ given by

$$x(t) = g_1 x_1(t) + g_2 x_2(t) + n(t)$$

where $x(t)$ is the mixed signal, and $x_1(t)$ and $x_2(t)$ are the two sources. Here, the weighting coefficients are $g_1$ and $g_2$ and $n(t)$ is noise that can be introduced into the system. The goal is to extract $x_1(t)$ and $x_2(t)$.

PCA consists of applying VMD to $x(t)$ and decomposing it into a variety of modes. Each mode belongs to a different spectral band. If the signal $x(t)$ is a mixture of source signals belonging to different spectral bands, then they are captured by the different modes in VMD. The high frequency noise of the out-band is recorded in other modes. If the noisy modes are refused, then the extracted source signals are the remaining modes. The PCA is used for selecting the appropriate signal modes. A set of observed correlated vectors are transformed by PCA into a set of

uncorrelated vectors. The restructuring is as follows:

$$z = A^T c$$

where, $A$ is an orthogonal matrix, $c$ is the set of correlated vectors or principal components, and $z$ is the set of uncorrelated vectors. The task then is to maximize the principal components of $z$, subjected to the restriction that the main components vectors are orthogonal to each other [7].

### Independent Component Analysis

Given a signal or a set of random variables, Independent Component Analysis (ICA) is used to discover underlying features. ICA is a statistical method that defines a generative model for the data. ICA assumes that the studied samples are a linear sum of underlying independent non-gaussian and mutually independent variables.

Independent Component Analysis consists of searching for a linear transformation that minimizes the statistical dependence between its components. This means that, in contrast to Principal Component Analysis, ICA tries to find not only the orthogonal linear transformations and it leads to better models [8]. However, this technique can only be applied to monophonic, 2-sources sum of linear signals [9]. This implies that this technique is more suitable for speech separation applications rather than polyphonic music. Another disadvantage of this technique is that ICA needs at least the same number of mixture observations as sources to separate [10].

### Independent Subspace Analysis

Independent Subspace Analysis is another statistical technique based on the Independent Component Analysis. However, this technique relaxes ICA's constraint to require at least as many mixture observations signals as input to the system. Another difference from ICA is that ISA uses dynamic components to represent non-stationary signals. The separated sources are tracked by similarity of dynamic components over small time frames. ISA proposed components grouping based on

the partition of matrix of independent component crossentropies called ixegram. In an audio section, the ixegram tests the reciprocal similarity of parts and the clustering of the ixegram yields the source subspaces and time trajectories [11].

#### Non-Negative Matrix Factorization

Non-negative Matrix Factorization is a source separation approach that performs in the time frequency domain. It represents the mixed source Y as the sum of the isolated independent signals. However, in contrast to the definition in the time-domain, here the independent signals are presented as the dot product between two vectors; the basis, that represents the frequency response of a particular source at a given time; and the the activation gains vector, the gain frequency response at a given point in time.

$$X_{i,j} = \sum_{n=1}^{N} B_{i,n} G_{n,j}$$

Where $X_{i,j}$ is the mixed signal and B and G are the basis and gain vectors for each of the N sources at a time j. Non-Negative Matrix Factorization aims to minimize the divergence between X and BG:

$$B, G = argmin_{B,G\geq} D(X, BG)$$

Where D represents the divergence function, which may be either the Kullback-Leiber Divergence or the Euclidean distance functions.

### 2.1.2 Masking

A mask is a matrix that has the same dimensions as the spectrogram of the mixed signal. This mask represents a spectrogram and each one of its time-frequency bins has a value from 0 to 1, which determines how much energy of the original mixture a source contributes. This means that a value of 1 for a particular bin will allow all the energy of that particular source to pass through the mixed source.

Masks are applied to the original mixture by multiplying element-wise both spectrograms, like a gate applied to the original mixed source spectrogram. This means

that a particular mask will generate a particular audio isolated source from the original mix. Therefore, if a mask $M^i \in [0.0, 1.0]^{TxF}$, represents the $i^{th}$ source, $Si$, in a mixture represented by a magnitude spectrogram, $|\mathrm{Y}| \in \mathbb{R}^{TxF}$, we can make an estimate of the source like so:

$$Si = M^i \odot |Y|$$

Where $\odot$ represent the element-wise multiplication operation. Therefore the estimated mask $M^i$ is such that it can produce a good estimate of source $Si$ when applied. Another characteristic of the estimated masks for the N sources is that:

$$\sum_{i=1}^{N} M^i = J$$

where $M^i$ is the $i^{th}$ mask estimated for each of the N sources that compounds the mix, and J is a only-ones matrix that has the same dimensions of the mixed source spectrogram. This also means that the source $i$ can be removed from the mix by inverting its mask $M^i$:

$$M_i^{-1} = J - M_i$$

then, $Y(t) \odot M_i^{-1}$ will return the mixed source without the $i^{th}$ source [3].

**REPET**

REPET, from Repeating Pattern Extraction Technique, is a time-frequency based analytical method to estimate a mask to perform source separation. It is based on the repeating characteristics of music. The basic concept behind the REPET algorithm is to classify the segments in the audio that repeats with regularity. Then it compares them against a repeating segment model derived from them, and to finally use time-frequency masking to remove the repeating patterns.

On the first stage of the algorithm, the magnitude spectrogram must be computed.

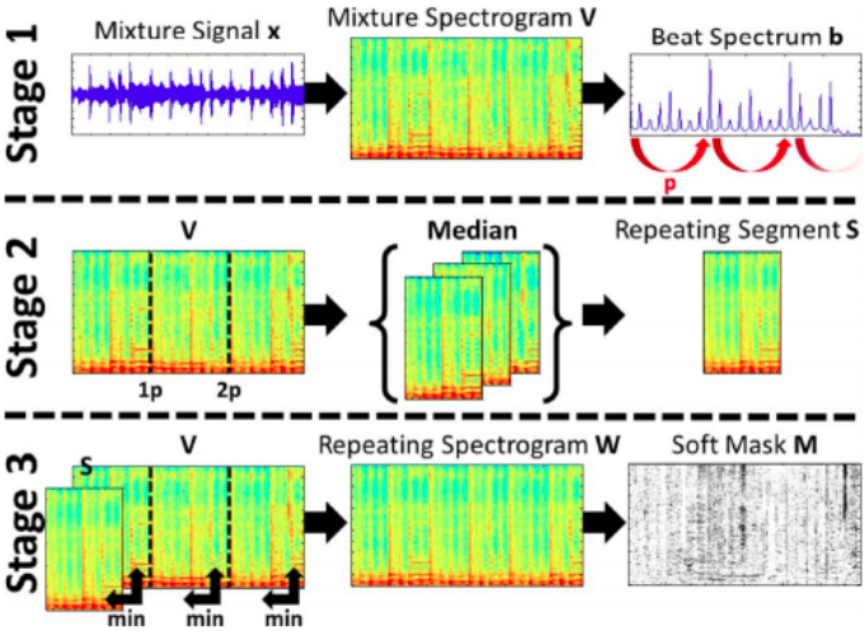


Figure 1: Overview of the REPET algorithm. Stage 1: calculation of the beat spectrum and estimation of the repeating period . Stage 2: segmentation of the mixture spectrogram and computation of the repeating segment model. Stage 3: derivation of the repeating spectrogram model and building of the soft time-frequency mask.

Then the autocorrelation is calculated for each bin of the spectrogram. For stereo signals, the average between the two channels is taken. Therefore, a beat spectrum is obtained, which allows to easily visualize the peaks, and to compute the moments in time for which a periodicity is more likely to occur.

Having the repetition period of the signal, the original signal is divided into L segments of equal size and, for each of these segments, the median between all the $l^{th}$ values of each of the segments is taken, as shown in the Figure 1.

In the last stage, this new matrix is used to compute a mask of the repetitions by taking the minimum of the element-wise values. Finally, the matrix is normalized using the absolute element-wise spectrogram. Thus, values between 0 and 1 are obtained for the whole softmask and this can be used to obtain the isolated voice by to multipling the resulting matrix by the original spectrogram element-wise [12].

### 2.1.3 Deep Learning approaches

With the rise of deep learning, several techniques have been developed in the field of source separation. These techniques had proven to have better performance than the analytical proposed approaches since the different encoders used are not arbitrary chosen but learned from the data. This section could also be referred to as data-driven approaches, since the perspective from which the problem is attacked is based on novel machine learning techniques applied to the estimation of encoders and decoders learnt directly from the provided data.

In this section, the state-of-the-art deep learning architectures to perform the audio source separation task are presented. This section is divided in two categories, one corresponding to the spectrogram based architectures, and the other presents the waveform based deep learning architectures.

#### Spectrogram based Architectures

While some of the analytical methods presented in the previous section performs quite well regarding the audio source separation task, they have certain limitations. The main limitation is that the analytical methods are based on linear models. Along with this, the previously presented algorithms may perform very well for source separation over one song, but outperforms in a whole dataset, and it has to be re-optimized for every single song.

To take advantage of deep learning techniques in the source separation task becomes an attractive alternative to overcome these difficulties. This is because since supervised learning models, have a non-linear component, it allows the network to learn much more expressive transformations. On the other hand, instead of estimating the bases and activations for a single mixture, a model based on deep learning actually learns the mapping over a complete dataset [13].

Speech separation in the Time-Frequency domain can be achieved in several ways. The first one through the direct estimation of the spectrogram representation for each one of the isolated sources that compound the mix. This can be achieved by

using nonlinear regression techniques, where the clean isolated source spectrograms are used as the target of the training process. Another method is to estimate the mask of each one of the isolated sources and perform element-wise multiplication between the mixed source spectrogram and each one of the estimated masks in order to obtain all the isolated sources of the mix [14].

In this section the most important time-frequency domain deep learning architectures are presented.

**U-net** U-net is an architecture proposed by Ronneberg, Fischer and Brox on 2015. It estimates the independent isolated sources indirectly, by estimating a soft-mask for each of the sources. This soft-mask is then multiplied by the mixed magnitude element-wise spectrogram in order to obtain each one of the independent isolated sources.

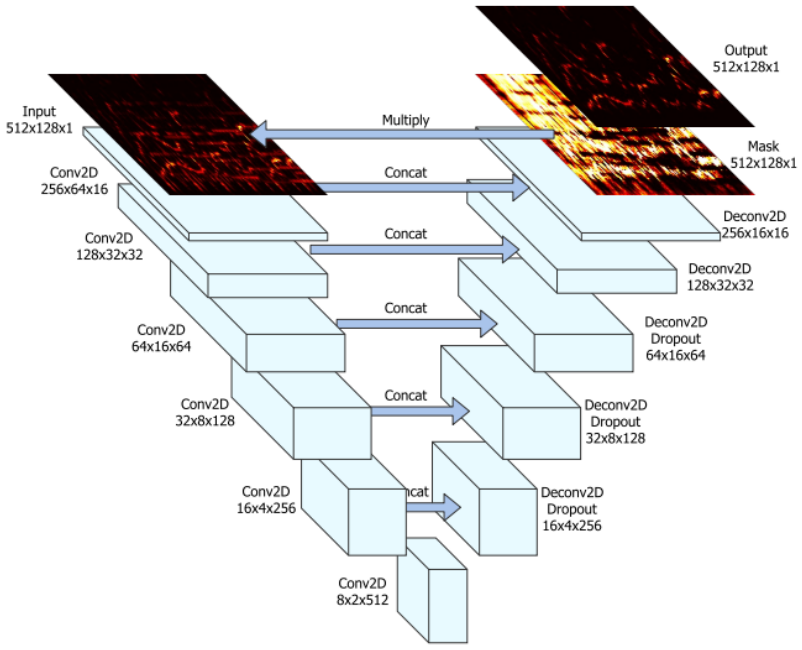


Figure 2: U-Net network architecture.

This network is based on a convolutional neural network, and the name its given following the "U" shape that the convolution and deconvolution encoding/decoding steps represents, as it can be seen on Figure 2. Each of the 2D convolutions reduces the dimensions of the spectrogram representation, but adds more autoencoders for

each step. Each deconvolution step takes as input not only the output of the previous layer, but also the level-wise output of the respective convolutional step. At the last layer, the soft-masks are estimated by backpropagation using a sigmoid activation function against the masks created by the target isolated sources.

Each encoder layer consists of a strided 2D convolution. Several publications has used a stride of 2 and kernel size 5x5, batch normalization, and LeakyReLU activation function with leakiness 0.2. In the decoder strided deconvolution are used (sometimes referred to as transposed convolution) with stride 2 and kernel size 5x5, batch normalization, plain ReLU, and use 0.5 dropout to the first three layers. The model is usually trained using the ADAM optimizer [15].

**Open-Unmix** Open-Unmix is a open source software source separation architecture based on the LSTM deep learning architecture. The inspiration to create this model arises from the need to provide to the open source software community a source separation model that could achieve state of the art results.

Open-Unmix, released in mid-2019, has as a second goal to serve as a source separation implementation for the community and artists. Because of this, the team released along with the paper an implementation of this architecture in Pytorch, one of the most used deep learning frameworks. The principles they sought to highlight in the implementation were MNIST-like.

MNIST is the most well-known dataset in the deep learning community. It consists of a set of approximately 70,000 handwritten numeric digits, along with their respective label. It is used as the main dataset for teaching classification tasks in supervised learning [16]. For many it is considered the 'hello world' of deep learning.

When Open-Unmix refers to the MNIST dataset, they do so to highlight the principles of its release to the open source community. The first of these principles is that it is simple to extend, through modular programming, which facilitates the replacement of system modules without further complication. Not-a-package is the next one of the principles, which refers to build the system in order to have self-contained portions of code, allowing easy change. The last of the MNIST-like principles is that

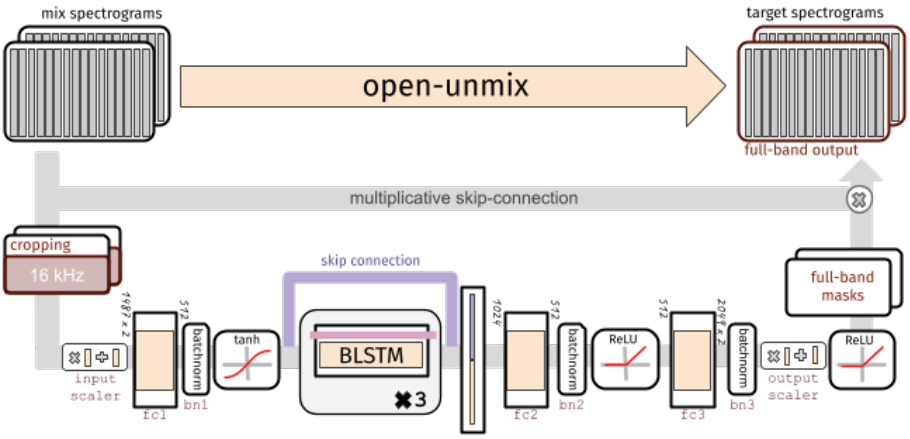


Figure 3: Open-Unmix architecture

the software should be hackable. This means that it must provide a complete setup through a clear pipeline, from downloading the data, through training, and finishing with the evaluation of the model on the test data [17].

Open-Unmix is based on a discriminative strategy, which means that the model directly learns a representation a mask of the isolated source. This is done by modifying its gradients according to the comparison between the estimated source and the ground truth isolated source.

The model operates on the spectrogram representations of the audio, which after going through a cutoff process of all frequencies above 16kHz and an input normalization, it is feed into a fully connected layer. This network has a batch normalization phase, which allows the different batches to have a more similar distribution between them. This point is crucial specially when dealing with audio signals, where dynamics are very important. The resultsing tensors are then feed into a *tanh* activation function, which avoids the exploding input and output in LSTM architectures.

The encoded signal is then passed through a three-layer bidirectional LSTM network. This allows the system to learn from audio segments of arbitrary length. However, since it learns from both the past and the future, it cannot be used for real-time applications. The skip connections through the BLSTM module allows a more accurate decoding process and a much faster convergence of the model.

The decoding process is based on two fully connected layers, which by means of batch

normalization and ReLU activation functions permits the mask reconstruction in the time-frequency domain. The estimated mask is then applied on the original cropped spectrogram to finally obtain an estimation of each of the original sources [18].

**Sams-Net** Sams-Net is a spectrogram-based source separation architecture. It is characterized by using a sliced-attention based neural network. This allows it to establish spectral feature interaction with a multi-head attention mechanism. Along with this, Sams-Net implementations are able to be run using parallel computing strategies because the attention mechanism has no temporal dependencies, unlike LSTM architectures.

Deep learning networks based on attention mechanisms allow the system to learn interactions between features directly from the data. This can be done without the need to introduce human based knowledge. The attention mechanism is a structure that allows capturing sequence distributions together with determining which parts of the sequence are more important than others.

The model is constituted by a structure of 3 layers. First, the audio is taken and the STFT is performed to compute the spectrogram. This is then passed through a transformation module that consists of a convolutional neural network that expands the channels of the feature map. The feature maps are feed to the attention module, which is repeated N times.

The attention module is compound by two parts: Multi-head attention and Depth-wise separable CNN. The multihead attention module takes the normalized feature map and passes it through three convolutional neural networks, and then applies scaled dot-product to each one of the three outputs corresponding to queries, keys and values of the attention based network. The outputs are then passed through a fourth convolutional neural network, which restores the dimensional space of the feature map.

The Depthwise separable CNN, in the other hand, allows to perform depth-wise and point-wise convolutions applied to the output of the previous layer and using skip connections. The architecture is shown in Figure 4.

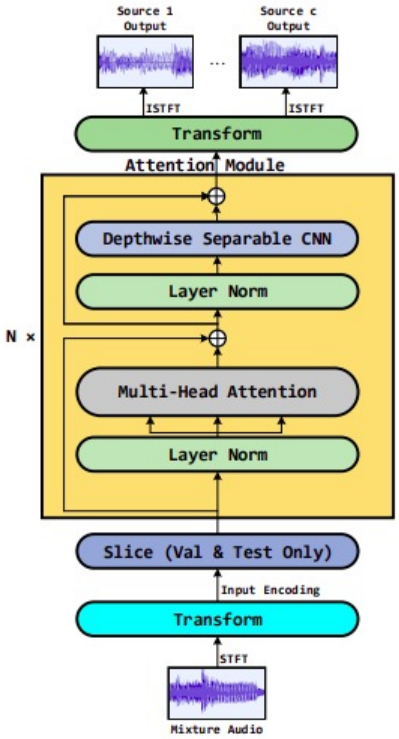


Figure 4: Sams-Net architecture

For the test and validation stages, the input is passed through a sliced-attention layer before the attention module operation. The spectrogram is divided into equal parts to feed the multi-head attention layer. The resulting vectors are concatenated into one single final vector. With the sliced operation, the scope of attention is narrowed down to the intra-chunk features. The reason for applying this operation is due to the nature of the songs, which often present repetition patterns like melody, structure, notes, chords and timbre. For the training stage only small randomly chosen chunks are applied.

Finally, the reconstruction of the isolated sources is given by the use of a second transformer. This is a transpose CNN layer that allows to add the dimensions of the feature map channels to recreate the masks in the time-frequency domain [19].

### Waveform based Architectures

The time-frequency or spectrogram based representation of the mixed signal, has been the most used approach in the source separation field. However, this method

has several disadvantages:

- STFT is a generic signal transformation that is not necessarily optimal for speech separation.
- The phase reconstruction from the estimated sources is a non-trivial problem, and the erroneous estimation of it introduces an upper bound on the accuracy of the reconstructed audio.
- Successful separation from the time-frequency representation requires a high-resolution frequency decomposition of the mixture signal, which requires a long temporal window for the calculation of STFT. This calculation could take from 32ms to 90ms, which makes it impossible to apply in real-time applications [14].

Those reasons motivates the research of waveform-based deep learning methods that could perform better than the time-frequency models. Some of the main source separation architectures are presented below.

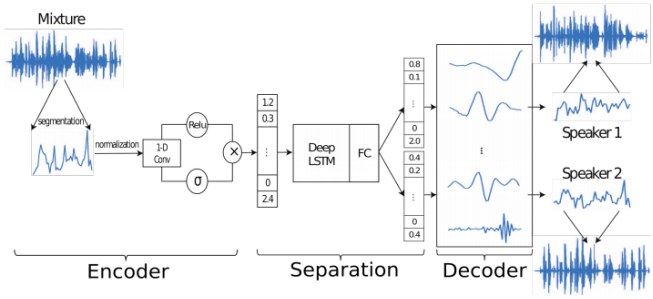


Figure 5: Time-domain Audio Separation Network (TasNet) models the signal in the time-domain using encoder-decoder framework, and perform the source separation on nonnegative encoder outputs. Separation is achieved by estimating source masks that are applied to mixture weights to reconstruct the sources. The source weights are then synthesized by the decoder. [14]

**TasNet** TasNet is a waveform convolutional Neural Network. It models the waveform using a convolutional autoencoder with a nonnegativity constraint. With TasNet, the phase estimation of the isolated sources is no longer a problem, since the

waveform is estimated sample by sample and no by decomposing the STFT into a separated representation of the magnitude spectrogram and phase. This architecture uses a linear deconvolution layer, that works as a decoder in the output to reconstruct the waveform as an inversion of the convolutional encoder [14].

**ConvTasNet** The ConvTasNet is an architecture introduced in 2019 by Yi Luo and Nima Mesgarani, the creators of the TasNet. ConvTasNet is a monaural source separation model designed primarily to separate voices and solve the cocktail party effect. It is an end-to-end architecture that works in the waveform domain, so it directly estimates isolated sources from masks.

The main difference between ConvTasNet and TasNet is that in its predecessor the mixture waveform was modeled by using of a convolutional encoder/decoder with a LSTM network, which consisted of an encoder with non-negativity constraints with respect to its output. At the same time, it used a linear decoder to reverse the encoded signal to reconstitute the waveform. However, the use of the LSTM network in the separation module severely limited the potential applications.

ConvTasNet uses exclusively convolutional layers in all the stages of the input signal processing. Thus, the ConvTasNet architecture consists of 3 fundamental layers:

- Convolutional Encoder: the mixed signal is divided into segments of the same size, which are then transformed to an N-dimensional representation using a 1-D convolution. C Vectors, one for each of the spearkers, are then estimated. These are the masks of each of the independent sources and are constrained in their non-negativity. The waveforms of each source are then reconstructed by the decoder. The constraint that the sumatory of all masks has to be equal to one is added in order to the model can learn that the architecture of encoders and decoders must perfectly reconstruct the mixed signal.

- Separation: This module is based on a fully-convolutional separation module, which is inspired by a Temporal Convolutional Network (TCN). The implementation of the TCN in the ConvTasNet consists of a stacked 1-D dilated

convolutional blocks with exponentially increasing dilatation factors. This is done in order to add large enough temporal context. The output of the TCN is passed then to a convolutional block.

- Decoder: In the decoder, a linear block is added as a bottleneck layer. This block determines the number of channels in the input and residual path of the subsequent convolutional blocks. Finally, the masked encoder features are used to estimated the isolated sources waveforms. This is made using a linear decoder [20].

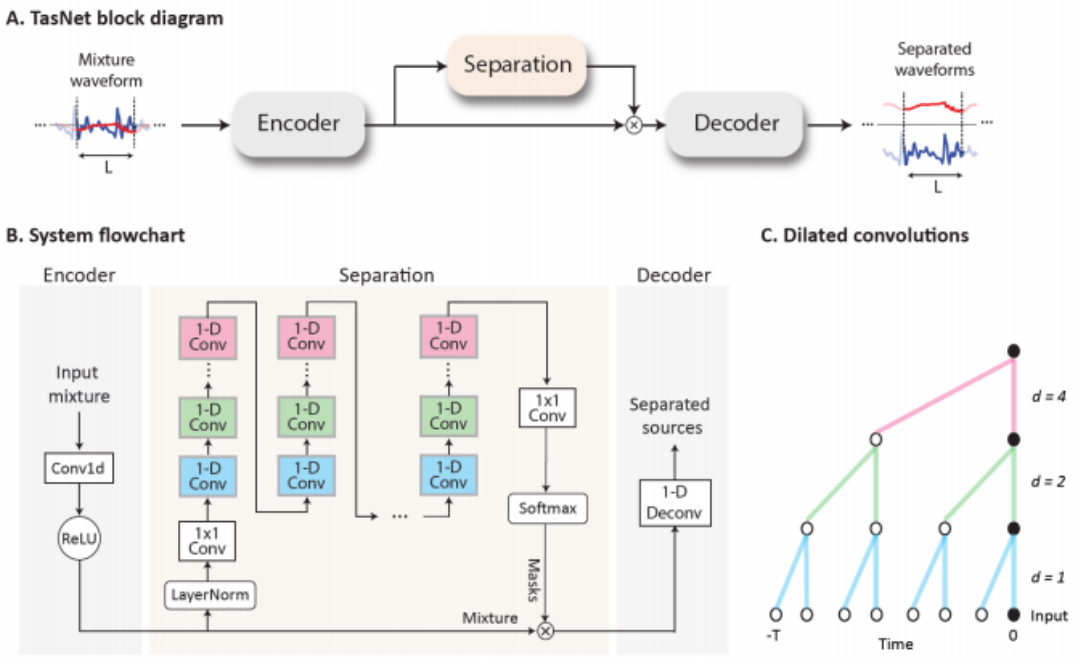


Figure 6: (A): the block diagram of the TasNet system. An encoder maps a segment of the mixture waveform to a high-dimensional representation and a separation module calculates a multiplicative function (i.e., a mask) for each of the target sources. A decoder reconstructs the source waveforms from the masked features. (B): A flowchart of the proposed system. A 1-D convolutional autoencoder models the waveforms and a dilated convolutional separation module estimates the masks based on the nonnegative encoder output. (C): An example of causal dilated convolution with three kernels of size 2. [20]

**DPTNet** The dual-path transformer network is an architecture proposed in 2020 by Jingjing Chen, Qirong Mao and Dong Liu. The presentation of this architecture surpasses the state of the art in voice separation models in monaural recordings.

This is done by incorporating an end-to-end architecture using an improved dual-path transformer to allow the system to learn from the audio context. This makes the model efficient in separating long audio segments.

This model is conceptually based on the ConvTasNet, as its structure follows the same 3-layer approach: encoder, separation layer and decoder. The encoder layer is used to convert the audio segments of the mix into a feature map, which is then input to the separation layer for mask creation. Finally the decoder layer reconstructs the wave signal of each of the isolated source estimations.

The encoder is mainly constituted by a series 1-D convolutions modules, which acts as a filter-bank W of N filters of length L. Thus, for a mixture of speakers $x \in \mathbb{R}^{1xT}$, it is subdivided into L-length overlapping vectors $R \in \mathbb{R}^{LxI}$, where I is the number of vectors. Finally the speech signal $x \in \mathbb{R}^{NxI}$:

$$x = ReLU(RW)$$

The separation layer, in turn, consists of three layers: segmentation, dual-path transformer processing and overlap-add. In the segmentation layer $X$ is segmented into overlapping chunks and then all the chunks are concatenated in a 3-D tensor. The dual-path transformer processing also consists of three layers:

- Scaled dot-product attention: effective self-attention mechanism for associating different positions of the input and calculating their representations.
- Multi-head attention: composed of multiple scaled dot-product attention modules.
- Position-wise feed-forward network: fully connected neural network with two linear transformations and a ReLU activation function between them.

In the third and finel stage of the separation layer, overlap-add, the output of the dual-path transformer layer is used to learn a mask for each of the sources, by using

a 2-D convolutional layer. Finally, a transposed convolution module is used in the decoder to reconstruct each signal separately [21].

**Wave-U-Net** The Wave-U-Net architecture was introduced in 2018 by Stoller, Ewert and Dixon. It is an end-to-end model for music source separation inspired by the U-Net. This network supports K-channel separation and operates completely in the waveform domain, which allows it to overcome the outcomes present on the time-frequency models, like the phase-discarding problem while doing STFT. This way, Wave-U-Net has the challenge of processing an input signal with much more data than spectrogram-based models. The Wave-U-Net architecture is compound by L levels. The input signal is passed through the L levels of downsampling to then go up through L levels of upsampling. Thus, the architecture is a model based on encoder - separator - decoder.

At each level of the downsampling phase, convolutions on the one-dimensional signal are executed by means of a number filters to be learned by the network. After the convolutions, the resulting feature map is passed through a $LeakyReLU$ activation function and a decimate layer in order to obtain on each one of the levels a feature map that is half the length of the previous level feature map.

In the upsampling phase, Wave-U-Net performs for each level a linear interpolation process that is learned by the model itself. The upsampling factor, like the U-Net, is performed with the intention of doubling the length of each feature map with respect to the previous level. After upsampling, the resulting feature map is concatenated with the output feature map of the equivalent level from the downsampling phase as it can be seen in Figure 7. This concatenated vector is then passed through a convolutional layer and a $LeakyReLU$ activation function. This allows the network to learn about the feature map of the equivalent level, which results in a faster convergence of the algorithm.

Finally, in the output layer, the concatenation and convolution process is repeated using the estimated filters, but a *tanh* activation function is used instead of the $LeakyReLU$. The principle that the final mixture $x(t)$ is equal to the sum of the

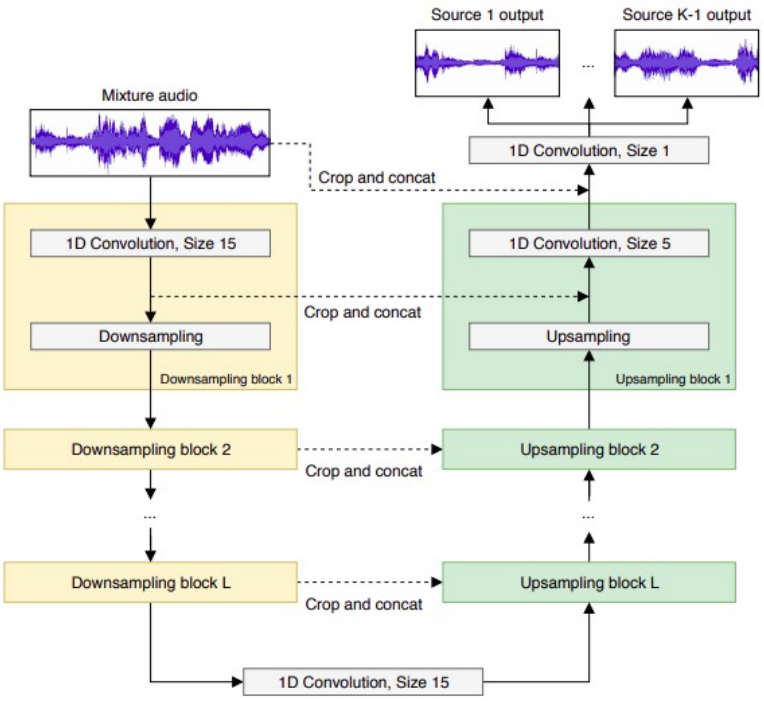


Figure 7: Wave-U-Net architecture [22]

respective isolated signals $x_i(t)$ is incorporated in the loss function:

$$x(t) = \sum_{i=1}^{N} x_i(t)$$

To introduce this constraint to the model only N-1 filters are applied in the last layer of the system, thus forcing the $n^{th}$ source to be calculated as:

$$x_i(t) = x - \sum_{i=1}^{N-1} x_i(t)$$

It should be noted that the authors of the Wave-U-Net propose an upsampling by means of interpolation learned by the network, instead of a strided transposed convolution, as proposed in the U-Net. This allows the network to avoid the generation of aliasing artifacts by signal reconstruction.

The Wave-U-Net presents the following main contributions to the state of the art. First, it proposes a new model that allows separation directly in the time domain and

that can take temporal contexts as input. Along with this, a method is presented to avoid the generation of aliasing artifacts at the edges of the output windows. The model proposes a type of upsampling learned by the architecture, to avoid the oversimplifications of a linear interpolation and the artifacts introduced by the strided transposed convolutions [22].

## 2.2 Datasets

In the context of the source separation tasks, 5 main tasks can be distinguished. These tasks correspond to the following:

- Music Source Separation: It corresponds to the task of separating one or several specific instruments having as the only starting point the final mix of the song. In this task we usually aim to separate Vocals, Drums, Bass and Other, the latter being the residual mix of all the other instruments, sounds and effects that are not contained in the Vocals, Drums and Bass.

- Speech Source Separation: It corresponds to the task of separating two or more distinct speakers from a single audio file containing the mixture of these speakers. This task aims to solve the cocktail party effect problem, since what it seeks is to clearly separate speakers in a context of more than one conversation occurring simultaneously. There are some datasets that seek to make the problem more complex by adding sound effects such as reverberation, audio degradation, background noises, among others.

- Speech Enhancement: The purpose of this source separation task is, as its name suggests, to enhance the quality of an audio corresponding to a speaker. In general, datasets consisting of a single speaker plus background noise or reverberation are used. This problem is especially relevant for audio companies with products dedicated to non-professional users, since it allows to improve the quality of an audio using as a starting point audio recorded in non-treated spaces and with non-professional equipment.

- Environmental Sound (Universal) Source Separation: Corresponds to the task of separating universal sounds. Unlike the previous tasks that seek to isolate or enhance a specific element of the mix; speech or specific instrument, this task seeks the separation of arbitrary sounds [23]. This could be, for example, isolate the sounds of different birds in a recording context in the middle of a forest.

- Audio-visual Source Separation: It corresponds to the task of separating the audio of speakers using as a starting point not only the audio mix between speakers, but also using a video file and the corresponding annotations indicating which speaker is speaking in which moment of the video and for how long. The video serves as complementary information for training the model.

For all the tasks described above, it is necessary to have a significant and sufficiently large dataset to train models and methods successfully. Some of the most commonly used datasets for each of the source separation tasks described above are presented below.

**Music Source Separation Datasets**

- MUSDB18 [24]: This dataset consists of 144 professionally produced songs made available to the community through Creative Commons. It was created from the audio stems made available by the Cambridge Music Technology Institute. The authors generated linear mixes between the different instruments and grouped them into 4 categories: Vocals, Drums, Bass and Others. The files are available in .stem format, which allows quick access to each of the elements and to the final mix.

- DAMP-VSEP dataset [25]: This dataset consists of different music where the tracks are separated in two: backing track and singer. The final mix of both tracks is also provided. The dataset was made available by SMule, a mobile karaoke application where users can sing and record themselves using their mobile phone.

**Speech Source Separation**

- wsj0-2mix dataset [26]: This is a single channel speech separation dataset based on the WSJ0 dataset [27], which consists of a read speech corpus of news texts from the Wall Street Journal. The wsj0-2mix dataset took this corpus and created variations to achieve approximately 40 hours of audio. The variations consist of gain-weighted mixing between 0db and 10db SNR of segments from the WSJ0 dataset. This way a dataset of over 40 hours of spoken audio was compiled.

- WSJ0 Hipster Ambient Mixtures (WHAM) dataset [28]: This dataset is based on the wsj0-2mix dataset but it adds a unique noise background scene to each of the two-speaker mixes. The background noises were recorded at various locations in the city of San Francisco in late 2018 using an Apogee Sennheiser binaural microphone.

- WHAMR dataset [29]: This dataset is based on WHAM, but in this release ithe authors added reverberation effects to the speakers with added noise. The impulse responses used for reverberation come from the pyroomacoustics package. Thus, a more complete dataset is released for speech separation tasks in noisy contexts.

- LibriMix dataset [30]: It is a dataset consisting of a mix between the LibriSpeech dataset and the noises used by the WHAM dataset. It is an open source dataset and represents a free alternative to WHAM. LibriMix in its extended bandwidth version (16kHz) is about 450Gb, so a reduced version of the dataset called MiniLibriMix with only 800 speakers was also released. [31].

- Kinect-WSJ dataset [32]: This is a version of WSJ0-2MIX with added reverb and noise. The Microsoft Kinect device was used and microphones arranged in a linear array in order to use real ambient noise, like the ones used in the CHiME Challenges. The complete dataset its about 230Gb corresponding to almost 100,000 samples.

- SMS_WSJ dataset [33]: This dataset is based on the WSJ10 dataset, but adds the spatial dimension by extending multichannel speakers in space. The metadata of this dataset includes the location of each speaker in a two-dimensional space, which allows a model to learn to separate speakers using the multichannel mixed audio source.

- VCTK dataset [34]: The VCTK dataset is a set of recordings of English speakers. The performance of 110 English speakers with their respective accents from various English-speaking regions was recorded. They were recorded reciting different phrases of an average length of about 3 seconds. The performances were recorded with two different microphones, the Sennheiser MKH 800 and the DPA 4035 and the recording was carried out in a semi-anechoic chamber located at the University of Edinburgh. The recordings were originally made at 96kHz, 24 bits, and then transformed to 48kHz, 16 bits.

**Speech Enhancement**

- DNS Challenge's dataset [35]: This dataset was released by Microsoft for the 2020 Deep Noise Suppression Challenge. It consists of isolated speech audios, room impulse responses and noise for the creation of the dataset. The speech was obtained from other Speech Source Separation datasets such as LibriMiniMix, VCTK, among others. The noises were obtained from FreeSound.org and the Simulated Room Impulse Response responses correspond to the Simulated Room Impulse Response Database 26 and 28. The base dataset was provided by [36].

**Environmental Sound (Universal) Source Separation**

- Free Universal Sound Separation (FUSS) [37]: This dataset consists of the arbitrary mix of sounds and their respective gain level annotations for each datapoint. It was the official dataset used in the DCASE2020 Challenge Task 4: Sound Event Detection and Separation in Domestic Environments. The

audios were mainly obtained from the FSD50k dataset, consisting of approximately 50,000 audios uploaded to Freesound.org and with their respective metadata manually annotated [38]. The mixes consist of 10-second excerpts of 1 to 4 distinct audios, convoluted with simulated room impulse responses and then added together.

**Audio-visual Source Separation**

- AVSpeech [39]: This dataset was compiled and made available by Google for Audio-visual source separation tasks. It consists of thousands of hours of video and audio segments available on the web. The dataset contains the respective annotations to use image-tracking models in a complementary way to the speech source separation task.

## 2.3 Evaluation

Since in the source separation task there are more outputs (estimated isolated sources) than inputs (original mix), this task is considered to be an undetermined problem. Source separation aims to solve the cocktail party effect, which is a perceptual phenomenon and for this reason, the best way to evaluate the performance of a specific architecture would be to have a group of experts, such as musicians, audio experts and musicologists, and having them to listen and annotate the estimated isolated sources in a treated room and then compare them with the ground truth sources. This technique is called MUSHRA test and it provides the best results in terms of measuring the accuracy of the models. However it mains disadvantages are are that it take a lot of time to perform, and it is are also expensive, if the listeners are not volunteers.

To overcome the disadvantages, there is an alternative way to measure the effectiveness of the source separation system, which consists of computing some statistics between the estimated sources and the ground truth isolated sources. The most used statistics are Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR),

and Source-to-Artifact Ratio (SAR), and they have become the standard metrics to evaluate and compare the effectiveness of one model over another.

Given an estimated source $s_i$, it can be described as the sum of three elements:

$$s_i = s_{target} + e_{interf} + e_{noise} + e_{artif}$$

where $s_{target}$ is the ground truth isolated source, and $e_{interf}$, $e_{noise}$, and $e_{artif}$ are error terms for interference, noise, and added artifacts, respectively. With these elements the different statistics can be computed [40]:

- Source-to-Artifact Ratio is interpreted as the amount of unwanted artifacts an estimated source has with relation to the ground true isolated source. It is computed as:
$$SAR = 10 \log_{10} \frac{|s_{target} + e_{interf} + e_{noise}|^2}{|e_{artif}|^2}$$
- Source-to-Interference Ratio is interpreted as the amount of other sources that can be heard in a particular estimated source. It is computed as:
$$SIR = 10 \log_{10} \frac{|s_{target}|^2}{|e_{interf}|^2}$$
- Source-to-Distortion Ratio is an overall measure of how good an estimated source sounds. It is usually the most important statistic reported in papers.
$$SDR = 10 \log_{10} \frac{|s_{target}|^2}{|e_{interf} + e_{noise} + e_{artif}|^2}$$

$$SNR = 10 \log_{10} \frac{|s_{target} + e_{interf}|^2}{|e_{noise}|^2}$$

## 2.4 Source Separation Toolboxes

There are a few open-source projects in the source separation community which allow researchers and developers to experiment with the most important source separation

architectures in a fast and reproducible manner. These tools are maintained by their own community and right now they represent one of the best ways to perform fast benchmarking between architectures over the most commonly used datasets.

### 2.4.1 Asteroid

Another important Source Separation toolbox is Asteroid, a PyTorch-lightning library that, as Nussl, enables fast experimentation of source separation architectures over common and also custom datasets to reproduce the most relevant papers in the source separation field. Asteroid provides the most relevant architectures as PyTorch-lightning models and a few others as a recipe. A recipe in Asteroid is a set of scripts to build a source separation system with a directory structure template that makes it easy for researchers and developers to build, train, evaluate and contribute their own architecture implementations. Asteroid also provides a publication tool which allows researchers to publish and download pretrained models through Zenodo. The supported architectures of Asteroid are ConvTasnet, Tasnet, Deep clustering, Chimera ++, DualPathRNN, Two step learning, SudoRMRFNet, DPTNet, DCCRNet, DCUNet and they have recently added FastNet. The supported datasets are WSJ0-2mix, WSJ03mix, WHAM, WHAMR, LibriMix, Microsoft DNS Challenge, $SMS\_WSJ$, MUSDB18, FUSS, AVSpeech and the Kinect-WSJ [41].

# Chapter 3

# Dataset

This section describes the methodology used for the creation of the dataset to perform the source separation task between foreground speech and background music in the context of podcasts and radio shows. The methodology used corresponds to the experimental method, since existing models and architectures were used to obtain benchmark results on this new dataset. These results are then compared in the Results chapter with the metrics of these same models trained on the most known datasets for the source separation task.

This section is organized as follows. First, the steps performed to obtain the dataset and the reasons to chose the sources for speech and music are described. Next, the pipeline used to evaluate these models on the reduced dataset is presented. Finally, the models chosen to be trained on the dataset and the modifications made to these models are described.

## 3.1 Dataset compilation

In the first instance, in order to get a better idea of the most standard format in which datasets are presented, several datasets were reviewed in the context of source separation. For this purpose, I used the Asteroid toolkit, which includes a series of source separation models already implemented, ready to be trained and

evaluated. Furthermore, Asteroid includes implementations of the dataloaders of the most commonly used datasets in the context of source separation.

As we try to present data as similar as possible to the context of Podcasts, the main characteristic of this dataset is that it must be constituted by background music and foreground speech. In order to create the dataset, different Asteroid supported datasets were analyzed, however, none of the available datasets had the most important speech and music conditions to generate a synthetic Podcast-like dataset.

The key conditions for considering both music and speech sources in this synthetic dataset are, on the one hand, to have high quality recordings. This means having audios with a sampling frequency of at least 44.1 kHz. On the other hand, a wide variety of songs licensed under creative commons is required, so that the dataset can then be released for use by the rest of the scientific community. Finally, in the case of music, it is important that the songs used are as similar as possible to those used in podcasts. This means that not only the quality of the mix must be excellent, but it must also represent musical styles used in real Podcasts.

Among the datasets available in Asteroid, the few that contain exclusively music, MUSDB18 and DAMP-VSEP, neither has a wide range of songs and styles at the same time, and no high-definition audio is available. In the case of MUSDB18, only 144 songs are available. Although previous research has adopted an audio augmentation strategy, using this dataset and coherent as well as incoherent mixtures of the individual stems, it was felt that having a dataset with realistic music was more important for the training process. On the other hand, DAMP-VSEP, which contains a much larger dataset of music, has the disadvantage that the recordings were made directly with the cell phones of the Smule app users, so in order to have songs with lyrics, a mix should be made between the backing track and the sung voice, both having very different qualities. At the same time, DAMP-VSEP only makes its tracks available in compressed format (m4a).

Considering the speech datasets made available by Asteroid are wsj0-2mix, WHAM,

WHAMR and Librimix. However, none of the above provides the audios in a resolution higher than 16kHz, and since these speechs are considered to be mixed with music, and thus obtain the synthetic podcast dataset, they were discarded.

After discarding the music and speech datasets available from Asteroid, various other music and speech datasets were investigated. Finally, two datasets were chosen, one for background music and one for foreground speech. These datasets had to comply with the characteristics of having both speech and music in high quality, not only in terms of sampling rate, but also in terms of recording conditions, in the case of speech; and musical quality, in the case of the music dataset. For these reasons two datasets were chosen: On one side, the well-known VCTK [34] speech dataset, and on the other, a set of the most popular songs from Jamendo [42], a creative commons music streaming application.

The VCTK dataset is a set of aproximately 88.000 recordings of English speakers. The performance of 110 English speakers with their respective accents from various English-speaking regions was recorded. They were recorded reciting different phrases of an average length of about 3 seconds. The performances were recorded with two different microphones, the Sennheiser MKH 800 and the DPA 4035 and the recording was carried out in a semi-anechoic chamber located at the University of Edinburgh. The recordings were originally made at 96kHz, 24 bits, and then transformed to 48kHz, 16 bits. The recordings were downsampled to 44.1kHz and saved in FLAC format. The reason for this re-sampling process is to have the same sampling frequency for both voice and music as inputs of the models to be trained.

Regarding the music dataset, a set of songs considered the most popular ones uploaded to Jamendo were used. The reason for wanting to obtain the most popular songs was to be able to have music as similar as possible to music with commercial licences, which is frequently used in radio programs and podcasts. In this sense, the music dataset had to be high quality in both technical and musical terms.

To obtain the most popular songs, the Jamendo API was queried by the popularity_total sort option. These option, according to the documentation, allows to

| | **Train** | **Validation** | **Test** |
|---|---|---|---|
| Speech | 70662 | 8833 | 8833 |
| Music | 15496 | 1937 | 1937 |

Table 1: Number of audio files in each one of the subsets of the PodcastMix.

| | **Train** | **Validation** | **Test** |
|---|---|---|---|
| Mean | 22.67 | 22.708 | 22.649 |
| SD | 2.946 | 2.948 | 2.899 |

Table 2: Speakers age distribution

obtain the most popular songs of all time. The concept of popularity for Jamendo considers various factors such as the number of downloads, plays, likes, among many others.

Using the Jamendo API, the metadata of the songs was saved in a json dictionary, including licenses, upload date, artist, tags, album and URL for direct download in the selected quality. Using this information, a set of the 17432 most popular Jamendo songs was downloaded.

The dataset was then divided into 3 subsets: train, validation and test. The proportions used to divide the dataset were 80% for train, 10% for validation and 10% for test. In this way, the respective metadata files for both speech and music were generated separately, keeping the references to the files to be used as loading dataframes within the dataloader. All the metadata available in Jamendo regarding music, its licenses, authors, album, year, duration, genres and tags was also preserved. On the VCTK side we also kept the metadata: speaker's gender, age and english accent.

The exact number of tracks for each subset of the dataset can be apreciated in Table 1. In the other hand, it can be seen in Table 2 that the distribution of speakers in terms of age is quite homogeneous for each of the subsets. However, it can be apreciated in Figure 8 that the dataset contains more female speakers than male, but the distribution across the different subsets is equally distributed.

Regarding the music dataset, is interesting to note that most of the tags in the music are refering to the voice or vocal tag, as it can be apreciated in Figure 9. This means that most of the music in the dataset is not instrumental, which probably will add

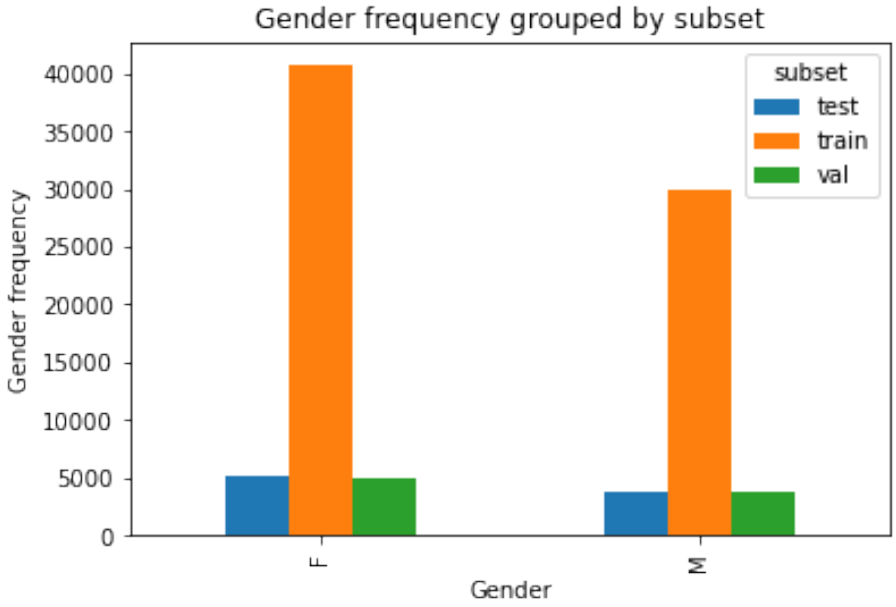


Figure 8: Gender frequency grouped by subset

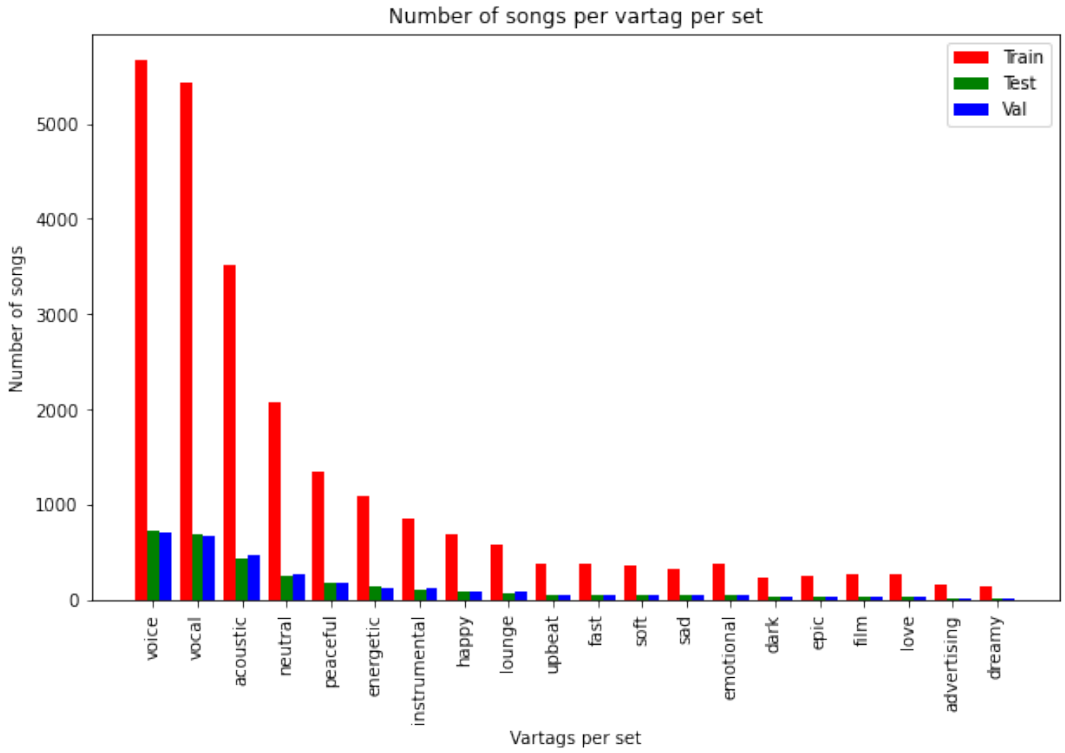


Figure 9: Songs per vartag per subset

more difficulty to the source separation task between the backgound music and the speech, since there will be human voices in both sources.

Other relevant statistics from the dataset can be found in the Appendix 1.

# Chapter 4

# Methods

This section systematizes the background music and foreground speech source separation task. Thus, this section describes mathematically the problem to be solved and its respective evaluation. Some specific characteristics of mixtures in podcast contexts are assumed and the methods used to evaluate the correct separation between sources are defined.

This chapter is organized as follows. First, the background music and foreground speech source separation task is mathematically defined. Next, the most important metrics used to evaluate the performance of the models trained for this task are listed. Finally, the details of the experiments performed are listed, together with the architectures chosen to train and evaluate the dataset.

## 4.1 Background Music and Foreground Speech Source Separation

As detailed in the State of the Art chapter, the Source Separation task can be defined as obtaining the isolated sources that make up a mix, using exclusively the final mix as input. In this sense, a mix $x(t)$ can be defined as:

$$x(t) = \sum_{i=1}^{N} g_i x_i(t)$$

With N number of sources, $g_i$ a gain weighting of the $i^{th}$ source and $x_i(t)$ the $i^{th}$ source.

In the case of the task of separating background music and foreground speech, the number of sources is 2. On the other hand, it can be observed that in podcasts if there is music and speech playing at the same time, the loudness level of the music is never higher than the level of the speech. Thus, the problem can be written as follows:

$$x(t) = g_{speech} x_{speech}(t) + g_{music} x_{music}(t), \qquad g_{music} \leq g_{speech} \tag{4.1}$$

Assuming that both sources $x_{speech}$ and $x_{music}$ may not be normalized, the $g_{music}$ and $g_{speech}$ factors are required to ensure that the loudness level of the voice is greater or equal than the loudness of the music. For this purpose the RMS metric was used as loudness, which is defined as:

$$RMS(x) = \sqrt{\sum_{t=0}^{T} (x(t))^2}$$

With T equals to the length of the audio segment. If we would like to have both sources sounding with equivalent level of loudness, we can assume that the speech level is the reference one, and the music loudness must match the speech one.

$$RMS(g_{music} x_{music}) = RMS(x_{speech}) \tag{4.2}$$

$g_{music}$ is a positive constant between the range [0, 1]:

$$\begin{aligned} g_{music} RMS(x_{music}) &= RMS(x_{speech}) \\ g_{music} &= \frac{RMS(x_{speech})}{RMS(x_{music})} \end{aligned} \tag{4.3}$$

In order to have a mix where the loudness of the music is always lower or equivalent to the loudness of the speech, $x(t)$ should be defined as:

$$x_{mix}(t) = x_{speech}(t) + f_{reduction} \frac{RMS(x_{speech}(t))}{RMS(x_{music}(t))} x_{music}(t)$$

Where $f_{reduction}$ is a random reduction factor that belongs to the range of [0, 1].

Thus, it is assumed that any linear mix between a speech file and music that follows the equation presented above will have similar characteristics to those of both amateur and professionally created podcasts. The fact that the voice is always above the loudness of the music is a crucial factor of this type of narrative format.

Taking as a starting point the dataset described in the previous chapter and the mathematical formalization of the background music and foregound speech source separation task, it is possible to create a virtually infinite synthetic dataset, consisting of random mixtures between the speech and music subsets. These mixtures also take in account an RMS normalization to match the loudness levels, and then a reduction factor between 0 and 1 to lower the loudness level of the music. This must be done to guarantee that these synthetic mixes are as close as possible to real podcasts. Using this virtually infinite dataset, it is possible to train source separation models and evaluate their performance.

## 4.2 Evaluation

For the evaluation of this background music and source separation task, the most important metrics used in the speech enhancement, music source separation and multispeech source separation tasks were considered. These metrics are Source-to-

Distortion Ratio (SDR), Source-to-Interference Ratio (SIR), and Source-to-Artifact Ratio (SAR).

The reason for using these metrics is that this source separation task, as well as those mentioned above, are based on the reconstruction of not only one, but both sources. In this sense, the goal of this task is not just to isolate the speech, as it could be thought for on-demand remix contexts; or exclusively separate the music, for fingerprinting recognition contexts. Rather, it seeks to reconstruct both sources as accurately as possible. Thus, the metrics contained in the BSSEval package, including 'sdr', 'sar' and 'sir' are suitable for this purpose, and represents a set of objective metrics with easily compare the accuracy of different models.

An important point that should be mentioned is that because this dataset is made up of synthetically created podcasts, it cannot necessarily be assumed that a model with a high test set accuracy will perform well in separating the actual podcast sources. The evaluation subset, defined as 10% of both speech and music data, with an on-the-fly mixing strategy for generating synthetic podcasts has the bias that both, the test set and the training set, were created under the same strategy.

For this reason, a second evaluation set is proposed, consisting of 20 real podcasts with their respective isolated sources. These podcasts were obtained from different sources and provided directly by the authors, who were given the job of rendering their podcast projects from their respective DAWs, to obtain a file for the voice or voices, another one for the music and a final one with the final linear mix.

## 4.3 Experiments

For the experiments, the two most important architectures were selected to perform a benchmark. These architectures are ConvTasNet, as a waveform-based models, and UNet, which is an architecture that operates in the time-frequency domain.

The reasons for choosing these architectures are based on the fact that they are still the two most commonly used architectures as a point of comparison when evaluating new source separation models. Although ConvTasNet and UNet do not represent

| Parameter | Value |
|---|---|
| Epochs | 100 |
| Multi_speakers | yes |
| N of filters | 1024 |
| Stride | 4 |
| Kernel size | 16 |
| N of blocks | 6 |
| N of repeats | 3 |
| N of channels after bottleneck | 512 |
| N of channels in the skip connections | 512 |
| N of channels in convolutional blocks | 512 |
| Optimizer | Adam |
| Learning rate | 0.0001 |
| Sample rate | 44100 |
| segment | 2 |

Table 3: ConvTasNet training parameters.

the best performing source separation models, when trained over different datasets and to perform other tasks, they are still some of the most important architectures to compare with, and every time a new model or dataset emerges is compared with at least these two architectures.

The ConvTasNet implementation chosen to be trained and evaluated over the PodcastMix is a model developed by the Asteroid team. This model is based on the paper presented at TASLP 2019 by Yi Luo and Nima Mesgarani [20]. The details of the implementation can be seen in Table 3:

It can be appreciated that the main differences between [20] and the implementation by the Asteroid Team are the sample rate and the length of the segments. The reason for using 44.1kHz instead of the 8Khz sample rate used in the original paper, was to be able to estimate high quality sources to allow the model to be used with professional purposes. However, one of the consequences of this decision was that the model became very large in terms of memory used, and the machine used to train only allowed 2 seconds excerpts of audio due to memory restrictions (32Gb).

The implementation of the UNet was developed from scratch in pytorch based on the UNet implementation presented in [15]. This consists on 6 convolutional-deconvolutional layers with concatenation between the layers of the same level as

| Parameter | Value |
|---|---|
| Epochs | 100 |
| Multi_speakers | yes |
| Stride | 2 |
| Kernel size | 5 |
| Optimizer | Adam |
| Learning rate | 0.0001 |
| FFT size | 2048 |
| Window size | 2048 |
| Hop size | 441 |
| Sample rate | 44100 |
| segment | 2 |

Table 4: UNet training parameters.

the separator block. As encoder and decoder the torch implementation of the Short Time Fourier Transform and its respective inverse function were used directly.

Like in [15], a 50% dropout for the first 3 layers, ReLU activation function for each convolutional and deconvolutional layer and a Softmax activation function for the last layer were implemented.

As it can be appreciated in Table 4, the main differences between the current implementation of the UNet and the original one proposed by [15] are the parameters of the STFT. Since the ConvTasNet had the limitation that it could only be trained using 2 seconds excerpts, in order to be able to compare the results between both architectures, the UNet was also trained with the same amount of samples. This two main changes, training segment and sample rate, forced the network to change the STFT parameters in order to preserve the size of the kernel, stride and number of convolution/deconvolutional blocks.

Both ConvTasNet and UNet were trained during 100 epochs, using a LogL2 loss in the time domain function.

$$LOGL2_{time} = \frac{10}{TK} \sum_{k} \log_{10} \sum_{t} |\hat{y}_{t,k} - y_{t,k}|^2$$

Where $T$ are the number of samples of the signals and $K$ are the number of sources to be separated.

Also, both models implemented a normalization initial layer which can be described as follows.

$$x^{norm}(t) = \frac{x(t) - mean(x(t))}{std(x(t))}$$

Because of this, to compute the loss function and to evaluate the network the sources and estimated sources needed also to be normalized:

$$x_i^{norm} = \frac{x_i - mean(x(t))}{std(x(t))}$$

With this modification the networks learns to estimate a normalized representation of the sources. Thus, in order to obtain an estimated source within normal audio representations, the inverse process has to be performed:

$$\hat{y}(t) = \hat{y}(t)^{norm} std(x(t)) + mean(x(t))$$

Where $\hat{y}(t)^{norm}$ is the output of the model, which is normalized since the network was trained with normalized inputs and the estimations compared against the normalized sources.

Both ConvTasNet and UNet calculate their loss function in the waveform domain, so UNet, despite being trained to learn softmask representations, passes the resulting spectrograms through the decoder back to audio representations and evaluate the quality of the separation in the waveform domain. Both networks were trained with 2-second segments of synthetic podcast audio, with a sample rate of 44.1 kHz.

The Dataloader script used to load the data and train both architectures considered an on-the-fly mixing of the speech and music files in order to create the synthetic podcast. The criteria used for the creation of the synthetic podcasts by the dataloader are the following:

- For each new epoch, both music and speech files are randomized.

- For each item, the current music index is taken and a non-silent random 2 seconds segment is obtained.
- A speech buffer consisting of random files from the same speaker is created with a length of at least 2 seconds.
- If multi-speaker is used, a single audio file from a different speaker is overlapped in a random position of the buffer, to emulate speakers interruptions once every 10 items.
- The speech buffer is shifted in a random position to prevent always getting buffers starting at the beginning of a speech.
- The music and the speech buffers are normalized in terms of RMS, in order to have similar loudness.
- The music is weighted by a random factor $f_{reduction}$ with an uniform distribution between 0.01 and 1 to replicate podcast contexts where the music is always quieter than the speech.
- Both audio segments are overlapped to obtain the synthetic podcast.

# Chapter 5

# Results and discussion

Once the training sessions were completed, the models were evaluated. Two test sets were used for this purpose, one corresponding to the PodcastMix test partition, synthetically elaborated using the same methodology used in the training set. On the other hand, a test set consisting of 20 segments of 20 seconds length of real podcasts was used. These podcasts were provided with their respective isolated reference sources for evaluation.

Using these evaluation sets, the results obtained can be appreciated in Table 5. It can be seen that ConvTasNet gets a higher SDR score comparing with the UNet evaluated over the synthetic test set. Specifically, a SDR value of 13.736 was obtained for the ConvTasNet and 11.621 for UNet. However, the UNet seems to outperform the ConvTasNet in real podcasts scenarios with a SDR of 7.382 against -2.008 for the UNet and the ConvTasNet respectively. These results can be explained by the following reasons.

| | ConvTasNet Synthetic Dataset | UNet Synthetic Dataset | ConvTasNet Real Podcasts | UNet Real Podcasts |
|---|---|---|---|---|
| sdr | 13.736 | 11.621 | -2.008 | 7.382 |
| sir | 21.944 | 19.990 | 3.953 | 17.132 |
| sar | 14.753 | 12.648 | 8.020 | 8.661 |

Table 5: Evaluation results over Synthetic and real podcasts.

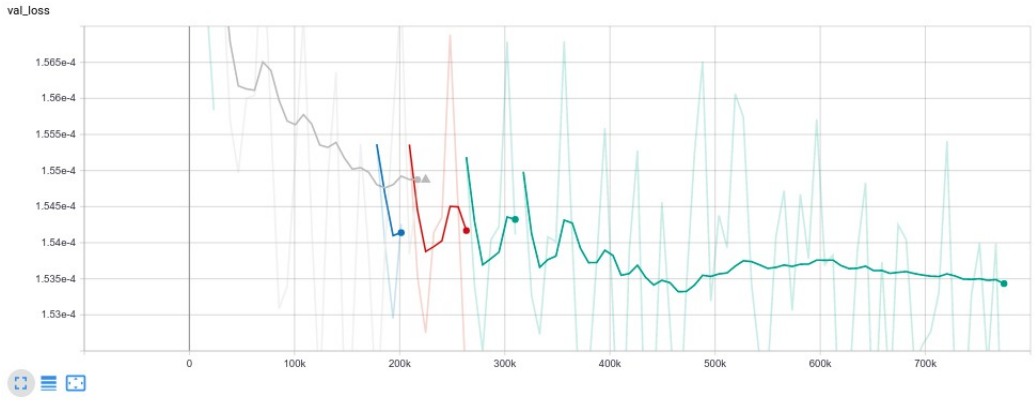


Figure 10: Training loss function of the UNet model

First, the models operate in different domains, being ConvTasNet a model that performs its separation in the waveform domain. On the other hand, the UNet operates in the time-frequency domain. This is a fundamental difference, since being able to encode the signals into a high-resolution spectrogram allows the UNet to learn timbre factors, which ConvTasNet cannot learn direclty from the waveform.

Second, it has been previously discussed [23] that STFT models such as the UNet outperforms learnable bases models like the ConvTasNet. This means that the ConvTasNet tends to work very well in-domain, but when extending the problem to other domains such a collection of real podcasts it presents problems for separating the sources.

In addition to the previous point, it is worth mentioning that the speech audios used to train the models belong to the VCTK dataset. As mentioned in the Datasets section, the speech that compounds the VCTK were recorded using two omnidirectional microphones, the Sennheiser MKH 800 and the DPA 4035. In podcast contexts it is more common to find recordings using microphones with directional patterns in order to isolate better the sound of the speakers and to obtain a better quality sound. Microphones with directional patterns, unlike omnidirectional microphones, record pressure gradients and are subject to the proximity effect, which accentuates the low frequencies of the voice. These two factors may have influenced the poor generalization of ConvTasNet in contrast to UNet.

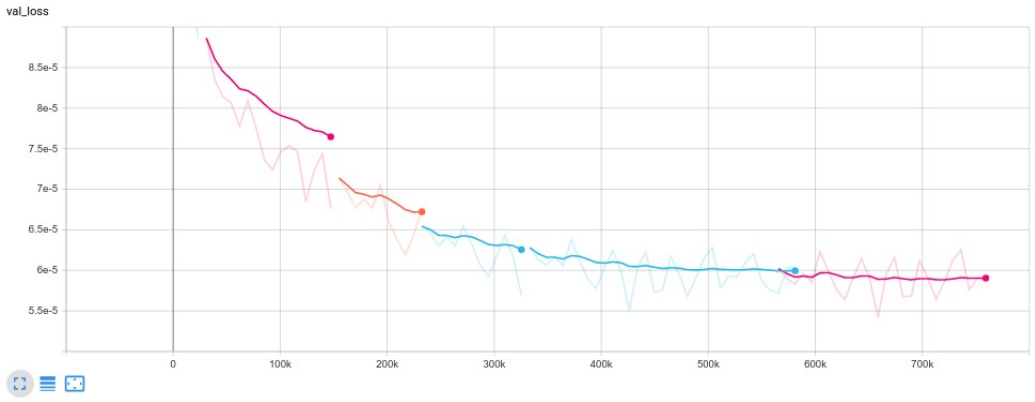


Figure 11: Training loss function of the ConvTasNet model

In figures 11 and 10 the loss curves of both models during training are shown. From these it can be appreciated that the descendent gradient curves are not following a stochastic behavior, but that indeed the models are converging by learning how to minimize the loss function.

In addition to these two evaluations, a third set of real podcasts was compiled. These podcasts are intended to be used for subjective evaluation of source separation, so the isolated references were not needed. The compiled dataset consists of 40 excerpts from real podcasts of 20 seconds duration. Each one of the podcasts in this dataset must meet the following 4 properties.

1. Be recorded high quality. At least 44.1kHz with no high cut filter.
2. Have at least 20 seconds of voice and music.
3. The music and content of the speech is Creative Commons.
4. The music is not included in the training set.

The complete dataset, including the training, validation and test set, along with the test set of real podcasts, plus the set of podcasts without isolated references was uploaded to the Zenodo platform. Along with the sets, samples of the separations performed by both the ConvTasNet and the UNet models were uploaded for the 3 evaluation sets.

# Chapter 6

# Conclusion

Throughout this document, the work carried out in the context of the master's thesis has been presented. In it, we have formalized the problem of background music and foreground speech source separation in the field of Music Information Retrieval. A review of the literature and methods considered state of the art in the field of source separation has been made and a synthetic dataset called PodcastMix, for the training and evaluation of this task has been presented. The dataset is built from the VCTK dataset and a selection of the most popular music from the creative commons streaming platform Jamendo.

Along with this, two of the most emblematic and cited deep learning source separation models have been selected to be trained using the PodcastMix. The selected models were the UNet and the ConvTasNet. For the first one, a version of the UNet was implemented from scratch using pytorch-lightning and based on [15]. On the other hand, the ConvTasNet implementation used was a version based on [20], implemented by the Asteroid team and released as an open source project in the library of the same name.

Both models included an audio normalization step in their encoders and denormalization at the end of the decoders. This allowed the models to learn representations of the mixes and their respective sources in a volume invariant way, which helped to improve the quality of the separations. According to the hyperparameters used,

models with a similar order of magnitude of parameters were obtained, 15.9 and 17.8 million parameters for the ConvTasNet and UNet, respectively. Both models were trained using the LogL2 loss function in the time domain and the Adam optimizer with a learning rate of 0.0001 was employed. The code of the models with their respective modifications are publicly available for modification, training and test in https://github.com/nschmidtg/thesis.

The trained models were then evaluated using two strategies. In the first instance they were evaluated using the test partition of the PodcastMix. On the other hand, a second evaluation was performed using a test set of real podcasts, with their respective reference ground truth isolated sources. In the case of ConvTasNet, SDR values of 13.736 were obtained for the synthetic PodcastMix test set, and -2.008 for the real podcasts test set. On the other hand, the UNet gave an SDR of 11.621 for the synthetic test set and 7.382 for the real test set.

These results allow us to conclude that the UNet is able to better separate both speech and music from a mix when comparing with the ConvTasNet. For the synthetic test set the ConvTasNet obtained better values than the UNet for both SDR, SAR and SIR. This result can be explained by the fact that, since UNet works in the time-frequency domain, it allows a better learning of the timbral components that differentiate one source from the other. Models based on spectrogram soft-masks such as UNet have been outperforming those based on waveform both in quality of separations and in training and inference time.

The fact that in podcasting contexts directional pattern microphones are used as opposed to the omnidirectional microphones used in the VCTK dataset may have influenced ConvTasNet's inability to generalize voice characteristics well compared to UNet.

Thus, this master thesis formalizes not only a new task in the field of source separation, the background music foreground speech source separation, but also presents a new dataset called PodcastMix to train new models to perform this task. Moreover, this work makes available the ConvTasNet and UNet models pre-trained on the

dataset, to be used for inference and separation of Podcasts. A benchmark is also presented with respect to the evaluation metrics of these models, to provide a baseline for future researchers that aim to outperform the score of the presented models using the Podcast. Finally, the code used for the elaboration of the dataset, training and evaluation of the models is made available to the community, for reproduction of this study or improvement of the results presented. All these contributions will be presented in a paper to be submitted to ICASSP 2022.

As future work, it would be interesting to train more deep learning models based on both waveform and time-frequency and evaluate how they behave in comparison with the presented ones. In particular, it would be interesting to train and evaluate the Demucs and OpenUnmix models, since they are the ones that nowadays are presenting better results at least for the music source separation task. Also, it would be interesting to apply some data augmentation to the on-the-fly creation of the synthetic podcasts. Reverb, equalization, impulse response convolutions, sound effects, side-chain compression and ducking are some very interesting examples of data augmentation that could be randomly applied to the mixes in order to obtain a broad variety of Podcasts.

# List of Figures

# List of Tables

# Bibliography


[1] The infinite dial 2020. http://www.edisonresearch.com/wp-content/uploads/2020/03/The-Infinite-Dial-2020-U.S.-Edison-Research.pdf (2020). Accessed 03-15-2021.

[2] Nielsen podcast insights. https://www.nielsen.com/wp-content/uploads/sites/3/2019/04/nielsen-podcast-insights-q3-2017.pdf (2017). Accessed 03-15-2021.

[3] Manilow, E., Seetharman, P. & Salamon, J. *Open Source Tools & Data for Music Source Separation* (https://source-separation.github.io/tutorial, 2020). URL https://source-separation.github.io/tutorial.

[4] Herault, J., Jutten, C. & ANS, B. Detection de grandeurs primitives dans un message composite par une architeture de calcul neuromimetique en apprentissage non supervise. *10° Colloque sur le traitement du signal et des images, 1985 ; p. 1017-1022* (1985).

[5] Dey, P., Satija, U. & Ramkumar, B. Single channel blind source separation based on variational mode decomposition and pca. In *2015 Annual IEEE India Conference (INDICON)*, 1–5 (2015).

[6] Dragomiretskiy, K. & Zosso, D. Variational mode decomposition. *IEEE Transactions on Signal Processing* **62**, 531–544 (2014).

[7] MESHRAM, S. Audio separation using principal component analysis (pca). https://sameermeshram.com/engr/projects/audio-separation-using-pca/. Accessed: 2021-02-15.

[8] Comon, P. Independent Component Analysis, a new concept? *Signal Processing* **36**, 287–314 (1994). URL `https://hal.archives-ouvertes.fr/hal-00417283`.

[9] López, J. *Informed Source Separation for Multiple Instruments of Similar Timbre*. Ph.D. thesis, Universitat Pompeu Fabra (2013). URL `https://doi.org/10.5281/zenodo.3754245`.

[10] Comon, P. & Jutten, C. *Handbook of Blind Source Separation, Independent Component Analysis and Applications* (2010).

[11] Casey, M. & Westner, A. Separation of mixed audio sources by independent subspace analysis (2000).

[12] Rafii, Z. & Pardo, B. Repeating pattern extraction technique (repet): A simple method for music/voice separation. *IEEE Transactions on Audio, Speech, and Language Processing* **21**, 73–84 (2013).

[13] Pons, J. Deep neural networks for music: teaching materials. `http://www.jordipons.me/apps/teaching-materials/` (2020). Accessed 02-16-2021.

[14] Luo, Y. & Mesgarani, N. Tasnet: Surpassing ideal time-frequency masking for speech separation (2018).

[15] Jansson, A. *et al.* Singing voice separation with deep u-net convolutional networks. In *ISMIR* (2017).

[16] Kussul, E. & Baidyk, T. Improved method of handwritten digit recognition tested on mnist database. *Image and Vision Computing* **22**, 971–981 (2004). URL `https://www.sciencedirect.com/science/article/pii/S0262885604000721`. Proceedings from the 15th International Conference on Vision Interface.

[17] Stöter, F.-R., Uhlich, S., Liutkus, A. & Mitsufuji, Y. Open-unmix - a reference implementation for music source separation. *Journal of Open Source Software* **4**, 1667 (2019). URL `https://doi.org/10.21105/joss.01667`.

[18] Stöter, F.-R., Uhlich, S., Liutkus, A. & Mitsufuji, Y. Open-unmix: Technical details. https://sigsep.github.io/open-unmix/details.html. Accessed: 2021-03-23.

[19] Li, T., Chen, J., Hou, H. & Li, M. Sams-net: A sliced attention-based neural network for music source separation (2020). 1909.05746.

[20] Luo, Y. & Mesgarani, N. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **27**, 1256–1266 (2019). URL http://dx.doi.org/10.1109/TASLP.2019.2915167.

[21] Chen, J., Mao, Q. & Liu, D. Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation (2020). 2007.13975.

[22] Stoller, D., Ewert, S. & Dixon, S. Wave-u-net: A multi-scale neural network for end-to-end audio source separation (2018). 1806.03185.

[23] Kavalerov, I. *et al.* Universal sound separation (2019). 1905.03330.

[24] Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I. & Bittner, R. Musdb18-hq - an uncompressed version of musdb18 (2019). URL https://doi.org/10.5281/zenodo.3338373.

[25] Smule, I. DAMP-VSEP: Smule Digital Archive of Mobile Performances - Vocal Separation (2019). URL https://doi.org/10.5281/zenodo.3553059.

[26] Hershey, J. R., Chen, Z., Le Roux, J. & Watanabe, S. Deep clustering: Discriminative embeddings for segmentation and separation. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2016). URL http://dx.doi.org/10.1109/ICASSP.2016.7471631.

[27] John S. Garofolo, D. P., David Graff & Pallett, D. Csr-i (wsj0) complete (2007). URL https://catalog.ldc.upenn.edu/LDC93S6A.

[28] Wichern, G. *et al.* WHAM!: extending speech separation to noisy environments. In *Proc. Interspeech*, 1368–1372 (2019). URL `http://dx.doi.org/10.21437/Interspeech.2019-2821`.

[29] Maciejewski, M., Wichern, G., McQuinn, E. & Roux, J. L. Whamr!: Noisy and reverberant single-channel speech separation (2019). `1910.10279`.

[30] Cosentino, J., Pariente, M., Cornell, S., Deleforge, A. & Vincent, E. Librimix: An open-source dataset for generalizable speech separation (2020). `2005.11262`.

[31] Joris, C. & Manuel, P. Minilibrimix dataset (2020). URL `https://doi.org/10.5281/zenodo.3871592`.

[32] Sivasankaran, S., Vincent, E. & Fohr, D. Analyzing the impact of speaker localization errors on speech separation for automatic speech recognition. In *2020 28th European Signal Processing Conference (EUSIPCO)* (2021).

[33] Drude, L., Heitkaemper, J., Boeddeker, C. & Haeb-Umbach, R. SMS-WSJ: Database, performance measures, and baseline recipe for multi-channel source separation and recognition. *arXiv preprint arXiv:1910.13934* (2019).

[34] Veaux, C., Yamagishi, J. & Macdonald, K. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (2017).

[35] Reddy, C. K. A. *et al.* The interspeech 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework (2020). `2001.08662`.

[36] Xia, Y. *et al.* Weighted speech distortion losses for neural-network-based real-time speech enhancement (2020). `2001.10601`.

[37] Wisdom, S. *et al.* What's all the fuss about free universal sound separation data? *in preparation* (2020).

[38] Fonseca, E. *et al.* Freesound datasets: a platform for the creation of open audio datasets. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, 486–493 (Suzhou, China, 2017).

[39] Ephrat, A. *et al.* Looking to listen at the cocktail party. *ACM Transactions on Graphics* **37**, 1–11 (2018). URL `http://dx.doi.org/10.1145/3197517.3201357`.

[40] Vincent, E., Gribonval, R. & Fevotte, C. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing* **14**, 1462–1469 (2006).

[41] Pariente, M. *et al.* Asteroid: the PyTorch-based audio source separation toolkit for researchers. In *Proc. Interspeech* (2020).

[42] Pierre Gérard, S. Z., Laurent Kratz. Jamendo streaming service (2021). URL `https://jamendo.com`.

# Appendix A

# First Appendix

Other relevant statistics from the PodcastMix dataset

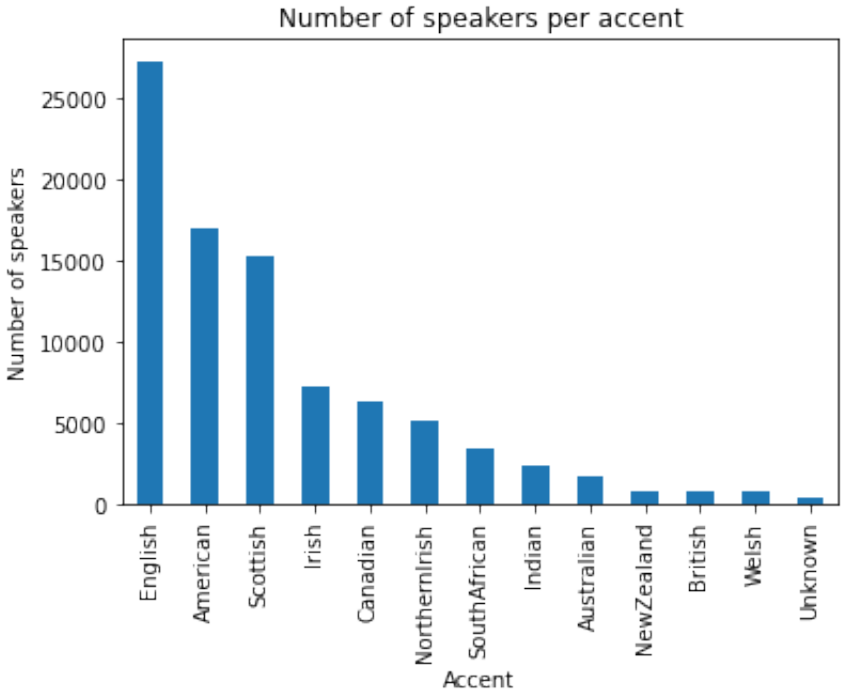


Figure 12: Speakers per accent

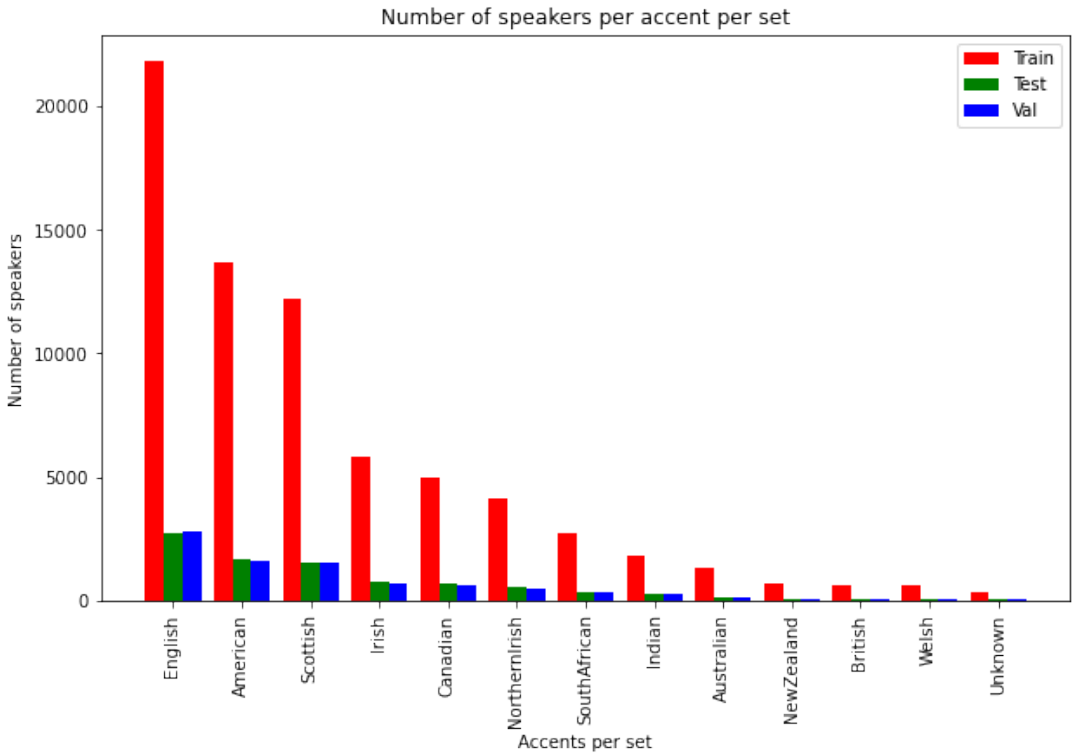


Figure 13: Number of speakers per accent per set

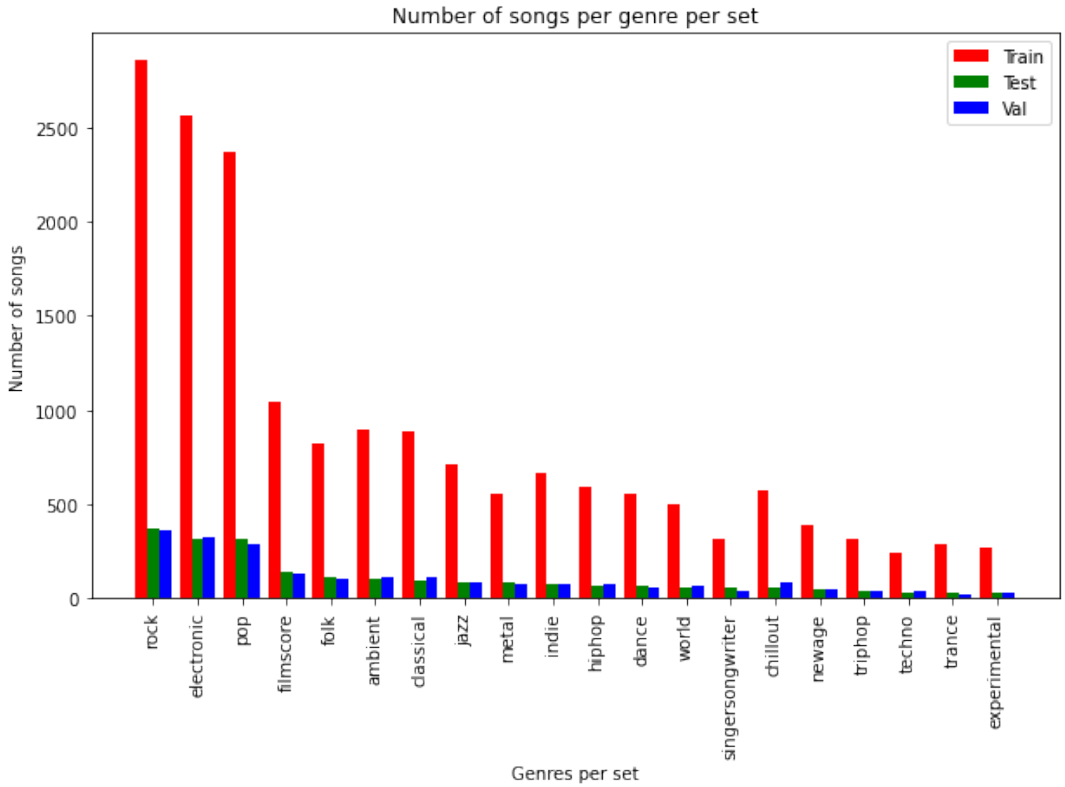


Figure 14: Songs per genre per subset

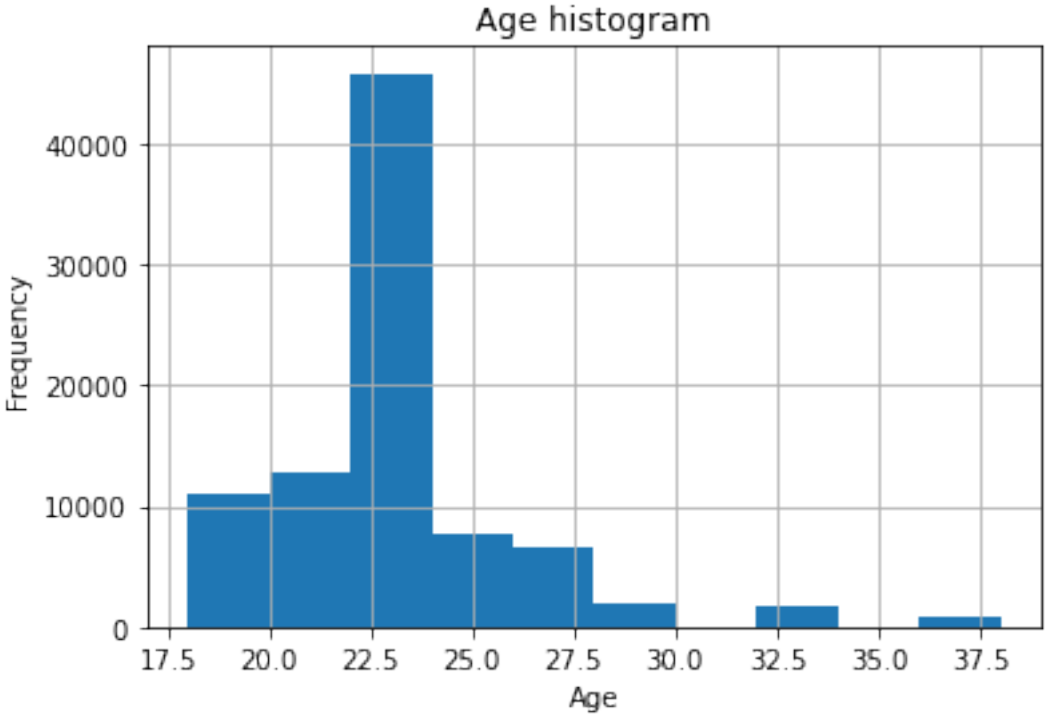


Figure 15: Age histogram