PREPRINT:
G. Peeters and G. Richard, *Deep learning for Audio and Music*,
in "Multi-faceted Deep Learning: Models and Data", J. Benois-Pineau,
A. Zemmari Edts, Springer.

Deep Learning for Audio and Music

Geoffroy Peeters and Gaël Richard

Abstract This chapter provides an overview of how deep learning techniques can be used for audio signals. We first review the main DNN architectures, meta-architectures and training paradigms used for audio processing. By highlighting the specifies of the audio signal, we discuss the various possible audio representations to be used as input of a DNN — time and frequency representations, waveform representations and knowledge-driven representations — and discuss how the first layers of a DNN can be set to take into account these specificity's. We then review a set of applications for three main classes of problems: audio recognition, audio processing and audio generation. We do this considering two types of audio content which are less commonly addressed in the literature: music and environmental sounds.

1 Introduction

As in computer vision (CV) or natural language processing (NLP), deep learning has now become the dominant paradigm to model and process audio signals. While the term ”deep learning” can designate any algorithm that performs deep processing, we define it here as a deep stack of non-linear projections, obtained by non-linearly connecting layers of neurons; a process vaguely inspired by biological neural networks. Such algorithms were denoted by Artificial Neural Network (ANN) in the past and by Deep Neural Network (DNN) since [HOT06].

Deep Learning encompasses a large set of different architectures and training paradigms, distinguishing the way neurons are connected to each others, how spatial or temporal information is taken into account, which criteria are being optimized, for which task the network is supposed to be used for.

Telecom Paris, IP-Paris e-mail: geoffroy.peeters@telecom-paris.fr, gael.richard@telecom-paris.fr

This is often considered as a "zoo" of possible architectures. However most of these architectures share the common use of the back-propagation algorithms [RHW86] to estimate the best parameters of the non-linear projections. In this chapter, we describe the DNN building blocks used for audio processing. While some of them are audio translations of CV or NLP networks, others are specific to audio processing.

We focus on two types of audio content which are less commonly addressed in the literature: music and environmental sounds. For a recent and good overview of DNN applied to speech processing we refer the reader to [KLW19].

Differences between speech, music and environmental sounds. While a speech audio signal usually contains a single speaker (single source), both music and environmental sounds audio signals are made of several simultaneous sources. In the case of music, some sources are polyphonic (as the piano) and can then produce several pitches simultaneously. This makes the analysis of music and environmental sounds particularly challenging. Speech is highly structured over time (or *horizontally* in reference to the commonly used conventions in time-frequency representations of audio signals). This structure arises from the use of a vocabulary and a grammar specific to a language. Music is also highly structured both horizontally (over time) and vertically (various simultaneous sound events). This structure arises from the music composition rules specific to a culture (harmony for Western music, modes/raga for Eastern/Indian music). In the opposite, environmental sounds have no specific temporal structure.

Deep learning for music processing. Music processing is associated with an interdisciplinary research field known as Music Information Research (MIR)[1]. This field is dedicated to the understanding, processing and generation of music. It combines theories, concepts, and techniques from music theory, computer science, signal processing perception, and cognition. MIR deals with the development of algorithms for

- *describing the content of the music* from the analysis of its audio signal. Examples of this are the estimation of the various pitches, chords, rhythm, the identification of the instruments being used in a music, the assignment of "tags" to a music (such as genres, mood or usage) allowing to recommend music from catalogues, the detection of cover/plagiarism in catalogue or in user-generated contents.
- *processing the content of the music*. Examples of this are enhancement, source separation.
- *generating new audio signals or music pieces*, or transferring properties from one signal to another.

Deep learning for environmental sounds processing. Environmental sound processing is associated with the research field known as Detection and

[1] http://ismir.net

Classification of Acoustic Scenes and Events (DCASE)[2]. The latter deals with the development of algorithms for

- *classifying acoustic scenes* (identify where a recording was made – for example in a metro station, in an office or in a street –),
- *detecting sound events* (detect which events occur over time in an audio scene – a dog barking, a car passing, an alarm ringing –),
- *locating these events in space* (in azimuth and elevation angles).

Historical perspectives. Using DNN algorithms to represent the audio signal has been proposed as early as [WHH$^+$90] where Time-Delay Neural Network (TDNN) where proposed to allow the representation of the time-varying natures of phonemes in speech. Later, [BM94] in their "connectionist speech recognition" convincingly demonstrated the use of the discriminative projection capabilities of DNN to extract audio features. This has lead, among others, to the development of the "tandem features" [HES00] which use the posterior probabilities of a trained Multi-Layer-Perceptron (MLP) as audio features or the "bottleneck features" [GKKC07] extracted from the bottleneck part of a MLP. This has lead today to the end-to-end speech recognition systems which inputs are directly the raw audio waveforms and the output the transcribed text [SWS$^+$15, SVSS15]. As 2012 is considered a landmark year for CV (with the AlexNet [KSH12] network wining the ImageNet Large Scale Visual Recognition Challenge), it is also one for speech recognition with the publication of the seminal paper [HDY$^+$12], jointly written by the research groups of the University of Toronto, Microsoft-Research, Google, and IBM-Research demonstrating the benefits of DNN architectures for speech processing.

The same year [HBL12] published a manifesto promoting the use of DNN for non-speech audio processing (MIR and DCASE). In this paper, the authors demonstrated that any hand-crafted feature (such as MFCC or Chroma) or algorithms (such as pitch, chord or tempo estimation) used so far are just layers of non-linear projections and pooling operations and can therefore be profitably replaced by the trainable non-linear projections of DNN. DNN has now become the dominant paradigm in MIR and DCASE .

Chapter organization. In part 2, we first review the commonly used DNN architectures, meta-architectures and training paradigms used for audio processing. In part 3, we review the various types of audio representations used as input to DNN and the proposals made to adapt the first layers of the DNN to take into account the audio specificities. In part 4, we present a set of common MIR and DCASE applications for content description, processing and generation. We also discuss how Semi-Supervised Learning and Self-Supervised Learning are currently developed in these fields to face the lack of large annotated datasets. Finally, in part 5, we discuss future directions for deep learning applied to audio processing.

[2] http://dcase.community

2 DNN architectures for audio processing

A DNN architecture defines a function f with parameters θ which output $\hat{y} = f_\theta(x)$ approximates a ground-truth value y according to some measurements. The parameters θ are (usually) trained in a supervised way using a set of of N inputs/outputs pairs $(x^{(i)}, y^{(i)})\ \ i \in \{1, \ldots, N\}$. The parameters θ are then estimated using one variant of the Steepest Gradient Descent algorithm, using the well-known back-propagation algorithm to compute the gradient of a Loss function w.r.t. to the parameters. The function f defines the architecture of the network. We first review the most popular architectures.

2.1 DNN architectures

Multi-Layer-Perceptron (MLP). An MLP is an extension of the Perceptron [Ros57] in which many perceptrons[3] are organized into layers in a Fully-Connected (FC) way. FC denotes the fact that each neuron $a_j^{[l]}$ of a layer $[l]$ is connected to all neurons $a_i^{[l-1]}$ of the previous layer $[l-1]$. The connection is done through multiplication by weights $w_{ij}^{[l]}$, addition of a bias $b_j^{[l]}$ and passing through a non-linearity activation g (the common sigmoid, tanh or ReLu functions): $a_j^{[l]} = g(\vec{a}^{[l-1]}\vec{w}_j^{[l]} + b_j^{[l]})$. Each $\vec{w}_j^{[l]}$ therefore defines a specific projection j of the neurons of the previous layers.

Convolutional Neural Network (CNN). The FC architecture does not assume any specific organisation between the neurons of a given layer $[l]$. This is in contract with Vision where neurons representing nearby pixels are usually correlated (the adjacent pixels that form a "cat's ear") and far away ones uncorrelated. It would therefore be beneficial to consider a *local connectivity* of the neurons i. Also in the FC architecture the weights are specific to each connection and never re-used. This is in contrast with Vision where the neurons representing the two "cat's ears" would benefit from having the same projection, hence from *sharing their weights*. These two properties led to the development of the CNN architecture [FM82, LBBH98]. In this, the projections are defined by J small filters/kernels $\vec{W}_j$ (which size (h, w) is usually (3,3) or (5,5)) which are convolved along the two spatial dimensions (height and width) of the input images $\vec{X}$ (or previous layer output $\vec{A}^{[l-1]}$). These filters are the trainable parameters of the network. In classic CV, such filters would allow to detect edges or corners. The output of the convolution is a new set of J images $\vec{A}_j^{[l]}$ considered as a 3D tensor $\vec{A}^{[l]}$ of depth J. This tensor then serves as input to the following layers: $\vec{A}_{j'}^{[l+1]} = g(\vec{A}^{[l]} \circledast$

[3] While the Perceptron uses a Heaviside step function, MLP uses non-linear derivable functions.

$\vec{W}_{j'}^{[l+1]} + b_{j'}^{[l+1]})$ (where g denotes a non-linear activation, $\circledast$ the convolution operator, $\vec{W}_{j'}^{[l+1]}$ is a tensor of dimensions (h, w, J)). Spatial invariance (such as detecting the presence of a "cat's ear" independently of its position in the image) is achieved by applying pooling operators. The most popular pooling operator is the max-pooling which only keeps the maximum value over a spatial region. CNN is the most popular architecture in CV.

Temporal Convolutional Networks (TCN). While attempts have been made to apply CNN to a 2D representation of the audio signal (such as its spectrogram), recent approaches [DS14] use **1D-Convolution** directly applied on the raw audio waveform $x(n)$. The filters $\vec{W}_{j'}$ have then only one dimension (the time) and are convolved only over the time axis of the input waveform. The motivation of using such convolution is to learn better filters than the ones of usual spectral transforms (for example the sinus and cosinus of the Fourier transform). However, compared to images, audio waveforms are of much higher dimensional. To understand this we consider their respective Receptive field (RC). The RC is defined as the portion of the input data to which a given neuron responds. Because images are usually low dimensional (256x256 pixels), only a few layers is necessary in CV to make the RC of a neuron cover the whole input image. In contrast, because input audio waveform are very high dimensional (1 second of audio leads to 44100 samples), the number of layers to make the RC cover the whole signal becomes very large (as the number of parameters to be trained). To solve this issue, [vdODZ$^+$16] have proposed in their WaveNet model the use of **1D-Dilated-Convolutions** (also named convolution-with-holes or atrous-convolution). For a 1D-filter w of size l and a sequence $x(n)$, the usual convolution is written $(x \circledast w)(n) = \sum_{i=0}^{l-1} w(i)x(n-i)$; the dilated convolution with a dilatation factor d is written $(x \circledast_d w)(n) = \sum_{i=0}^{l-1} w(i)x(n-(d \cdot i))$, i.e. the filter is convolved with the signal only considering one over d values. This allows to largely extend the RC and then allows the model to capture the correlations over longer time ranges of audio samples. The 1D-Dilated-Convolutions is at the heart of the **Temporal Convolutional Networks (TCN)** [BKK18] which is very popular in audio today. The TCN adds a causality constraint (only data from the past are used in the convolution) and, similar to the ResNet cells, stacks two dilated-convolutions on top of each other (each followed by a weight normalization, ReLu and DropOut) with a parallel residual path.

Recurrent Neural Network (RNN). While CNN allows representing the spatial correlations of the data, they do not allow to represent the sequential aspect of the data (such as the succession of words in a text, or of images in a video). RNN [RHW86] is a type of architecture, close to the Hopfield networks, in which the internal/hidden representation of the data at time t, $\vec{a}^{<t>}$, does not only depend on the input data $\vec{x}^{<t>}$ but also on the internal/hidden representation at the previous time $\vec{a}^{<t-1>}$: $\vec{a}^{<t>} = g(\vec{x}^{<t>}\vec{W}_{xa} + \vec{a}^{<t-1>}\vec{W}_{aa} + \vec{b}_a)$. Because of this, RNN architectures have become the standard for processing sequences of words in NLP tasks[4]. While RNN can theoretically represent long-term dependencies, because of a problem known as the vanishing gradient through time, they cannot in practice. For this reason, they have been replaced by the more sophisticated cells Long Short Term Memory (LSTM)[HS97] or Gated Recurrent Units (GRU)[CVMG$^+$14] in which a set of gates (sigmoids) allow the storage and delivery of information from a memory over time.

2.2 DNN meta-architectures

The above MLP, CNN and RNN architectures can then be combined in "meta-architectures" which we describe here.

Auto-Encoder (AE). AE is a type of network made of two sub-networks. The encoding network ϕ_e projects the input data $\vec{x} \in \mathbb{R}^M$ in a latent space $\vec{z} \in \mathbb{R}^d$ of smaller dimensionality ($d << M$): $\vec{z} = \phi_e(\vec{x})$. The decoder network then attempts to reconstruct the input data from the latent dimension $\hat{\vec{y}} = \phi_d(\vec{z})$. The encoding and decoding networks can be any of the architectures described above (MLP, CNN, RNN). The training is considered unsupervised since it does not necessitate ground-truth labels. We train the network such that $\hat{\vec{y}}$ is a good reconstruction (usually according to a Mean Square Error (MSE) loss) of the input $\vec{x}$: $\arg\min_{\phi_e,\phi_d} ||\vec{x} - (\phi_d \circ \phi_e(\vec{x}))||^2$. AEs are often used for feature learning (learning a representation, a latent space, of the input data). Many variations of this vanilla AE have been proposed which allow improving the properties of the latent space, such as Denoising AE, Sparse AE or Contractive AE.

Variational Auto-Encoder (VAE). For generation, the most popular form of AE is probably today the VAE [KW14a]. In contrast to the vanilla AE, the VAE is a generative model, i.e. a model in which one can sample points $\vec{z}$ in the latent space to generate new data $\hat{y}$. In a VAE, the encoder models the posterior $p_\theta(\vec{z}|\vec{x})$ while the decoder (the generative network) models the likelihood $p_\theta(\vec{x}|\vec{z})$. However because $p_\theta(\vec{z}|\vec{x})$ is untractable, it is approximated by $q_\phi(z|x)$ (variational Bayesian approach) which is set (for mathematical simplicity) to a Gaussian distribution which parameters $\vec{\mu}$ and $\vec{\Sigma}$ are the outputs of the encoder. Minimizing the Kullback-Leibler divergence between $q_\phi(z|x)$ and $p_\theta(\vec{z}|\vec{x})$ is mathematically equivalent to maximizing an ELBO (Evidence Lower BOund) criteria. For the later, a prior $p_\theta(\vec{z})$ needs to be set. It is set (again for mathematical simplicity) to $\mathcal{N}(0,1)$. The goal is then to maximize $\mathbb{E}_q[\log p(x|z)]$. This can be estimated using a Monte-Carlo method, i.e. maximizing $\log p(x|z)$ (the reconstruction error) over samples $z \sim q_\phi(z|x)$

[4] They are then often combined with representation of the vocabulary using word-embedding techniques

given to the decoder. Given the smoothness of the latent space $\vec{z}$ obtained (in contrast to the one of vanilla AE) it is adequate for sampling and generation.

Generative Adversarial Network (GAN). Another popular type of network for generation is the GAN [GPAM+14]. GAN only contains the decoder part of an AE here named "Generator" G. Contrary to the VAE, z is here explicitly sampled from a chosen distribution $p(z)$. Since z does not arise from any existing real data, the Generator $G(z)$ must learn to generate data that look real, i.e. the distribution of the generated data p_G should look similar to the ones of real data p_{data}. Rather than imposing a distribution (as in VAE), this is achieved here by defining a second network, the "Discriminator" D, which goal is to discriminate between real and fake (the generated ones) data. D and G are trained in turn using a minmax optimisation. For G fixed, D is trained to recognize real data from fake ones (the ones generated by G)[5]. For D fixed, G is then trained to fool D[6].

Encoder/Decoder (ED). While the goal of AE is to encode the data into a latent space $\vec{z}$ such that it allows reconstructing the input, ED [CVMG+14] or Sequence-to-Sequence [SVL14] architectures aim at encoding an input sequence $\{\vec{x}^{<1>} \ldots \vec{x}^{<t>} \ldots \vec{x}^{<T_x>}\}$ into $\vec{z}$ which then serves as initialization for decoding a sequence $\{\vec{y}^{<1>} \ldots \vec{y}^{<\tau>} \ldots \vec{y}^{<\tau_y>}\}$ into another domain. Such architectures are for example used for machine translation where an input English sentence is translated into an output French sentence. Both sequences have usually different length $T_x \neq \tau_y$. In machine translation both encoder and decoder are RNNs (or their LSTM or GRU versions). In image captioning [VTBE15], a deep CNN is used to encode an input image into $\vec{z}$; $\vec{z}$ then serves as initialization of a RNN decoder trained to generate the text of image captions.

Attention Mechanism. In the original ED for machine translation [CVMG+14], $\vec{z}$ is defined as the internal states of the RNN after processing the whole input sequences, i.e. at the last encoding time step $\vec{a}^{<T_x>}$. It quickly appeared that doing so prevents from correctly translating long sentences. [BCB14] therefore proposed to add to the ED architecture, an attention mechanism. The latter provides a mechanism to let the decoder chose at each decoding time τ the most informative times t of the encoding internal states $\vec{a}^{<t>}$. This mechanism is a small network trained to align encoding and decoding internal states.

Transformer. Recently it has been shown [VSP+17] that only the attention mechanism was necessary to perform machine translation. The transformer still has an encoder and a decoder part but those are now simple stacks of so-called self-attention mechanisms coupled with a FC. At each layer, the self-attention mechanisms encode each element of the sequence taking into account its relationship with the other elements of the sequence. This is done

[5] $D(x \sim p_{data})$ should output "real" while $D(G(z))$ should output "fake"

[6] $D(G(z))$ should output "real"

using a simple query, key and value mechanism. The transformer has become very popular for sequence processing.

2.3 DNN training paradigms and losses

The most popular training paradigms for DNN are classification, reconstruction and metric learning.

Classification. The simplest case of classification, is the *binary classification*. In this, the network has a single output neuron (with sigmoid activation) with predicts the likelihood of the positive class $\hat{y} = p(y = 1|x)$. The training of the network is achieved by minimizing the Binary-Cross-Entropy (BCE) between y and $\hat{y}$ over the N training examples: $\mathcal{L} = -\sum_{i=1}^{N}[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$. The goal of *multi-class classification* is to predict a given class c among C mutually exclusive classes. Each class c is represented by an output neuron y_c (with a softmax activation) which predicts $\hat{y_c} = p(y = c|x)$ The training of the network is then achieved by minimizing the general cross-entropy between the y_c and the $\hat{y_c}$. The goal of *multi-label classification* is to predict a set of class $\{c_i\}$ among C non-mutually exclusive classes. The most usual solution to this problem is to consider each class c as an independent binary classifier (with sigmoid activation) and then train the network by minimizing the sum of the BCE of each class c.

Reconstruction. When the goal of the network is to reconstruct the input data (such as with AE), the simple MSE between the output and input data is used: $MSE = \sum_{i=1}^{N} ||\vec{x}^{(i)} - \hat{\vec{y}}^{(i)}||^2$.

Metric Learning. Metric learning aims at automatically constructing distance metrics from data, in a machine-learning way. DNN provides a nice framework for this. In this, the parameters θ of a network are learnt such that a distance function $g(f_\theta(x), f_\theta(y))$ is minimized for similar training samples x and y and maximized for dissimilar samples. Methods proposed for that, mainly differ on the way these two constrains are represented: they are represented in turns in Siamese networks [BGL+94] and contrastive loss [HCL06], they are represented simultaneously in the triplet loss [SKP15]). In the later, three data are simultaneously considered: an anchor a, a positive p (similar to a) and a negative n (dissimilar to a). The goal is to train the network such that $P = f_\theta(p)$ will be closer to $A = f_\theta(a)$ than $N = f_\theta(n)$ is to A. For safety a margin α is added leading to the definition of the triplet loss to be minimized $\mathcal{L} = \max(0, g(A, P) + \alpha - g(A, N)$. g can be a simple Euclidean distance.

3 DNN inputs for audio processing

A wide variety of audio representations are used as input for DNN. These representations can be broadly classified in 1) time and frequency representations; 2) waveform representations 3) knowledge-driven representations and 4) perceptual-driven representation. The latter is not discussed in details in this chapter but the interested readers are referred to [RSN13] for an overview of popular perceptually-based representations for audio classification tasks.

3.1 Using time and frequency representations as input

A recorded audio signal $x(t)$ represents the evolution of the sound pressure x over time t. In its discrete version, the time dimension is discretized in samples m resulting in a discrete sequence $x(m)$. The number of samples sampled from x during one second is named “sampling rate”. A common value for it is 44100 Hz. One second of audio signal is then represented by the sequence $\{x(1), \ldots x(44100)\}$. To represent a piece of music of 4 minutes duration, this would lead to a very high number of values .

For a discrete non-periodic signal, the Discrete-Fourier-Transform (DFT) is used to represent $x(m)$ over discrete frequencies $k \in [0, N-1]$:

$$X(k) = \sum_{m=0}^{N-1} x(m) e^{-j 2\pi \frac{k}{N} m}$$

Since the content of the audio signal varies over time (for example it is assumed that the phoneme rate in speech is around 4Hz), DFTs are computed over successive time frames of the signal (obtained through multiplication with an analysis window $h(m)$) leading to the well-known Short-Time Fourier Transform (STFT):

$$X(k, n) = \sum_{m=0}^{N-1} x(m) h(n-m) e^{-j 2\pi \frac{k}{N} m}$$

$X(k, n)$ represents the content of the audio signal at frequency k and around time n.

The complex value STFT matrix $X(k, n)$, can be represented by its real and imaginary parts or by its amplitude (which represents the amount of periodicity at a given frequency) and phase (which represents the location at a given frequency). Most approaches that use the STFT to represent the audio, only consider its amplitude. It is then often denoted as the **spectrogram**. Since the later can be displayed as an “image”, the first audio-DNN used standard computer vision CNNs applied to this spectrogram-image.

Recently, it has been proposed to use directly the **complex STFT** as input to DNN with the goal of benefiting from the location information contained in the phase. For this, either a (real,imaginary) or a (amplitude, instantaneous frequency) representation have been tested.

Before the rise of deep learning for audio, the most popular audio representation for speech tasks (recognition/ identification/ diarization), MIR or DCASE tasks was the **Mel-Frequency-Cepstral-Coefficients (MFCC)s**. Those are obtained by computing the real cepstrum representation (Discrete Cosine Transform (DCT) applied to the logarithm-amplitude of the DFT) on a Mel-scale representation[7]. It can be shown that in the case of a source-filter sound production model (see section 3.3), the cepstrum allows to separate the contribution of the filter (the lowest coefficients of the cepstrum) from the source (highest coefficients). These lowest coefficients are therefore usually used to obtain a compact representation of the spectrum envelope (or formants of the various vowels in vocal signals or the timbre of musical instruments) independently of their pitch. In the MFCCs computation, the DCT is used to make the various dimensions of the MFCC somehow decorrelated. This is needed since those are often represented in speech acoustical models using Gaussian mixture distributions with diagonal covariance matrices. Because this de-correlation of the input is not required in the case of DNN, the **Log-Mel-Spectrogram (LMS)** (hence without the DCT de-correlation) has been widely adopted. This leads to a time versus mel-band-frequency matrix representation.

In the DFT, the time and frequency resolution (we mean by resolution the possibility provided by the representation to distinguish two adjacent time or frequency components) remains constant over time and frequency. This limitation led to the development of the wavelet analysis [Mal89] which allows for a finer spectral resolution at low-frequencies and finer temporal resolution at high-frequency. The **Constant-Q-Transform (CQT)** [Bro91] has been proposed as a form of wavelet analysis adapted to musical signals, i.e. which allows distinguishing the various possible pitches of the musical scale. As for the wavelet representation, this is achieved by using analysis windows $h(m)$ which durations are inversely proportional to the various musical pitch frequencies. The CQT follows a logarithmic frequency scale (as the musical pitches). It is therefore said to be shift-invariance in pitch, i.e. transposing a note (changing its pitch) simply results in a shift of its harmonic pattern (the sequence of its harmonics) along the log-frequency axis. This is however not entirely true as we will discuss later considering the source/filter decomposition.

[7] The Mel scale is a perceptual scale of pitch height perception. A mel-filter bank is then a set of filters whose bandwidth center frequencies are equally spaced on the Mel scale (or logarithmically spaced in Hertz).

Spectrogram images versus natural images

While spectrograms are often processed using CNN and hence considered as images, there is a large difference between this image and a natural image, such as a cat picture.

In **natural images** (see Figure 3.1 left), the two axis x and y represent the same concept (spatial position). The elements of an image (such as a cat' ear) have the same meaning independently of their positions over x and y. Also neighboring pixels of an image are usually highly correlated and often belong to the same object (such as the cat's ear). The use of CNN, and its inherent properties (hidden neurons are only locally connected to the input image, parameters are shared between the various hidden neurons of a same feature map and max pooling allows spatial invariance) are therefore highly appropriate to process such data.

In **time-frequency audio representations** (such as the spectrogram, the LMS or the CQT) (see Figure 3.1 right), the two axis x and y represent profoundly different concepts (time and frequency). The elements of a spectrogram (such as a time-frequency area representing a sound source) has the same meaning independently of its position over time but not over frequency. There is therefore no invariance over y, even in the case of log-frequencies. Neighboring pixels of a spectrogram are not necessarily correlated since a given sound source (such has an harmonic sound) can be distributed over the whole frequency in a sparse way (the harmonics of a given sound can be spread over the whole frequency axis). It is therefore difficult to find a local structure using a CNN.

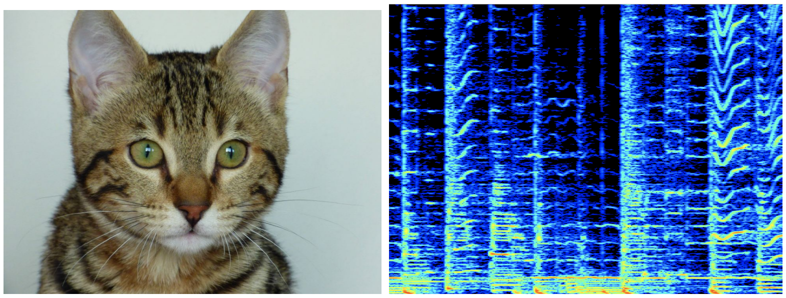

Fig. 1 [Left part] Natural image of cats, [Right part] image of a spectrogram

DNN models for time and frequency representations as inputs

It is generally considered that DNNs learn a hierarchical feature representation of the input. However one can easily consider that the first layers are more associated with this feature learning while the last layers to the task at hand, e.g. a classification task. We review here which choices have been made so far to allow these first layers to deal with audio inputs as time and frequency representations.

In **speech**, one of the first attempt to apply DNN to the audio signal is using a so-called Time-Delay Neural Network (TDNN) [WHH$^+$90]. This architecture is similar to a 1-D convolution operating only over time. In [WHH$^+$90], this convolution is applied to a Mel-gram (16 normalized Mel-scale spectral coefficients). No convolution are performed over the frequency axis. In the works following the "connectionist speech recognition" approaches [BM94] ("tandem features" [HES00] or "bottleneck features" [GKKC07]), a context window of several successive frames of a feature vector (such as MFCC) is used as input to an MLP. Here the convolutions over time is replaced by a context-window. No convolution are performed over the frequency axis. In [LPLN09], a Convolutional Deep Belief Networks (CDBN)[8] is used to process the audio input. The audio input is a 160 dimensional spectrogram which is then PCA-whitened to 80 dimensions[9]. The filters (named bases in [LPLN09]) of the first and second layers are of length 6 and are convolved over the PCA-whitened spectrogram. By visual comparison, it is shown that the learned filters (bases) are related to the different phonemes of speech. Following this, the seminal paper [HDY$^+$12] defines the new baseline for speech recognition system as the DNN-HMM model. In this, the acoustic model part of the system is defined as a DNN model (more precisely as stacked RBMs).

In **music**, [Die14] also consider a 1D-convolution operating only over time. For a task of predicting latent representation[10] of music tracks (a regression problem), they use as input of a 1D-CNN a Mel-Spectrogram (MS) of 128 frequency bins. The filters of the first layer are of shape (time=4,frequency=128) and only convolved over time.

In the opposite [CFS16] consider time/frequency representation as natural images and apply a computer vision CNN to it. The network is a VGG-Net [SZ15], i.e. a deep stack of convolution layers with small (3,3) filters convolved over the time and frequency axis. With this architecture, they show that using MS as input performs better than STFT or MFCC.

However, as described in part 3.1, time/frequency representations cannot be considered as a natural image. When using CNN architectures, one should carefully choose the shape of the filters and the axis along which the convolution is performed.

One of the first work to deal with this necessary adaptation is [SB13]. For a task of onset detection (detecting the start of a musical events) they carefully design the filters to allow highlighting mid-duration variations over small-

[8] A CDBN is a stack of Restricted Boltzman Machine (RBM) with convolutions operations, hence trained in an unsupervised way.

[9] Each of the whitened dimension is therefore a combination of the initial 160 dimensions of the spectrogram.

[10] The latent representation resulting from a collaborative filtering model.

frequency ranges. For this, they specify filters of shape (time=7,frequency=3). An LMS representation is then convolved over time and frequency with these filters. Another specificity of their approach is to allow the representation of multi-scale analysis, i.e. STFT computed using various window durations (23 ms, 46 ms and 93 ms) to better face the time/frequency resolution trade-off. They use the depth of the input layer[11] to represent the various scales. The resulting onset detection algorithm has remain the state-of-the-art for a long time.

The work presented in [PLS16] is entirely devoted to this musically-motivated filter design. In their work, the shapes of the CNN filters are carefully chosen to allow representing the timbre (vertical filters extending over the frequency axis) or the rhythm (horizontal filters extending over the time axis) content of a music track. They show that carefully choosing the shape of the filters allows to obtain equivalent performances than the CV-based approach of [CFS16] (here renamed “black-box”) but with much less parameters.

3.2 Using waveform representations as input

While musically-motivated CNN filter shape is a promising path, one still has to manually design this shape for a given application. Also one has to decide what is the most appropriate 2D representation (STFT, LMS or CQT) and its parameters (window size, hop size, number of bands) for a given application.

For these reasons, the so-called “end-to-end” approaches have been developed. Those consider directly the raw audio waveform as input.

In **speech**, one of the first end-to-end approaches is the one of [JH11] where a RBM is used to model the raw speech signals.

In **music**, one of the first end-to-end approaches is the one of [DS14] who proposed, for a music auto-tagging task, to use 1D-convolution (a convolution over time with 1D-filters) on the waveform as a replacement to the spectrogram input. To compare both, [DS14] actually reproduce the computation of the spectrogram using 1D-convolution. While a spectrogram is computed using a succession of DFTs each computed on an audio frame of length N and each separated by a hop size S, the 1D-convolution is computed using 1D-filters of length N[12] and a stride of S[13]. However, their “end-to-end” approach under-performed the traditional spectrogram-based one. This may be due to the lack of Time Translation Invariance (TTI) of their representation.

Time Translation Invariance (TTI) is a property of a transform that makes it insensitive to time translation (or phase shift) of the input. The amplitude

[11] In CV the depth is used to represent the RGB channels of an image.

[12] In his experiments $N \in 256, 512, 1024$ for a sampling rate of 16 kHz.

[13] $S \in 256, 512, 1024$.

of the DFT (as used in the spectrogram) is TTI. This is because the DFT projects the waveform on orthogonal cosinus and sinus basis, and the modulus of the resulting vectors remain invariant to time translation (the phase of the vectors are however shifted according to the time translation). Mimicking this property with 1D-convolution would require (a) reducing the stride to S=1 (and using a very high sampling rate) or (b) having a different 1D-filter for each possible time translation. One will still needs to perform a max-pooling over time-steps for (a) or over filters for (b). Both are however computationally prohibitive.

Sample-CNN: One way to improve the TTI is to reduce the size of the 1D-convolution filters (hence also the stride). If the filters are smaller, then the number of time translation to be learned is also reduced. This is the idea developed in the Sample-CNN [LPKN17] [KLN18] network. The later can be considered as an equivalent to the VGG-Net for 1D-convolution applied to waveforms. It is a deep stack of 1D-convolution of small (3,1) filters applied to the waveform. Sample-CNN was shown to slightly outperforms the 2D-CNN on the spectrogram.

Multi-Scale: When computing a spectrogram, the choice of the window size fixes the trade-off between time and frequency resolution. One can think of the same for the choice of the filter size N of 1D-convolution. To get around this choice, [ZEH16] propose a multi-scale approach where the waveform is simultaneously convolved in parallel with filters of different sizes (1ms, 5ms and 10ms). The resulting outputs are then concatenated. This idea follows the one of the Inception network [SLJ+15] in computer vision.

3.3 Using knowledge-driven representations as input

When one has some knowledge of the sound production process it is possible to use this knowledge to better shape the input and/or the first layer of the network. Such commonly used sound production processes are the source/filter and the harmonic models. The *source/filter model* considers that the sound $x(t)$ results from the convolution of a periodic (excitation) source signal $e(t)$ (such as the glottal pulses in the case of voice) with a filter $v(t)$ (such as the vocal track in the case of voice): $x(t) = (v \circledast e)(t)$. The *harmonic model* considers that a sound with a pitch f_0 can be represented in the spectral domain as the sum of harmonically related components at frequencies $hf_0, h \in \mathbb{N}^+$ with amplitudes a_h.

[LC16] were among the first to use such models for a task of musical instrument recognition. Below a cut-off frequency, they consider the harmonic model: the spectrum of harmonic sounds is sparse and co-variant with pitch. It is therefore processed using convolution filters which only have values at octave intervals (mimicking Shepard pitch spiral array). Above this cut-off frequency, they consider the source/filter model: the spectrum is dense and independent of pitch (according to the source/filter model, transposed sounds have similar spectra). It is therefore processed with filters that extent over the whole upper-part of the spectrum.

Harmonic CQT. [BMS$^+$17] also use the harmonic assumption; this for a task of dominant melody and multi-pitch estimation. Contrary to natural images (where neighboring pixels usually belong to the same source), the harmonics of a given sound source are spread over the whole spectrum and can moreover be interleaved with the harmonics of other sound sources. [BMS$^+$17] propose to bring back this vicinity of the harmonics by projecting each frequency f into a third dimension (the depth of the input) which represents the values of the spectrum at the harmonics hf. Convolving this representation with a small time and frequency filter but which extends over the whole depth allow then to model easily the specific harmonic series of pitched sounds hence detecting the dominant melody. This representation is named Harmonic CQT and led to excellent results in this context. This approach has been extended with success by [FP19] for a task case of tempo estimation. In this, pitch frequencies are replaced by tempo frequencies and the CQT of the audio signal by the one of onset-energy-functions.

Source/Filter. Still for a task of dominant melody estimation, [BEP18] use the source/filter assumption. Rather than considering the audio as input to the network, they consider the output of a Non Negative Matrix Factorization (NMF) model. They use the NMF source/filter model of [DRDF10] and use the source activation matrix as input to the network. They show that including the knowledge of the production model allows to drastically reduce the size of the training set.

SincNet. In the end-to-end approaches mentioned above, the filters of the 1D-convolution are often difficult to interpret since their shape are not constrained. While being learned in the temporal domain, authors often display them in the frequency domain to demonstrate that meaningful filters have been learned (such as Gamma-tone filters in [Sai15]). With this in mind, the SincNet model [RB18] proposes to define the 1D-filters as parametric functions g which theoretical frequency responses are parameterizable band pass filters. To do so g is defined in the temporal domain as the difference between two sinc functions which learnable parameters define the low and high cutoff frequencies of the band-pass filters. They show that not only the obtained filters are much more interpretable but also the performances for a task of speaker recognition is much improved. This idea has been extended recently to the complex domain in the Complex Gabor CNN [NPM20].

HarmonicCNN. Combining the idea of SincNet with the harmonic model lead to the HarmonicCNN of [WCNS20]. In this the 1D-convolution is performed with filters constrained as for SincNet but extended to the harmonic dimensions (stacking band-pass filters at harmonic frequencies hf_c).

Neural Autoregressive models. A source/filter model $x(n) = (v \circledast e)(n)$ can be associated to an autoregressive model, i.e. the value $x(n)$ can be predicted as a linear combination of its P preceding values: $x(n) =$

$\sum_{p=1}^{P} a(p)x(n-p)$. Neural Auto-regressive models are a non-linear form of auto-regressive models in which the linear combination is replaced by a DNN. The two most popular models are probably the Wavenet [vdODZ^{+}16] and the SampleRNN [MKG^{+}17] architectures. In WaveNet, the conditional probability distribution $p(x_n|x_1,\ldots,x_{n-1})$ is modeled by a stack of dilated 1D convolutions. To facilitate the training, the problem is considered as a classification problem. For this $x(n)$ is discretized into 256 possible values (8 bits using μ-law) considered as classes to be predicted by a softmax. The model has been developed for speech generation and can be conditioned on side information $\mathbf{h}$ such as speaker identity or text: $p(x_n|x_1,\ldots,x_n-1,\mathbf{h})$. While WaveNet relies on dilated convolutions to allow both short term and long term dependencies in $p(x_n|x_1,\ldots,x_{n-1})$, SampleRNN, uses a stack of RNNs each operating at a different temporal scale[14].

DDSP. The recently proposed Differentiable Digital Signal Processing (DDSP) [EHGR20] is probably the DNN models which relies the most on the prior knowledge of the sound production process. Just as SincNet defines the 1D-filters as parametric functions g and the training consists in finding the parameters of g, DDSP defines the sound production model and the training consists in finding its parameters. The model considered here is the Spectral Modeling Synthesis (SMS) model [SS90]. It combines harmonic additive synthesis (adding together many harmonic sinusoidal components) with subtractive synthesis (filtering white noise); it also adds room acoustics to the produced sound through reverberation. In DDSP, the input audio signal x is first encoded into its pitch f_0 and a latent representation $\vec{z}$. Time-varying loudness $l(t)$, $f_0(t)$ and $\vec{z}(t)$ are then fed to a decoder which estimates the control parameters of the additive and filtered noise synthesizers.

4 Applications

4.1 Music content description

As described in part 1, Music Information Research (MIR) encompasses a large set of tasks related to the description of the music from the analysis of its audio signal. Since almost all possible audio front-ends and DNN algorithm have been tested for each task, it is useless to describe them all. We rather focus here on some iconic MIR tasks and an iconic DNN algorithm proposed to solve each.

Beat-tracking. “Beat” or “pulse” is the basic unit of time in music. It is often defined as the rhythm listeners would tap their foot to when listening to a piece of music. Beat-tracking is the task of estimating the temporal po-

[14] RNN layers operate at different temporal resolutions and are followed by up-sampling for the next scale.

sition of the beats within a music track. As far as 2011, i.e. before the rise of deep learning for audio, [BS11] already proposed a fully DNN system to estimate the beat positions. The input to the network is made of three Log-Mel-Spectrogram (LMS) computed with window sizes of 23.2 ms, 46.4 ms, 92.8 ms and their corresponding positive first order median difference. Since “beat” is a temporal phenomenon, [BS11] proposes to use an RNN architecture to estimate it. The network is made of three layers of bi-directional LSTM units. The last layer has a softmax activation that predicts at each time if the input time is a beat (1) or not (0). A peak-picking algorithm is then applied on the softmax output to detect the beats. This algorithm led to excellent results in the MIREX benchmark[15].

For the estimation of more high-level rhythm concepts such as the downbeat, which is considered to be the first beat of each bar, it is often necessary to rely on multiple representations (or features). For example in [DBDR17], four musical attributes contributing to the grouping of beats into a bar, namely harmony, rhythmic pattern, bass content, and melody are estimated by well designed representations which are in turn fed to parallel specific CNN.

Onset detection. An “onset” denotes the starting time of a musical event (pitched or non-pitched). Onset detection is the task of estimating the temporal positions of all onsets within a music track. The system proposed by [SB13] is a typical MIR DNN system. It uses a stack of convolution / max-pooling layers to progressively reduce the time and frequency dimensions and transfer those to the depth. It is then flattened and fed to a stack of FC layers with a sigmoid or a softmax output which perform the prediction. The novel idea proposed by [SB13] is to feed the network with chunks of spectrogram (each chunk represents 15 successive time frames of the spectrogram) and associate to it a single output y which represents the ground-truth for the middle frame of the chunk ($y = 1$ means that the middle frame of the chunk is an onset). These chunks can be considered as the “context windows” of [HES00] but benefit for the convolutional process. Contrary to the use of RNN, a music track is here processed as a bag of chunks which can be independently processed in parallel. The input to the network is made of the same three LMS computed with window sizes of 23.2 ms, 46.4 ms and 92.8 ms. This algorithm led to excellent results in the MIREX benchmark.

Music Structure: “Music Structure” denotes the global temporal organization of a music track into parts (such as intro, verse, chorus, bridge for popular music or movements for classical music). Music boundary detection is the task of estimating the transition times between these parts. To solve this, [SUG14] actually follow the same idea as for the onset detection [SB13]: a large temporal chunk is taken as input to a deep CNN which output predicts if the center frame of the chunk is a “music boundary” or not. However, here the input of the network is different: beside the LMS input a so-called

[15] MIREX (Music Information Retrieval Evaluation eXchange) is an annual evaluation campaign for MIR algorithms

Lag-Similarity-Matrix [Got03] is also used to better highlight the large-scale structure of the track. This path have been followed by [CHP17] leading to excellent results.

Dominant melody and multi-pitch estimation. Dominant melody refers to the temporal sequence of notes played by the dominant instrument in a music track (such as the singer in pop-music or the saxophone/trumpet in jazz music). Multi-pitch estimation refers to the estimation of the whole musical score (the temporal sequences of notes of each instrument). This is one of the most studied tasks in MIR. The DNN estimation methods proposed by [BMS+17] can be considered as a breakthrough. This method uses a Harmonic-CQT (already mentioned in part 3.3) as input to a deep CNN architecture which output is an image representing the pitch saliency of each time and frequency bins. The network is therefore trained to construct a "pitch saliency map" given an audio signal. A simple peak-picking or thresholding method can then be used to estimate the dominant melody or the multiple-pitches from this map.

Chord estimation. A chord is a set of multiple pitches that are heard as if sounding simultaneously. It is a convenient reduction of the harmonic content of a music track at a given time. Chords give rise to guitar-tabs which are largely used by guitarist or to real-book scores used by jazz players. Their estimation is both a segmentation task (finding the start and end time of each chord) and a labeling task (finding the correct chord label, such a C-Major, C7 or Cm7). Given its close relationship to speech recognition, the first chord estimation systems [SE03] relied on an acoustic model (usually a Gaussian Mixture Model (GMM) representation of Chroma features [Wak99]) connected to a language model (a hidden Markov model representing the chord transition rules specific to Western music[16]). [MB17] has proposed to solve the problem using a single DNN system. The specificity of this approach is to exploit the structural relationships between chord classes, i.e. the fact that while the label C-Major and Cm7 are different, their underlying chord construction share a large amount of notes. To do so, a CQT input is first encoded (using a convolutional-recurrent network architecture, i.e. a CNN followed by a bi-GRU) into the triplet of {root, pitches and bass} labels corresponding to the chord to be estimated. The outputs of those are then combined with the one of the encoder to estimate the final chord label. The authors show that constraining the training to learn the underlying structure of chords, allows increasing the chord recognition accuracy especially for the under-represented chord labels.

Auto-tagging. Auto-tagging is probably the most popular MIR task. It consists on estimating a set of tags to be applied to describe a music track. Such tags can relate to the track's music-genre (such as rock, pop, classical), mood (such as happy, sad, romantic), instrumentation (such as piano, trumpet, electric guitar) or in other descriptive information. Some tags can

[16] For example, the "II-V-I" (two-five-one) cadential chord progression is very common and particularly popular in jazz music

be mutually exclusive (such as singing/instrumental) some other not (such as piano and drum which may occur together). One of the most cited DNN system for auto-tagging is the one of [CFS16]. The system is a Fully Convolutional Network (no FC layer are used) inspired by the VGG-Net architecture [SZ15]: it is a stack of 2D convolution layers with small (3,3) kernels followed by max-pooling layers with small (2,4) kernels. This progressively transfers the time dimension of the input to the depth which is finally connected to 50 sigmoid outputs (multi-label classification task). Among the various input representations tested, the Mel-Spectrogram provides the best results. On the Magna-Tag-A-Tune dataset [LWM$^+$09], their approach outperforms any pre-existing systems. While being the most cited auto-tagging paper, this model has also been criticized by [Pon19] for its lack of consideration of the audio specificities. It is basically a computer vision network applied to an audio representation. Unexplainedly, it works very well.

Music recommendation by audio similarity. Music recommendation by audio similarity aims at recommending a ranked list of music tracks to a user. THe ranking is based on their audio similarity with a target music track. This kind of recommendation allows to get around the "cold start" problem[17]. To compute such an audio similarity, past approaches modelled the content of a track using generative models (such as GMM) of hand-crafted features (through MFCC). The audio similarity of two tracks was then computed as the Earth mover's distance - Kullback-Leibler divergence between their respective GMMs [APS05]. This approach was computationally expensive and did not allowed to reproduce a ground-truth ranked list. Recently [PRP20] have proposed to apply DNN metric learning to this problem. Starting from ground-truth ranked lists, they first define a set of ranked triplets *Tr*={*anchor, positive and negative*} using their relative positions in the ranked lists. Using those, a triplet loss [SKP15] is then used to train a CNN similar to [CFS16] (VGG-Net). It is fed with chunks of 512 CQT frames. The network learns to project each track in a 128-dimensions "audio-similarity embedding" space. In this, the similarity between two tracks is obtained as their Euclidean distance.

Cover detection. "Covers" denotes the various recorded interpretations of a musical composition (for example "Let It Be" performed by The Beatles or performed by Aretha Franklin). The problem has received a lot of attention recently due to the large amount of User-Generated Content which necessitates scalable copyright monitoring systems. While it is hard to define exactly why two tracks can be considered "covers" of each other, it is easy to provide examples and counter-examples of those. This is the approach proposed by [DP19, DP20, DYS$^+$20]. They propose to represent the content of a music track using jointly the CQT, the estimated dominant pitch and estimated multi-pitch representations. Those are fed to deep CNN networks. The networks are then trained using also a triplet loss paradigm [SKP15]

[17] when no meta-data (as used for tag-based recommendation) or usage data (as used in collaborative filtering recommendation) are available

using sets of anchor tracks, positive examples (covers of the anchors) and negative examples (non-covers of the anchors). The output of the networks are considered as track embeddings and it is shown that, once trained, the distance between the embedding of two tracks indicate their cover-ness. This algorithms has provide a large increase in cover-detection performances.

4.2 Environmental sounds description

The research field associated to the *Detection and classification of Acoustic Scene and Events (DCASE)* has received a steep growing interest with high industrial expectations. Similarly to other fields, recent progress in environmental sounds recognition has been largely fuelled by the emergence of Deep Neural Networks (DNN) frameworks [Abe20],[VPE17],[MHB$^+$18]. Nearly all the concepts and architectures described above have been used on specific DCASE problems such as *Urban scene analysis* (traffic events recognition, scene recognition, etc.), *bio-acoustic sounds recognition* (bird songs recognition, sea mammals identification, etc.) or *biological sounds* (deglutition, digestion, etc.).

However, the extreme diversity of potential sounds in natural soundscapes has favoured the development of specific methods which can more easily adapt to this variability. An interesting strategy is to rely on **feature learning approaches** which are proven to be more efficient than traditional time or time-frequency audio representations [SBER18]. Sparse representations, matrix factorizations and dictionary learning are some of the emblematic examples of this strategy. For example, some methods aim to decompose the audio scene recordings into a combination of basis components which can be obtained using **non-negative matrix factorization** (NMF)[BSER17] or shift-invariant probabilistic latent component analysis (SIPLCA) [BLD12]. In [BSER17], it was in particular shown that such a strategy when associated to DNN in a problem of acoustic scene classification allows to opt for simpler neural architectures and to use smaller amount of training data.

In terms of network structure and architectures, **Resnets** and shallow inception models have been shown to be particularly efficient on Acoustic source classification [SSL20] [MG20]. Resnets are specific networks in which each layer consists of a residual module and a skip connection bypassing this module [HZRS16]. It was recently shown that they can be interpreted as an ensemble of smaller networks which may be an explanation for their efficiency [VWB16].

For applications of predictive maintenance (anomalous sound detection), architectures based on auto-encoders are getting particularly popular due to their capacity to be learned in an unsupervised way. This is particularly interesting for this problem since there is usually a very low number of observations, if any, of the anomalous sounds to be detected [KSU$^+$19].

Another interesting avenue for environmental sound recognition is around approaches that are suitable for few-shot learning or transfer learning such as relation networks (prototypical networks [SSZ17] or Matching networks [VBL+16]). Matching networks use an attention mechanism over a learned latent space to predict classes for the unlabelled points and can be interpreted as a weighted nearest-neighbour classifier applied within an embedding space. In prototypical networks, the core idea is that there exists a latent space (e.g. embedding) described by a single prototype representation for each class. More precisely, a non-linear mapping of the input into an embedding space is learned using a neural network and takes a class's prototype to the mean of its support set in the embedding space. Classification can be performed for an embedded query point by simply finding the nearest class prototype. The capacity of prototypical networks to go beyond more straightforward transfer learning approaches and their efficacity for sound event recognition are shown in [PSS19].

4.3 Content processing: source separation

Blind Audio Source Separation (BASS) is the field of research dealing with the development of algorithms allowing the recovery of one or several source signals $s_j(t)$ from a given mixture signal $x(t) = \sum_j s_j(t)$ without any additional information (the separation is blind). It has close relationships with speech enhancement/denoising.

For a long time, BASS algorithms relied on the application of Computational Auditory Scene Analysis (CASA) principles [BC94] or matrix decomposition methods. Among the latter, Independent Component Analysis (ICA) assumes that the various sources are non-Gaussian and statistically independent; NMF factorizes the mixture's spectrogram as the product of a non-negative source activation matrix with a non-negative source basis matrix (see [PLDR18] for an overview on music source separation).

In recent years DNN methods for BASS has allowed to largely improved the separation quality. Most of the DNN methods consider the BASS problem as a supervised task: a DNN model is trained to transform an input mixed signal $x(t)$ to an output separated source $s_j(t)$ or to an output separation mask $m_j(t)$ to be applied to the input to get the separated source $s_j(t) = x(t) \odot m_j(t)$.

U-Net. Such a DNN model often takes the form of a Denoising Auto-Encoder (DAE) where a model is trained to reconstruct the clean signal from its noisy version. Because of their (theoretically) infinite memory, the first models used RNNs (or their LSTM and GRU variations) for both the encoder and decoder [MLO+12, WHLRS14, EHWLR15]. Since then, it has been demonstrated that non-recurrent architectures, such as CNN, can also be applied successfully at a much lower cost. However, convolutional DAE while successful for image denoising have been found limited for audio reconstruction (the bottleneck layer does not allow to capture the fine details necessary to reconstruct an harmonic spectrogram). To allow the reconstruction of these fine details, the U-Net architecture has been proposed. This architecture was first proposed for the segmentation of biomedical images [RFB15]. It is an AE with added skip connections between the encoder and the encoder to allow the reconstruction of the fine details. In [JHM$^+$17], this architecture has been applied to a spectrogram representation to isolate the singing voice from real polyphonic music largely improving previously obtained results. Precisely, the network is trained to output a Time/Frequency mask $M_j(t,f)$ such that applied to the amplitude STFT of the mixture $|X(t,f)|$, it allows to separate the amplitude STFT of the isolated source $|S_j(t,f)| = |X(t,f)| \odot M_j(t,f)$. The signal $s_j(t)$ is then reconstructed by inverting $|S_j(t,f)|$ using the phase of the initial mixture spectrogram $\phi_X(t,f)$. However, using the phase of the original signal limits the performances of the system.

Complex-U-Net. To deal with this limitation, [CKH$^+$19] have proposed in the case of speech enhancement to use the complex-spectrogram as input, and to modify the network, the masks and the loss to deal with complex values. In this case the complex-mask does not only modify the amplitudes $|X(t,f)|$ but also apply changes to the phases $\phi_X(t,f)$ so as to estimate the complex-spectrogram of the isolated source $S_j(t,f)$

Wave-U-Net. Another way to deal with the problem of the phase is to bypass the STFT and process the audio waveform directly. Along this, [SED18] have proposed a Wave-U-Net which applies the U-Net directly to the waveform. In this, the encoder is made of a cascade of 1D-convolution/Decimation to progressively reduce the time-dimension of $x(t)$ to the bottleneck representation z. A cascade of Up-Sampling/1D-convolution is then used to decode z in the separated signals $s_j(t)$ (no masking filters are used here).

End-to-end. [LPS19] also propose to use directly the waveform but without the U-Net architecture. The architecture is here inspired by WaveNet [vdODZ$^+$16] and uses a stack of dilated convolutions with skip connections but while WaveNet aims at predicting the next sample value, it is used here in a non-causal way to predict the set of isolated sources of the center frame.

SEGAN. SEGAN (Speech Enhancement Generative Adversarial Network) [PBS17] is an architecture proposed for speech enhancement which also uses the WaveNet blocks to represent the waveform. Moreover it also uses a DAE architecture but here considered as the generator G in a GAN set-up. The generator is trained to generate enhanced signals that look like real signals.

AE as NMF. [SV17] reconcile the DNN and the NMF source separation research community by expressing an AE as a non-linear NMF. In NMF a positive observed matrix X is reconstructed as the product of a positive basis-matrix W with a positive activation-matrix H: $\hat{X} = W \cdot H$. Similarly in an AE, X is reconstructed by passing z in the decoder function ϕ_d: $\hat{X} = \phi_d(z)$. Considering only one linear layer for ϕ_d would therefore make ϕ_d play the same role as W and z the same role as H. The encoder part $z = \phi_e(X)$ would then be $H = W^{\ddagger} \cdot X$[18]. They then propose a Non-Negative AE as a stack of non-linear encoding layers $Y_0 = X, Y_1 = g(W_1 \cdot Y_0), Y_2 = g(W_2 \cdot Y_1) \dots H = Y_L$ followed by a stack of non-linear decoding layers $Y_{L+1} = g(W_{L+1} \cdot Y_L) \dots \hat{X} = Y_{2L}$. g can be chosen to be a positive non-linear functions. The latent representation H can then be considered as an activation matrix which activate the "basis" of the decoder ϕ_d. Based on this, the authors propose various source separation algorithms.

TasNet, ConvTasNet. With this in mind, the seminal networks TasNet [LM18] and ConvTasNet [LM19] can also be considered as examples of an encoder which provides the activation's and a decoder which reconstruct the signal. However, both TasNet and ConvTasNet directly process the waveform using 1D-Convolution. The decoder ϕ_d reconstructs the mixture waveform as a non-negative weighted sum of basis signals $\vec{V}$: $\hat{\vec{x}} = \vec{w}\vec{V}$. The weights $\vec{w}$ are the outputs of a simple encoder ϕ_e of the form $\vec{w} = \mathcal{H}(\vec{x}\vec{U})$ where $\mathcal{H}$ is an optional nonlinear function[19]. The separation is done by masking the weights $\vec{w}$ and keeping only the ones necessary to reconstruct $\vec{s}_j$ from $\vec{x}$: $\hat{\vec{s}}_j = (\vec{w} \odot \vec{m}_j)$. The masks $\vec{m}_j$ are the outputs of a "separation network" ϕ_s: $\vec{m}_j = \phi_s(\vec{w}) \in [0, 1]$. The latter is a Deep-LSTM in TasNet or stacks of 1D-Conv for ConvTasNet. As opposed to the U-Net approaches described above [JHM$^+$17, CKH$^+$19, SED18] which apply the masks on the original mixture, the masks are here applied on the weights.

Deep Clustering. [HCLRW16] propose a very different paradigm to train a DNN architecture for source separation. Deep Clustering uses a metric learning approach. For this, a DNN is trained to non-linearly project each time and frequency points (t, f) of a spectrogram in a space such that points that belong to the same source (to different sources) are projected in close neighboring (far away respectively). A simple K-means clustering algorithm of the projected points can then be used to perform the separation.

4.4 Content generation

In statistical classification or machine learning, we often distinguish between discriminative or generative approaches [Jeb04]. Generative approaches are particularly attractive for their capacity to generate new data samples from their model. Some of the most popular models include different forms of autoencoders (including Variational Auto-Encoders (VAEs) [KW14b, CWBv19], Auto-Regressive models [vdODZ$^+$16, PVC19, VSP$^+$17] and Generative Adversarial Networks (GANs) [DMP18, GBC16]. These general models have sparked great interest since their introduction, mainly due

[18] ‡ denotes the pseudo-inverse

[19] for example a ReLU, to make the weights positive

to their incredible capabilities to generate new and high quality images [RMC16a, SGZ+16] but have also more recently shown their capacity for audio content generation.

Auto-regressive and Attention-based models As already discussed in section 3.3, *WaveNet* is clearly one of the most popular neural autoregressive generative models for audio waveform synthesis [vdODZ+16]. It is capable of high quality speech and music synthesis but remains a complex model with a demanding sample-level auto-regressive principle. Nevertheless it is used in many other frameworks and in particular in encoder-decoder architectures such as Nsynth [ERR+17] or Variational Auto-Encoders (VAEs) as further discussed below. Another trend in synthesis, initially introduced for Text-To-Speech (TTS), aims for fully end-to-end generative models, where the signal is directly synthesized from characters. For example, the original *Tacotron* relies on a sequence-to-sequence architecture with attention mechanism to generate a linear-scale spectrogram from which the audio signal can be estimated using Griffin and Lim algorithm [GJ84]. Its extension, *Tacotron2* [SPW+18], combines the advantage of both previous models in using a sequence-to-sequence Tacotron-style model to generate mel-scale spectrograms followed by a modified WaveNet synthesizer.

Variational Auto-Encoders: VAEs were used in speech synthesis as extensions of wavenet autoencoders where the quantized latent space is conditioned on the speaker identity [vdOVK17]. For music synthesis, a generalisation of the previous concept was proposed in [MWPT19] under the form of an universal music translation network. The main idea is to have a so-called universal encoder that forces the embeddings of all musical domains to lie in the same space but separate reconstructing decoders for each domain exploiting an auxiliary conditioning network. Several experiments of music domain conversion were described including for example early attempts for orchestral music to piano translation. The regularisation principle at the heart of VAEs can also be extended as in [ECRSB18] to enforce that the latent space exhibits the same topology as perceptual spaces such as musical timbre. One of the main advantages of such approaches is that the latent spaces can be directly used to synthesize sounds with continuous timbre evolution. Such capabilities can also be achieved with Generative Adversarial Networks (GANs) as discussed below with the example of drum synthesis [NLR20b]. Another extension of VAEs is known as the Vector-Quantized VAE (VQ-VAE)[vdOVK17] which aims at learning a discrete latent representation or *codebook*. The VQ-VAE can achieve sharper reconstructions than classic VAEs and can extract high-level interpretable audio features that strongly correlate with audio semantic information such as phonemes, with applications for voice conversion [CWBv19] or such as musical timbre for sound transformation. Another interesting approach in that framework is the Jukebox method presented in [DJP+20]. It is built on a multiscale VQ-VAEs (e.g. operating at different temporal resolutions) and on simplified autoregressive *transformers* with sparse attention. This model was in particular used to synthesize entire songs with vocals.

Adversarial audio synthesis: Generative Adversarial Networks (GANs) have been initially used with success in speech synthesis [STS18] but their use was rapidly extended to music synthesis. For exemple, WaveGan [DMP18] performs unsupervised synthesis of raw-waveform audio. WavGan is based upon the two-dimensional deep convolutional GAN (DCGAN) architecture initially developed for image synthesis [RMC16b] and adapted to audio in considering intrinsic differences between audio and images (which resulted in the use of larger receptive fields and higher upsampling factors between layers). As discussed above in section 3, a number of audio representations have been used in neural audio processing. For example in GANsynth [EAC$^+$19], several audio representations are evaluated including Short-Term Fourier Transform (STFT) representations (log Magnitude, wrapped and unwrapped Phase) and Instantaneous frequency (IF). Some other representations, including the raw audio waveform and a variety of time-frequency representations (such as complex spectrogram, CQT or MFCC), were also compared for the task of adversarial audio synthesis in [NLR20a].

Numerous extensions or adaptations of the concepts of GANs were proposed including Style-GAN [KLA19], Cycle-GAN [ZPIE17] or Progressive Growing GANs [AHPG18, KALL18]. In audio synthesis, for example, [NLR20b] proposed a specific Progressive Growing GAN architecture for drum sound synthesis with a conditional generation scheme using continuous perceptual features describing timbre (e.g., boominess, brightness, depth).

Music style transformations: Besides audio content generation, changing the style or instrumentation of a given piece of music is receiving a growing interest from the research community. Some research work target a direct style transformation of an input audio signal, as for example in [GDOP18] using Convolutive NN or as in the universal music translation network discussed above [MWPT19]. However, most studies operate on symbolic music such as MIDI and can focus on one or several music attributes such as melody [NSNY19], instrumentation or timbre [HCCY19, HLA$^+$19], accompaniment [CcR19, HSP16] or general arrangement style [BKWW18, LS18]. An interesting work at the crossroads of accompaniment generation and style transfer is the so-called Groove2Groove model [CSR20]. It is a one-shot style transfer encoder-decoder neural network method for symbolic music trained in a supervised fashion using synthetic parallel data. In this model, the input to the style translation model is a full accompaniment but the output is entirely regenerated and does not contain any of the original accompaniment tracks.

4.5 Semi-Supervised Learning and Self-Supervised Learning

Supervised learning assumes that labeled data, i.e. data x with associated ground-truth label y, are available to train the parameters θ of a prediction model $\hat{y} = f_\theta(x)$. To train a DNN model, the amount of such labeled data can be very large. While such large labeled datasets exist for image or speech, this is not the case today for audio content such as music or environmental sounds. We review here two popular techniques to deal with this lack of annotated data: semi-supervised learning (teacher-student paradigm) and self-supervised learning.

4.5.1 Semi-Supervised Learning

Semi-Supervised Learning (Semi-SL) combines training with a small amount of labeled data and training with a large amount of unlabeled data. One popular form of Semi-SL used the so-called teacher-student paradigm. It is a supervised learning technique in which the knowledge of a teacher (a model trained on clean labeled data) is used to label a large set of unlabeled data which is used in turn to train student models.

SoundNet [AVT16] is one of the first models developed in audio (for a task of environmental sounds recognition) that use the teacher-student technique. The idea is to transfer the knowledge of computer vision (CV) networks to an audio network. For this, a large set of Audio-Video clips are considered. Each clip has a video track and an audio-track. The CV networks are applied to the video-tracks to annotate the corresponding audio-tracks which are then used to train the audio network. The teachers are CV networks previously trained for objects and scenes recognition (an ImageNet CNN and a Places CNN) The audio network is a deep stack of 1-D convolutions. The transfer is done by minimizing the Kullback-Leibler divergence between the output probabilities of the audio and image networks. The training is done using two-millions unlabeled videos. It is shown that using such a trained audio network as feature extractor for typical Acoustic Scene Classification tasks largely outperform previous methods.

In music description, [WL17] were the first to use the teacher-student paradigm. For a task of drum transcription, a teacher (a Partially-Fixed-NMF model) previously trained on clean labeled data is applied on a large unlabeled dataset which is then used to train a DNN student model. The author show that the student largely outperforms the teacher. For a task of singing voice segmentation, [MBCHP18] also propose to use the teacher-student technique but in a different way. A teacher (a deep CNN network) previously trained on a clean but small labeled dataset, is applied to a large set of data grabbed from the web with labels obtained by crowd-sourcing (hence very noisy). The outputs of the teacher are then used to filter out the noise from the data. These cleaned data serve as the training label for the student. The author also report larger performances for the student.

4.6 Self-Supervised Learning

Self-Supervised Learning (Self-SL) is a supervised learning technique in which the training data are automatically labeled.

To automatically create labels, one can use the natural temporal synchronization between the various modalities of multi-media data. This is denoted by **Audio-Visual Correspondence (AVC)**. One of the first approach that use the AVC is the **"Look, Listen and Learn"** L^3 network [AZ17] where videos are decomposed into their image and audio modalities. A vision and an audio sub-networks are then fed to a fusion network with a softmax output which aims at predicting if the input image and audio correspond. Corresponding pairs are the ones taken at the same time from the same video, while mismatched pairs are extracted from different videos It is showed that the two image and audio sub-networks trained in such a way can be used afterward for solving sound classification or visual classification (ImageNet) tasks with very large performances.

The **AVE-Net**[AZ18] is an extension of the L^3 network in which the fusion network is replaced by a simple Euclidean distance. The sub-networks are therefore forced to learn to (non-linearly) project the data in a space where the image content (e.g. a guitar player) and its corresponding sound (e.g. a guitar sound) are projected nearby. Since both audio and video are projected in the same space, cross-modal applications are possible (such as querying an image giving a sound or the opposite) as well as visually locating the **"object that sounds"** (the sub-part of the image which projection is the closest to the projection of the sound).

In the same spirit, [ZGR+18] propose to train a two branches (image and audio) network for a task of source separation: to provide the **"sound of the pixels"** which are selected on the image. The audio branch (a U-Net) is trained to separate the audio into a set of isolated components. A Self-SL approach is then used to learn the mapping between each of these components and the various parts of the images.

Another type of Self-SL relies on applying transformations to an audio signal x for which we can predict the effect on the ground-truth labels y. The **SPICE** (Self-supervised Pitch Estimation) [GFR+20] network uses such an approach. In this, a Siamese AE is used. The encoder is first applied to the original audio to obtain a latent variable z_1. The signal is then pitch-transposed by a factor p and encoded to obtain z_2. The network is then trained to allow predicting p from the difference between z_1 and z_2. It is showed that, while trained to predict pitch-transposition, the network can be used to perform pitch-estimation with results very close to networks trained in a fully supervised way.

5 Conclusion and future directions

The advances in deep learning has strongly impacted the domain of audio analysis and synthesis. For many applications, the current state of the art is exploiting to at least some extent some form of deep neural processing. The emergence of deep neural networks as pure data-driven approaches was facilitated by the access to ever-increasing super-computing facilities, combined with the availability of huge data repositories (although largely unannotated). Nevertheless, this poses a number of challenges especially in terms of complexity, explainability, fairness and needs for data. We would like to sketch below some of our view for future directions in Deep learning for audio and music.

- **Increased explainability using Audio models**. For decades, many audio models have been developed. Such models include perceptual models (only audible information is modelled), Signal-based models (parametric models capturing the nature or structure of the signal) or physics-based (exploiting the knowledge of the sound production mechanisms or sound-propagation characteristics). Besides complexity reduction objectives, relying on appropriate audio source models within the deep architecture allows to constrain or "guide" the network to converge to an appropriate solution or to obtain more interpretable or explainable networks. Some recent works have already exploited some aspects of this view : using non-negative factorization models with CNNs for audio scene classification [BSER17], or for speech separation [LM19] or coupling signal processing modules with deep learning for audio synthesis [EHGR20, WTY20].
- **Increased performance and explainability using Multimodality.** In many situations, the audio signal can be associated with other modalities ranging from videos (e.g; in audiovisual scenes), text (such as lyrics or music scores), body movements or EEG (for example of subjects listening music). Video has proven to be useful for many audio tasks including for example audio-visual music performances analysis [DEL$^+$19] and audio-visual scene/object recognition but there are still important challenges especially when the modalities are not observed synchronously [PEO$^+$20]. As other examples, many Informed source separation approaches [OLBR13, LDDR13] do exploit an additional modality for separation such as lyrics for singing voice [SDRB19, LOD13, MBP20] score for music remixing [EM12], sketches on spectrogram representations for selective source separation [SM09], or EEG for attention-based music source separation [cER20]. There are clear interest to further exploit concurrent cues, when available, to build better and more explainable models.

- **Increased fairness and ethics.** If this is an obvious problem for the applications of Deep learning in health or justice, it is also of utmost importance in audio. In speech recognition, we certainly do not want systems that are more efficient on male voices than female voices. Similarly in music, since most of the studies are in western music, a clear bias towards this type of music exist. For music recommendation systems, fairness should also be a central goal to avoid bias in terms of gender, ethnicity or commercial inequity. In terms of content, to comply with ethics rules it becomes necessary to be able to filter unappropriate or explicit content [VHM+20].

References

[Abe20] Jakob Abeßer. A review of deep learning based methods for acoustic scene classification. *Applied Sciences*, 10, 03 2020.

[AHPG18] Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel, and Luc Van Gool. Towards high resolution video generation with progressive growing of sliced wasserstein gans, 2018.

[APS05] Jean-Julien Aucouturier, François Pachet, and Mark Sandler. The way it sounds : Timbre models for analysis and retrieval of polyphonic music signals. *IEEE Transactions of Multimedia*, 7(6):1028–1035, 2005.

[AVT16] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NIPS (Conference on Neural Information Processing Systems)*, 2016.

[AZ17] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proc. of IEEE ICCV (International Conference on Computer Vision)*, 2017.

[AZ18] Relja Arandjelović and Andrew Zisserman. Objects that sound. In *Proc. of ECCV (European Conference on Computer Vision)*, 2018.

[BC94] Guy J Brown and Martin Cooke. Computational auditory scene analysis. *Computer speech and language*, 8(4):297–336, 1994.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[BEP18] Dogac Basaran, Slim Essid, and Geoffroy Peeters. Main melody extraction with source-filter nmf and c-rnn. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Paris, France, September 23–27, 2018.

[BGL+94] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[BKK18] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[BKWW18] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer. In *ISMIR*, 2018.

[BLD12] Emmanouil Benetos, Mathieu Lagrange, and Simon Dixon. Characterisation of acoustic scenes using a temporally constrained shit-invariant model. *15th International Conference on Digital Audio Effects, DAFx 2012 Proceedings*, 09 2012.

[BM94] Hervé A. Bourlard and Nelson Morgan. *Connectionist Speech Recognition A Hybrid Approach*, volume 247. Springer US, 1994.

[BMS+17] Rachel Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan Pablo Bello. Deep salience representations for f0 estimation in polyphonic music. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Suzhou, China, October, 23–27 2017.

[Bro91] J. Brown. Calculation of a constant q spectral transform. *JASA (Journal of the Acoustical Society of America)*, 89(1):425–434, 1991.

[BS11] Sebastian Böck and Markus Schedl. Enhanced beat tracking with context-aware neural networks. In *Proc. of DAFx (International Conference on Digital Audio Effects)*, Paris, France, 2011.

[BSER17] V. Bisot, R. Serizel, S. Essid, and G. Richard. Feature learning with matrix factorization applied to acoustic scene classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(6):1216–1229, 2017.

[CcR19] Ondřej Cífka, Umut Şimşekli, and Gaël Richard. Supervised symbolic music style translation using synthetic data. In *ISMIR*, 2019.

[cER20] giorgia cantisani, Slim Essid, and Gael Richard. NEURO-STEERED MUSIC SOURCE SEPARATION WITH EEG-BASED AUDITORY ATTENTION DECODING AND CONTRASTIVE-NMF. working paper or preprint, October 2020.

[CFS16] Keunwoo Choi, György Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, New York, USA, 2016.

[CHP17] Alice Cohen-Hadria and Geoffroy Peeters. Music structure boundaries estimation using multiple self-similarity matrices as input depth of convolutional neural networks. In *AES Conference on Semantic Audio*, Erlangen, Germany, June, 22–24, 2017.

[CKH+19] Hyeong-Seok Choi, Jang-Hyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, and Kyogu Lee. Phase-aware speech enhancement with deep complex u-net. *Proc. of ICLR (International Conference on Learning Representations)*, 2019.

[CSR20] O. Cífka, U. Simsekli, and G. Richard. Groove2groove: One-shot music style transfer with supervision from synthetic data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2638–2650, 2020.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[CWBv19] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2041–2053, 2019.

[DBDR17] S. Durand, J. P. Bello, B. David, and G. Richard. Robust downbeat tracking using an ensemble of convolutional networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):76–89, 2017.

[DEL+19] Z. Duan, S. Essid, C. C. S. Liem, G. Richard, and G. Sharma. Audiovisual analysis of music performances: Overview of an emerging field. *IEEE Signal Processing Magazine*, 36(1):63–73, 2019.

[Die14] Sander Dieleman. Recommending music on spotify with deep learning. Technical report, http://benanne.github.io/2014/08/05/spotify-cnns.html, 2014.

[DJP+20] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music, 2020.

[DMP18] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

[DP19] Guillaume Doras and Geoffroy Peeters. Cover detection using dominant melody embeddings. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Delft, The Netherlands, November 4–8 2019.

[DP20] Guillaume Doras and Geoffroy Peeters. A prototypical triplet loss for cover detection. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Barcelona, Spain, May, 4–8 2020.

[DRDF10] Jean-Louis Durrieu, Gaël Richard, Bertrand David, and Cédric Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE transactions on audio, speech, and language processing*, 18(3):564–575, 2010.

[DS14] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968. IEEE, 2014.

[DYS$^+$20] Guillaume Doras, Furkan Yesiler, Joan Serra, Emilia Gomez, and Geoffroy Peeters. Combining musical features for cover detection. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Montreal, Canada, October, 11–15 2020.

[EAC$^+$19] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. In *Proc. of ICLR (International Conference on Learning Representations)*, 2019.

[ECRSB18] Philippe Esling, Axel Chemla-Romeu-Santos, and Adrien Bitton. Bridging audio analysis, perception and synthesis with perceptually-regularized variational timbre spaces. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, 2018.

[EHGR20] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. In *Proc. of ICLR (International Conference on Learning Representations)*, 2020.

[EHWLR15] Hakan Erdogan, John R Hershey, Shinji Watanabe, and Jonathan Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 708–712. IEEE, 2015.

[EM12] Sebastian Ewert and Meinard Müller. Score-Informed Source Separation for Music Signals. In Meinard Müller, Masataka Goto, and Markus Schedl, editors, *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 73–94. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2012.

[ERR$^+$17] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *Proc. of ICML (International Conference on Machine Learning)*, pages 1068–1077, 2017.

[FM82] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[FP19] Hadrien Foroughmand and Geoffroy Peeters. Deep-rhythm for global tempo estimation in music. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Delft, The Netherlands, November 4–8 2019.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[GDOP18] Eric Grinstein, Ngoc Q. K. Duong, Alexey Ozerov, and Patrick Pérez. Audio style transfer. In *ICASSP*, 2018.

[GFR$^+$20] Beat Gfeller, Christian Frank, Dominik Roblek, Matt Sharifi, Marco Tagliasacchi, and Mihajlo Velimirović. Spice: Self-supervised pitch estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1118–1128, 2020.

[GJ84] D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[GKKC07] Frantisek Grézl, Martin Karafiát, Stanislav Kontár, and Jan Cernocky. Probabilistic and bottle-neck features for lvcsr of meetings. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–757. IEEE, 2007.

[Got03] Masataka Goto. A chorus-section detecting method for musical audio signals. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, pages 437–440, Hong Kong, China, 2003.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[HBL12] Eric J. Humphrey, Juan Pablo Bello, and Yann LeCun. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Porto, Portugal, 2012.

[HCCY19] Yun-Ning Hung, I Ping Chiang, Yi-An Chen, and Yi-Hsuan Yang. Musical composition style transfer via disentangled timbre representations. In *IJCAI*, 2019.

[HCL06] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[HCLRW16] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2016.

[HDY+12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[HES00] Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma. Tandem connectionist feature extraction for conventional hmm systems. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1635–1638. IEEE, 2000.

[HLA+19] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, Sageev Oore, and Roger B. Grosse. TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) pipeline for musical timbre transfer. In *ICLR*, 2019.

[HOT06] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSP16] Gaëtan Hadjeres, Jason Sakellariou, and François Pachet. Style imitation and chord invention in polyphonic music with exponential families. *ArXiv*, abs/1609.05152, 2016.

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[Jeb04] T. Jebara. *Machine Learning: Discriminative and Generative.* 2004.

[JH11] Navdeep Jaitly and Geoffrey Hinton. Learning a better representation of speech soundwaves using restricted boltzmann machines. In *2011 IEEE Inter-*

national Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5884–5887. IEEE, 2011.

[JHM+17] Andreas Jansson, Eric J. Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Suzhou, China, October, 23–27 2017.

[KALL18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[KLN18] Taejun Kim, Jongpil Lee, and Juhan Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms. 2018.

[KLW19] Uday Kamath, John Liu, and James Whitaker. *Deep learning for nlp and speech recognition*, volume 84. Springer, 2019.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[KSU+19] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada. Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(1):212–224, 2019.

[KW14a] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. of ICLR (International Conference on Learning Representations)*, 2014.

[KW14b] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. of ICLR (International Conference on Learning Representations)*, 2014.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LC16] Vincent Lostanlen and Carmine-Emanuele Cella. Deep convolutional networks on the pitch spiral for music instrument recognition. *arXiv preprint arXiv:1605.06644*, 2016.

[LDDR13] A. Liutkus, J. Durrieu, L. Daudet, and G. Richard. An overview of informed audio source separation. In *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4, 2013.

[LM18] Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 696–700. IEEE, 2018.

[LM19] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266, 2019.

[LOD13] L. Le Magoarou, A. Ozerov, and N. Q. K. Duong. Text-informed audio source separation using nonnegative matrix partial co-factorization. In *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2013.

[LPKN17] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789*, 2017.

[LPLN09] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009.

[LPS19] Francesc Lluís, Jordi Pons, and Xavier Serra. End-to-end music source separation: is it possible in the waveform domain? In *Proc. of Interspeech*, Graz, Austria, September 15–19 2019.

[LS18] Wei-Tsung Lu and Li Su. Transferring the style of homophonic music using recurrent neural networks and autoregressive models. In *ISMIR*, 2018.

[LWM+09] Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie. Evaluation of algorithms using games: The case of music tagging. In *ISMIR*, pages 387–392, 2009.

[Mal89] Stephane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

[MB17] Brian McFee and Juan Pablo Bello. Structured training for large-vocabulary chord recognition. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Suzhou, China, October, 23–27 2017.

[MBCHP18] Gabriel Meseguer Brocal, Alice Cohen-Hadria, and Geoffroy Peeters. Dali: A large dataset of synchronized audio, lyrics and pitch, automatically created using teacher-student. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Paris, France, September, 23–27 2018.

[MBP20] Gabriel Meseguer Brocal and Geoffroy Peeters. Content based singing voice source separation via strong conditioning using aligned phonemes. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Montreal, Canada, October, 11–15 2020.

[MG20] M. D. McDonnell and W. Gao. Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 141–145, 2020.

[MHB+18] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley. Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(2):379–393, 2018.

[MKG+17] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. In *Proc. of ICLR (International Conference on Learning Representations)*, 2017.

[MLO+12] Andrew Maas, Quoc V Le, Tyler M O'neil, Oriol Vinyals, Patrick Nguyen, and Andrew Y Ng. Recurrent neural networks for noise reduction in robust asr. In *Proc. of Interspeech*, 2012.

[MWPT19] Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman. A universal music translation network. In *Proc. of ICLR (International Conference on Learning Representations)*, 2019.

[NLR20a] Javier Nistal, Stefan Lattner, and Gaël Richard. Comparing representations for audio synthesis using generative adversarial networks, 06 2020.

[NLR20b] Javier Nistal, Stephan Lattner, and Gaël Richard. Drumgan: Synthesis of drum sounds with timbral feature condition-ing using generative adversarial networks. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Montreal, Canada, October 2020.

[NPM20] Paul-Gauthier Noé, Titouan Parcollet, and Mohamed Morchid. Cgcnn: Complex gabor convolutional neural network on raw speech. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Barcelona, Spain, May, 4–8 2020.

[NSNY19] Eita Nakamura, Kentaro Shibata, Ryo Nishikimi, and Kazuyoshi Yoshii. Unsupervised melody style conversion. In *ICASSP*, 2019.

[OLBR13] A. Ozerov, A. Liutkus, R. Badeau, and G. Richard. Coding-based informed source separation: Nonnegative tensor factorization approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(8):1699–1712, 2013.

[PBS17] Santiago Pascual, Antonio Bonafonte, and Joan Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[PEO$^+$20] S. Parekh, S. Essid, A. Ozerov, N. Q. K. Duong, P. Pérez, and G. Richard. Weakly supervised representation learning for audio-visual scene analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:416–428, 2020.

[PLDR18] Bryan Pardo, Antoine Liutkus, Zhiyao Duan, and Gaël Richard. *Applying Source Separation to Music*, chapter 16, pages 345–376. John Wiley & Sons, Ltd, 2018.

[PLS16] Jordi Pons, Thomas Lidy, and Xavier Serra. Experimenting with musically motivated convolutional neural networks. In *Proc. of IEEE CBMI (International Workshop on Content-Based Multimedia Indexing)*, 2016.

[Pon19] Jordi Pons. *Deep neural networks for music and audio tagging.* PhD thesis, Music Technology Group (MTG), Universitat Pompeu Fabra, Barcelona, 2019.

[PRP20] Laure Pretet, Gaël Richard, and Geoffroy Peeters. Learning to rank music tracks using triplet loss. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Barcelona, Spain, May, 4–8 2020.

[PSS19] J. Pons, J. Serrà, and X. Serra. Training neural audio classifiers with few data. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16–20, 2019.

[PVC19] R. Prenger, R. Valle, and B. Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621, 2019.

[RB18] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028. IEEE, 2018.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[RMC16a] A. Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.

[RMC16b] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[Ros57] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para.* Cornell Aeronautical Laboratory, 1957.

[RSN13] G. Richard, S. Sundaram, and S. Narayanan. An overview on perceptually motivated audio indexing and classification. *Proceedings of the IEEE*, 101(9):1939–1954, 2013.

[Sai15] Tara N. Sainath. Towards end-to-end speech recognition using deep neural networks. In *Proc. of ICML (International Conference on Machine Learning)*, 2015.

[SB13] Jan Schlüter and Sebastian Böck. Musical onset detection with convolutional neural networks. In *6th International Workshop on Machine Learning and Music (MML) in conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, Prague, Czech Republic, 2013.

[SBER18] Romain Serizel, Victor Bisot, Slim Essid, and Gaël Richard. *Acoustic Features for Environmental Sound Analysis*, pages 71–101. 01 2018.

[SDRB19] K. Schulze-Forster, C. Doire, G. Richard, and R. Badeau. Weakly informed audio source separation. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 273–277, 2019.

[SE03] A. Sheh and Daniel P. W. Ellis. Chord segmentation and recognition using em-trained hidden markov models. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, pages 183–189, Baltimore, Maryland, USA, 2003.

[SED18] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Paris, France, September, 23–27 2018.

[SGZ$^+$16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc.

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. of IEEE CVPR (Conference on Computer Vision and Pattern Recognition)*, pages 815–823, 2015.

[SLJ$^+$15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[SM09] P. Smaragdis and G. J. Mysore. Separation by "humming": User-guided sound extraction from monophonic mixtures. In *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 69–72, 2009.

[SPW$^+$18] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, pages 4779–4783. IEEE, 2018.

[SS90] Xavier Serra and Julius Smith. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24, 1990.

[SSL20] Youngho Jeong Sangwon Suh, Sooyoung Park and Taejin Lee. Designing acoustic scene classification models with cnn variants. In *DCASE challenge, technical report*, 2020.

[SSZ17] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. 03 2017.

[STS18] Y. Saito, S. Takamichi, and H. Saruwatari. Statistical parametric speech synthesis incorporating generative adversarial networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):84–96, 2018.

[SUG14] Jan Schlüter, Karen Ullrich, and Thomas Grill. Structural segmentation with convolutional neural networks mirex submission. In *MIREX (Extended Abstract)*, Taipei, Taiwan, 2014.

[SV17] Paris Smaragdis and Shrikant Venkataramani. A neural network alternative to non-negative audio models. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, pages 86–90. IEEE, 2017.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[SVSS15] Tara N Sainath, Oriol Vinyals, Andrew Senior, and Haşim Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. IEEE, 2015.

[SWS+15] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform cldnns. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of ICLR (International Conference on Learning Representations)*, 2015.

[VBL+16] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. 06 2016.

[vdODZ+16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[vdOVK17] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc.

[VHM+20] Andrea Vaglio, Romain Hennequin, Manuel Moussallam, Gael Richard, and Florence d'Alché Buc. Audio-Based Detection of Explicit Content in Music. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 526–530, Barcelona, France, May 2020. IEEE.

[VPE17] Tuomas Virtanen, Mark Plumbley, and Dan Ellis. *Computational Analysis of Sound Scenes and Events.* 09 2017.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[VTBE15] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[VWB16] Andreas Veit, Michael J. Wilber, and Serge J. Belongie. Residual networks behave like ensembles of relatively shallow networks. In *NIPS*, 2016.

[Wak99] Gregory H. Wakefield. Mathematical representation of joint time-chroma distributions. In *Proc. of SPIE conference on Advanced Signal Processing Algorithms, Architecture and Implementations*, pages 637–645, Denver, Colorado, USA, 1999.

[WCNS20] M Won, S Chun, O Nieto, and X Serra. Data-driven harmonic filters for audio representation learning. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Barcelona, Spain, May, 4–8 2020.

[WHH+90] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. In *Readings in speech recognition*, pages 393–404. Elsevier, 1990.

[WHLRS14] Felix Weninger, John R Hershey, Jonathan Le Roux, and Björn Schuller. Discriminatively trained recurrent neural networks for single-channel speech separation. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 577–581. IEEE, 2014.

[WL17] Chih-Wei Wu and Alexander Lerch. Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data. In *Proc.*

of ISMIR (International Society for Music Information Retrieval), Suzhou, China, October, 23–27 2017.

[WTY20] X. Wang, S. Takaki, and J. Yamagishi. Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:402–415, 2020.

[ZEH16] Zhenyao Zhu, Jesse H Engel, and Awni Hannun. Learning multiscale features directly from waveforms. *arXiv preprint arXiv:1603.09509*, 2016.

[ZGR$^+$18] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 570–586, 2018.

[ZPIE17] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.