Master thesis in Sound and Music Computing

Epidemic Sound AB

Universitat Pompeu Fabra (Music Technology Group)

Uncovering underlying high-level musical content in the time domain

Leveraging self-supervised neural networks, inductive bias, and aural skills to learn deep audio embeddings with applications to boundary detection tasks

Oriol Colomé Font

Supervisors:

Carl Thomé

Carlos Lordelo

July 2023

Contents

1 Introduction **1**

1.1 Motivation 3

1.2 Objectives 5

1.3 Contributions 5

1.4 Assumptions 6

1.5 Music boundary detection as a downstream task 7

1.5.1 Related work 8

2 Methods **10**

2.1 Deep Learning modeling 10

2.2 Implementation details 12

2.2.1 Deep architecture design 13

2.2.2 Optimizer and learning rate 15

2.2.3 Audio augmentation and transformation pipeline 15

2.2.4 Loss function 19

2.2.5 Online triplet mining and batch normalization 20

2.2.6 The *embeddiogram*, a deep audio feature representation 22

3 Evaluation **26**

3.1 Datasets and metrics 26

3.1.1 The GTZAN dataset 26

3.1.2 The Million Song Dataset 26

3.1.3 The SALAMI dataset 27

3.1.4 Metrics 27

4 Results **30**

5 Conclusions **35**

5.1 Conclusions 35

5.2 Discussion 36

6 Future Work **37**

List of Figures **39**

List of Tables **40**

Bibliography **41**

Dedication

I dedicate this master's thesis to my closest family and friends, who have supported and inspired me throughout this journey. Their love and encouragement have motivated me to pursue my academic goals, and their belief in me has never faltered. I am grateful for their patience, understanding, and sacrifices as I pursued my studies.

Their presence in my life has made this journey meaningful, and this thesis is a tribute to their support.

Acknowledgement

I want to extend my deepest gratitude to...

- Carl Thomé and Carlos Lordelo, supervisors and fellow MIR researchers. Their expertise has been pivotal to the success of this thesis. Their guidance, supervision, and encouragement have enabled me to overcome the challenges encountered during this journey. Their commitment to mentorship is genuinely inspiring, and their influence has significantly shaped my academic path.

  Working with them has been an absolute privilege.

- Cody Hesse and Sebastian Löf, fellow interns at Epidemic Sound. Their generosity in sharing their time, knowledge, and insights has critically enhanced the quality of this research work.

- The Music Technology Group (MTG) and Universitat Pompeu Fabra (UPF). My profound thanks go to Xavier Serra for his faith in accepting a non-traditional profile into the master's program.

  I sincerely thank you for your integral roles in completing this thesis. I deeply value the knowledge and experience gained under their guidance and will cherish it always.

Contents

Music and science have always been my two greatest passions.

When the opportunity to work at Epidemic Sound presented itself, it was as if the universe had conspired to bring my interests together, allowing me to explore the fascinating Music Information Retrieval (MIR) field from a top-notch industrial perspective.

With the support and encouragement of my loved ones, I took a leap of faith and moved from Barcelona to Stockholm to embark on this exciting journey.

This work humbly aims to help untangle the complexities of music and offer a more musically-driven perspective to MIR. It aims to contribute to a deeper understanding of the intricate relationships within musical data by bridging the gap between music theory and computational analysis.

I am deeply grateful to the Epidemic Sound team for their guidance, expertise, and camaraderie throughout this project. I would also like to extend my heartfelt appreciation to my family for their unwavering support and belief in my abilities.

My motivation for writing this piece stems from a desire to grow as a musician, developer, and individual. I believe that by delving into the world of MIR, I can expand my horizons while making a meaningful contribution to the field.

This work is intended for any curious person, regardless of their background. I hope it will inspire others to explore the fascinating intersection of these two disciplines.

Abstract

This thesis posits the existence of invariant high-level musical concepts that persist regardless of changes in sonic qualities, akin to the symbolic domain where essence endures despite varying interpretations through different performances, instruments, and styles, among many other, almost countless variables.

In collaboration with Epidemic Sound AB and the Music Technology Group (MTG) at Universitat Pompeu Fabra (UPF), we used self-supervised contrastive learning to uncover the underlying structure of Western tonal music by learning deep audio features for music boundary detection. We applied deep convolutional neural networks with triplet loss function to identify abstract and semantic high-level musical elements without relying on their sonic qualities. This way, we replaced traditional acoustic features with deep audio embeddings, paving the way for sound-agnostic and content-sensitive music representation for downstream track segmentation tasks.

Our cognitively-based approach for learning embeddings focuses on using full-resolution data and preserving high-level musical information which unfolds in the time domain. A key component in our methodology is triplet networks, which effectively understand and preserve the nuanced relationships within musical data. Drawing upon our domain expertise, we developed robust transformations to encode heuristic musical concepts that should remain constant. This novel approach combines music and machine learning intending to enhance machine listening models' efficacy.

Preliminary results suggest that, while not outperforming state-of-the-art, our musically-informed technique has significant potential for boundary detection tasks. Most likely, so does for nearly all downstream sound-agnostic and content-sensitive tasks constrained by data scarcity, as it is possible to achieve competitive performance to traditional handcrafted signal processing methods by learning only from unlabeled audio files.

The question remains if such general-purpose audio representation can mimic human hearing.

Keywords: MIR; music structure analysis; boundary detection; deep audio embeddings; audio representations; representation learning; embeddings; transfer learning; multi-task learning; multi-modal learning; aural skills

Chapter 1

Introduction

Music [1], a cornerstone of human culture, has developed hand in hand with our societies since ancient times. However, its study presents unique challenges due to its subjectivity, fluctuating ground truth, and the vast array of styles and cultural contexts.

Prior research emphasizes the necessity of selecting suitable audio features to distinguish and label unique music segments for music segmentation tasks. The challenge lies in acquiring annotated data for these feature transformations, which can be time-consuming and expensive. In response, scholars have turned to unsupervised deep learning using readily available audio data [1, 2]. This method has substantially improved the generalization capacity of machine-listening models in downstream applications. Among others, segmentation algorithms have seen significant enhancements and delivered state-of-the-art results [3, 4].

Beyond applications in music structure analysis, the evolution of neural networks has empowered the creation of latent representations encapsulating crucial musical traits, technically known as representation —or feature— learning. Such spaces can be imploded into low dimensional and digestible representations, as shown in Figure

[1]Throughout this thesis, "music" refers to the Western tradition, defined by the conventions, practices, and aesthetics that primarily evolved in Europe and North America. This term's usage does not negate the diversity and richness of other musical traditions worldwide. Instead, it specifies this study's concentration on a particular cultural context.

1.

This evolution has simplified the computational process for various task-specific elements, including but not limited to:

- Creating an audio embedding that excels across numerous applications without the necessity for fine-tuning [5].
- Boosting the classification of environmental sounds [6].
- Enhancing vocal-centric music tasks through cross-domain audio embeddings [7].
- Integrating task-specific and pre-trained features to optimize audio classification [8].
- Developing a music similarity search engine for video production [9].
- Improving Music Emotion Recognition (MER) performance [10].
- Evaluating the effectiveness of speaker recognition via pre-trained model embeddings [11].
- Embedding songs for artist identification to facilitate similarity comparisons [12].
- Addressing cross-modal text-to-music retrieval issues [13].
- Enabling automated music rearrangement [14, 15].

Furthermore, deep audio embeddings offer the benefit of transferability, establishing a foundation for multiple tasks. Compared to training a model from scratch, this approach saves computational resources and time [16, 17, 18].

Unsupervised settings to learn those embeddings confront the persistent challenge of acquiring labeled data, an endeavor often time-consuming and expensive. By capitalizing on the vast volumes of available unlabeled music data, we bypass the need

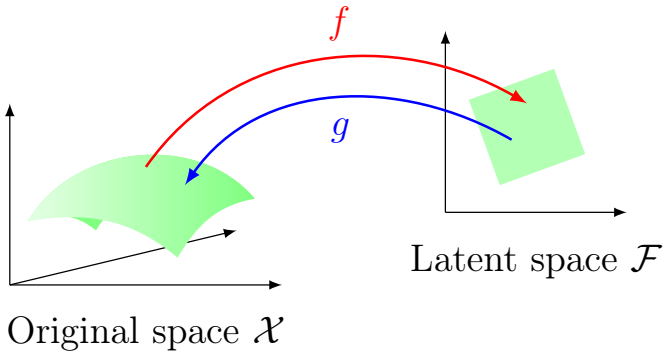

Figure 1: Dimensionality reduction and latent space representation: Mapping between the original high-dimensional space $\mathcal{X}$ and the lower-dimensional latent space $\mathcal{F}$ using functions f and g.

for laborious hand annotation, offering a pathway to potentially more comprehensive generalization.

It is yet to be determined whether a general-purpose audio representation can successfully emulate human hearing [5], even though some techniques have demonstrated generalizability across many music-understanding tasks [4, 19]. Deep learning's role in music understanding remains relatively nascent, with scarce work on deep music representations, a dearth of large-scale datasets, and a lack of a universal community-driven benchmark [5, 20].

All in all, given the proven effectiveness of deep audio embeddings in existing research and their potential for transfer learning, they hold considerable promise; therefore, we will leverage deep convolutional neural networks to replace traditional acoustic features with sound-agnostic and content-sensitive embeddings with applications for boundary detection and track segmentation tasks.

1.1 Motivation

This thesis, a collaboration between Epidemic Sound (ES) and the Music Technology Group (MTG) at Universitat Pompeu Fabra (UPF), has been driven by personal, industrial, and academic motivations.

ES, a Swedish company, curates a vast global library of over 40,000 royalty-free

music tracks and 90,000+ sound effects. Based in Barcelona, the MTG specializes in innovative sound and music technologies, such as information retrieval, digital signal processing, interactive music systems, and computational musicology.

The collaboration aims to deepen our understanding of music's fundamental structures, which could enhance ES's technical offerings and further advancements in MIR research.

As a musician, my passion for music has driven me to investigate the foundational elements of musical composition, various creation techniques, and their intricate relationships. I am dedicated to extracting valuable information embedded within music across all domains, focusing on audio and sheet music (symbolic domain). Musicians who endeavor to bridge the gap and uncover musical truths within the tonal paradigm, such as Heinrich Schenker [21], or those who challenge it, like Arnold Schoenberg [22], George Russell [23], and Ernst Levy [24], have been a continual source of inspiration. They aim to identify abstract concepts that reinforce or disrupt the tonal foundation, advancing the tonal landscape and providing a solid base for musicians' growth, development, and understanding of tonal and atonal paradigms.

Advancements in AI research, building on the theoretical groundwork laid by pioneers such as Alan Turing and Claude Shannon, have led to significant breakthroughs [25]. With AI being applied broadly in skyrocketing popular tech products [26], researchers and corporations must stay abreast of this rapidly evolving field.

The emergence of self-supervised models capable of learning embedding spaces to retrieve musical content from audio signals presents new opportunities. These models, which autonomously extract information from audio data, can potentially transform multiple aspects of the music industry. Industrially, these embedding spaces can be used to devise innovative products, enhance user experiences for content creators, and stimulate innovation and collaboration across the industry.

1.2 Objectives

Can deep audio embeddings be learned from reordering, scrambling, and augmenting sequences of musical information to improve unsupervised music boundary detection?

Our research aims to develop and learn a unified numerical understanding —or embedding [2]— for a piece of music that integrates the high-level relationships between elements as they unfold over time.

This approach aims to replicate human auditory capabilities by understanding and identifying abstract and semantic musical elements independent of their sonic qualities.

1.3 Contributions

Our study is centered around the following topics of investigation:

- We demonstrate that the size of the audio file dataset used for learning embeddings directly influences the results of music segmentation. Larger datasets lead to improved results.

- We show that achieving competitive performance with traditional handcrafted signal processing methods is possible by learning solely from unlabeled audio files.

- While our musically-informed technique does not currently surpass existing state-of-the-art baselines, it shows significant promise in boundary detection tasks, especially when the training set is expanded.

[2] A learned representation or embedding is a numerical output - usually a fixed-size vector - produced by a machine-learning model. Good non-supervised representations should be versatile across various tasks and require limited supervision, therefore appears as an attractive methodology to be employed [5].

1.4 Assumptions

This thesis posits that enduring, high-level musical elements persist regardless of sonic changes, mirroring the nature of the symbolic domain. A single waveform can correspond to a finite set of representations. Still, one sheet music excerpt can yield infinite interpretations.

The premise of our work is that regardless of the waveform or production style, abstract high-level features can be discerned, offering objective and thorough insight into a composition's musical content, analogous to analyzing sheet music for its basic behaviors as shown in Figure 2.

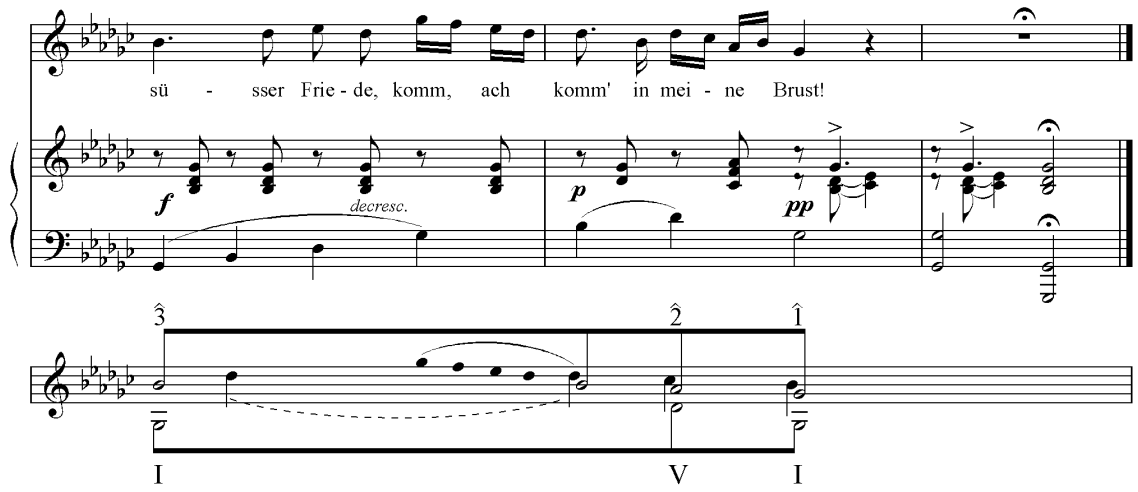

Figure 2: Small excerpt of *Wandrers Nachtlied, Op. 4, D. 224* by Franz Schubert. Passage's original score, the schenkerian unfolding of the melody, the chord degrees analysis, and their tonal function.

These characteristics aid listeners in recognizing and recreating musical structures, promoting deeper interaction with music. Mental shortcuts or heuristics enable a swift understanding of complex musical notions, revealing underlying patterns.

For example, despite differing cultural contexts, styles, time and key signatures, and sonic properties, we suggest a concept commonality between Figure 3 and Figure 4. Both pieces display the same 'musicological flavor' derived from their melodic and harmonic contour envelopes through non-diatonic major thirds.

This concept aligns with the theories of Kurt Koffka, a founding member of the

Gestalt school of psychology, who advocated for a holistic approach to understanding complex forms, as opposed to the structuralist practice of dissecting mental processes into their elemental components [27]. In the context of music, Gestalt psychology principles suggest that our minds process auditory input similarly to visual input by seeking patterns and structures. This understanding significantly enhances the comprehension of cognitive processes in musical perception and organization, influencing music theory, cognition, and therapy [28].

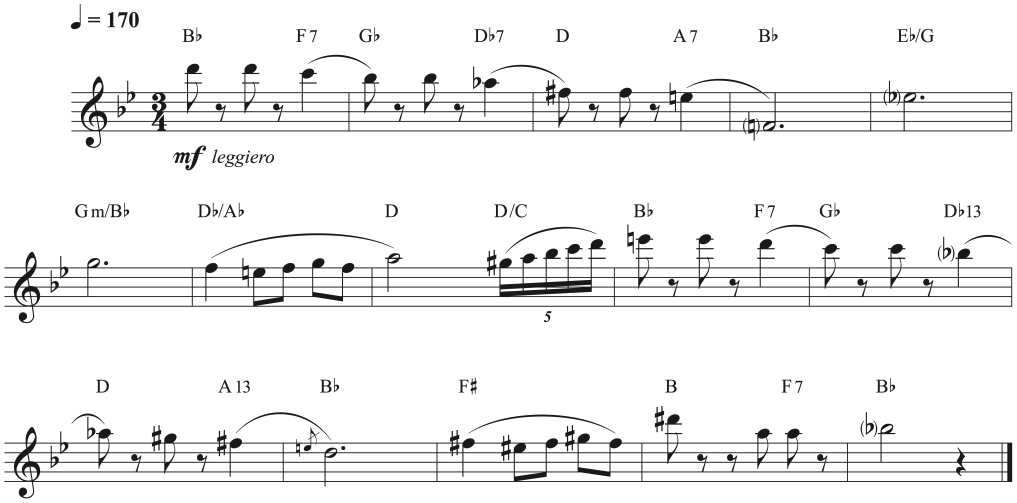

Figure 3: A small excerpt from Mahler's 9th Symphony, 2nd movement: The melodic and harmonic contour propels through non-diatonic major thirds.

1.5 Music boundary detection as a downstream task

While the usefulness of similar self-supervised learned embeddings can be evaluated in countless downstream tasks [4, 19], we have chosen music boundary detection. Also known as track segmentation and commonly tackled with spectral analysis, it is a subset of music structure analysis (MSA) suitable for its popularity [29], complexity, and product-compelling nature.

This interdisciplinary field aims to understand the structure of music [30]. Due to subjectivity, ambiguity, and data scarcity, audio-based MSA faces non-solved challenges like boundary placement ambiguity and similarity quantification [31].

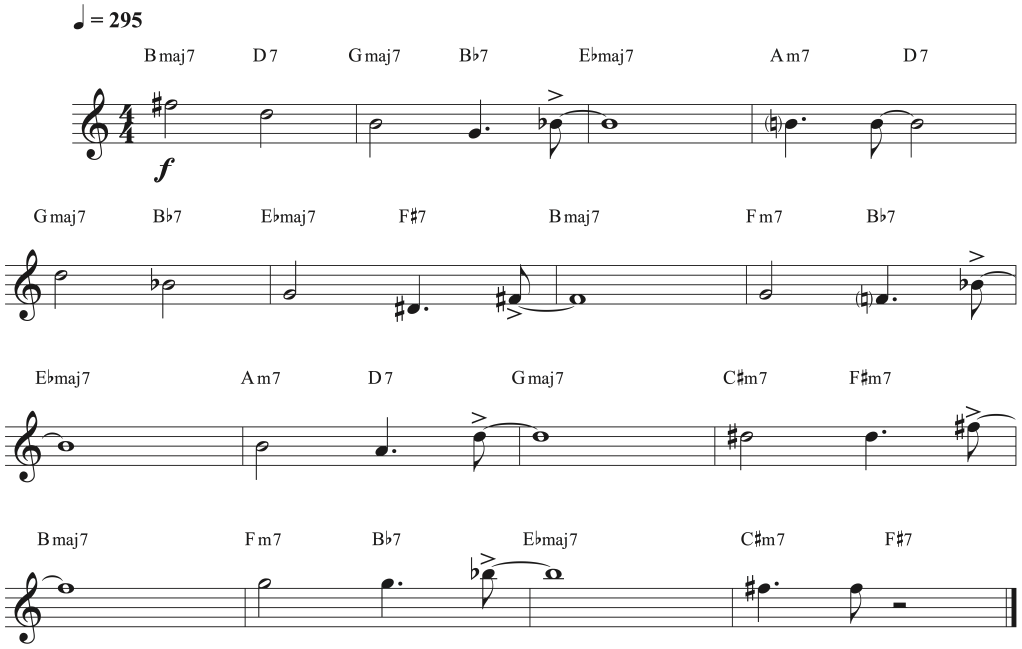

Figure 4: John Coltrane's Giant Steps head, featuring rapid chord changes in non-diatonic major thirds.

1.5.1 Related work

Related research investigates audio embeddings derived through unsupervised methods to enhance the performance of music segmentation algorithms [1, 32]. Both studies leverage the power of deep learning and data-driven feature learning, presenting advancements over traditional manually-engineered methods.

The study in [1] presents a novel approach for music segmentation, utilizing audio embeddings learned via few-shot learning and a music auto-tagging model. This method, which replaces the traditionally handcrafted MFCC and CQT features, significantly improves multi-level music segmentation, achieving state-of-the-art results and outperforming existing baselines.

While [1] approach and ours are very similar, there are some contrastive differences worth mentioning. These nuances stem from how much the approaches are task-tailored: the authors train their model on a dataset of audio features labeled with their corresponding music segments. They use a sampling strategy to create positive and negative examples likely from the same or different music segments. The method

by [1] requires music with clear beats and onsets. The sampling strategy relies on the fact that beats or onsets typically separate music segments. If the music does not have clear beats or onsets, it will be difficult for the method to create positive and negative examples likely to be from the same or different music segments.

Our approach is slightly different in two ways. First, we use physical time as input data rather than audio features. Second, our triplet selection pipeline is more abstract and designed for broader music understanding scope —yet still to be evaluated— rather than specifically tailored to music segmentation.

On the other hand, [32] explores using unsupervised deep learning methods for creating meaningful music representations and applying them to music structure analysis tasks. The research employs the same kind of deep architecture trained on millions of tracks from a music streaming service. The study reveals that these embeddings, derived via unsupervised learning, are effective at capturing musical structures, particularly when integrated with traditional feature engineering methods and a multi-level section fusion algorithm to merge short sections into longer ones.

Both studies emphasize the significant potential of deep learning-based feature learning in enhancing music segmentation tasks, which have traditionally relied on manual feature engineering methods only. Furthermore, they demonstrate how using deep learning techniques improves the performance of segmentation algorithms and enables a more nuanced and detailed analysis of music structure.

The results prove that unsupervised deep-learning techniques which derive audio embeddings offer more robust and efficient alternatives to exclusively traditional manual feature-engineering methods in music segmentation.

Chapter 2

Methods

This work employs unsupervised training of neural networks to extract meaningful features for music segmentation methodologies. As assertively stated by [1], this constitutes a logical evolution in applying contemporary machine learning techniques to music segmentation.

2.1 Deep Learning modeling

Deep learning (DL), a subset of machine learning (ML), employs deep neural networks (DNNs) to decipher hierarchical data representations, allowing models with multiple processing layers —as shown in Figure 6—to abstract data across numerous levels [33, 34]. DL models have significantly improved various domains due to their nonlinear modeling capabilities and scalability with large datasets.

They leverage the central algorithm, backpropagation, which calculates gradients of the loss function related to network parameters through forward and backward pass processes, thereby optimizing model performance. A high-level representation can be seen in Figure 5.

These computational models, somewhat rooted in biological neural systems, utilize layers of artificial neurons for tasks such as pattern recognition, classification, and regression. They evolve by adjusting connections between neurons during learning,

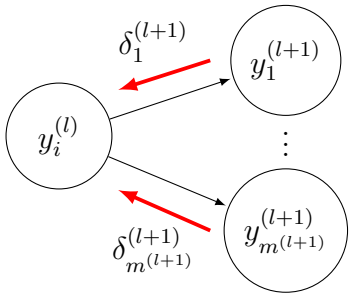

Figure 5: Backpropagation of errors through the network; once evaluated for all output units, the errors $\delta_i^{(L+1)}$ can be propagated backward.

a concept initiated in the mid-20th century but modernized with Frank Rosenblatt's single-layer perceptron in 1957, capable of learning linearly separable patterns [35].

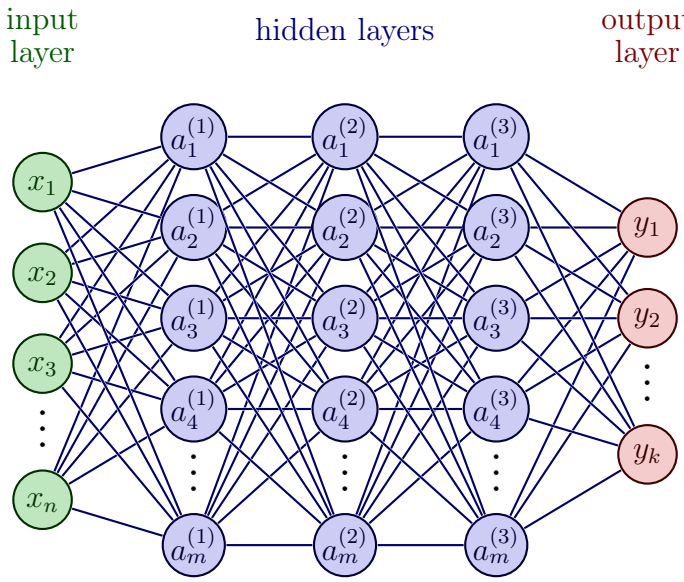

Figure 6: Network graph of a perceptron with D input units and C output units. The l^{th} hidden layer contains $m^{(l)}$ hidden units. Each neuron in a layer receives input from the previous layer and computes an output value using an activation function. The output of the last layer represents the prediction or classification result.

Another crucial component in the operation of DNNs is the activation function. This function is applied at every layer, transforming the linear combination of the input with the layer weights into an output passed onto the next layer. It introduces non-linearity into the network, enabling it to learn complex patterns and relationships. See Figure 7 for further visual understanding.

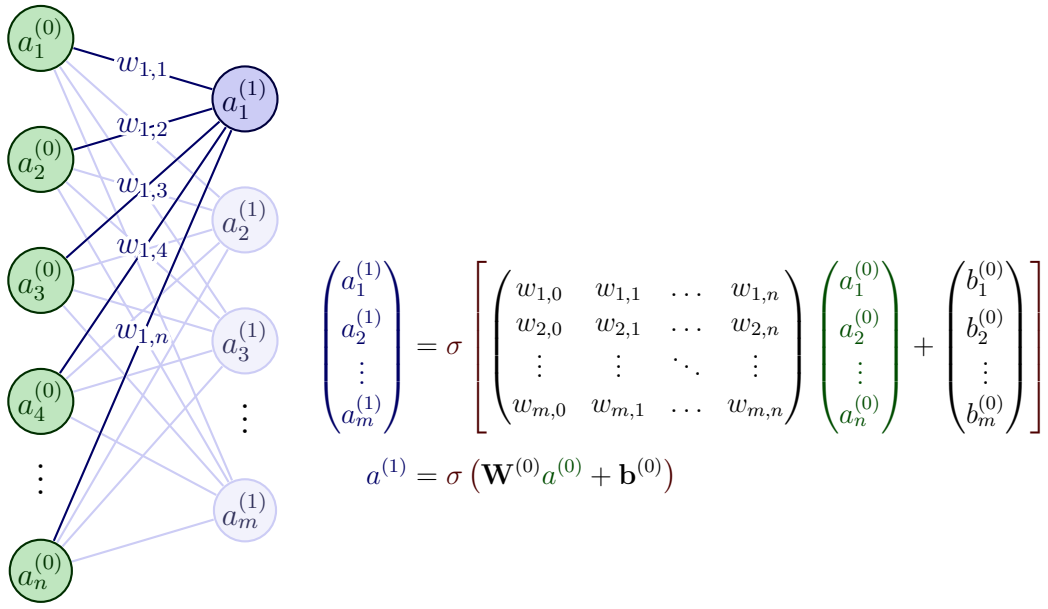

Figure 7: Input $a_i^{(0)}$ and output $a_j^{(1)}$ layers are connected by weight matrix $\mathbf{W}^{(0)}$ and bias vector $\mathbf{b}^{(0)}$, processed by activation function σ.

2.2 Implementation details

Our implementation falls broadly under cognitive modeling, which seeks to simulate human cognitive processes and problem-solving through a computerized model. We advocate for exposure-based learning in music, encouraging active engagement with as many musical styles, genres, techniques, and learning methods as possible. This approach promotes comprehensive musical proficiency and efficient performance when encountering novel and unseen data in practical applications.

According to Piaget's theory of cognitive development, children gain knowledge through sensory experiences and gradually develop abstract reasoning and schemas—basic cognitive structures [36]. These schemas evolve by incorporating new information through assimilation and accommodation [37].

In pattern recognition, models are designed to exhibit robustness against known invariances—transformations of input data —thereby ensuring consistent output. Even unknown invariances not explicitly considered in the model's design can be accommodated due to the model's inherent learning capacity.

Our implementation slightly modifies the Contrastive Learning of Music Representations (CLMR) method [38], which learns valuable, discriminative music representations without explicit labels by contrasting positive augmentations of a musical piece against negative ones. CLMR falls under a branch of ML known as self-supervised learning (SSL) [39], where models learn autonomously from unlabeled data to create their own supervisory signals [37]. This method resembles how humans learn from observations and interactions, transforming unsupervised problems into supervised ones by auto-generating labels. SSL benefits include reduced dependence on labeled data, contributing to more robust and generalizable data representations.

Although SSL has proven effective in speech and audio, its application to music audio remains relatively unexplored. This is primarily due to the unique challenges of modeling musical knowledge, especially regarding music's tonal and pitched characteristics [4].

2.2.1 Deep architecture design

This work suggests employing a Triplet Siamese Network (TSN), a model architecture known for its efficacy in music similarity retrieval tasks [12]. The aim is to minimize the loss function between a triplet of anchor, positive, and negative samples, utilizing online triplet mining for optimizing memory resources [40]. This SSL model aims to distinguish between similar and dissimilar samples effectively.

Introduced by Bromley and LeCun [41], Siamese Networks are DL architectures designed for tasks requiring comparison or similarity assessment between instances. The architecture comprises identical subnetworks that share the same parameters, improving memory usage and computational efficiency. Rather than learning specific features of individual classes, they focus on a similarity metric, making them ideal for imbalanced datasets. Each subnetwork processes an input independently, combining the outputs to yield a similarity score. Training with shared weights enables the model to learn invariant input representation, improving comparison efficiency. This is accomplished through a specialized loss function called triplet loss, which aims to minimize the distance between similar inputs and maximize it for dissimilar ones.

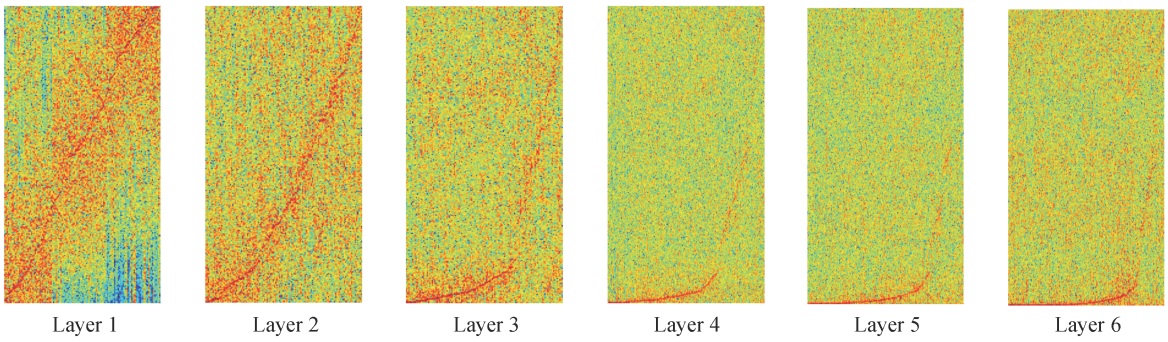

Figure 8: The spectrum of the estimated filters in the intermediate layers of SampleCNN is sorted by the frequency of the peak magnitude. The x-axis represents the index of the filter, and the y-axis represents the frequency range from 0 to 11 kHz. The model used for visualization is 39—SampleCNN with 59,049 samples as input. [42]

The loss function will be explained later in Subsection 2.2.4.

The TSN extends the traditional binary contrastive architecture by comparing three input instances instead of two. It strives to learn an embedding space where similar instances are closer and dissimilar ones are more distant.

Encoder: SampleCNN

The SampleCNN model [42] is a CNN designed for raw waveform audio data, treating each audio sample as an independent channel and applying 1-dimensional convolution along the temporal axis. Our implementation is an adaptation of [38] using *PyTorch* [43] and *PyTorch Lightning* [44].

With only 2.4 million trainable parameters, this fully convolutional model reduces computational requirements and learns features at different scales through its multi-resolution architecture. See layer specifications in Table 1.

The original model has been subtly adjusted by introducing an average pooling operation to the final convolutional layer. This modification is strategic for handling various input data sizes. Condensing the feature maps to a fixed-size output ensures consistency for subsequent layers like fully connected ones, streamlining computations while maintaining crucial information.

Layer Type	In Channels	Out Channels	Num. of layers
Conv + ReLU	1	128	1
Conv + BatchNorm + ReLU + MaxPool	128	128	2
	128	256	1
	256	256	6
	256	512	1
Conv + BatchNorm + ReLU + AvgPool	512	512	1
Fully connected layer	512	128	1

Table 1: Layer specifications for our SampleCNN model implementation; extracted and extended from [38].

2.2.2 Optimizer and learning rate

We've employed the AdamW optimizer [45], an optimized variant of the widely-used Adam optimizer for training neural networks. AdamW adeptly balances the learning rate across network weights, providing an efficient strategy for weight decay management by isolating it from gradient updates.

The learning rate, set at 0.003, is a pivotal parameter dictating the step size at each iteration towards loss function minimization. It's a delicate balancing act —a high rate promises swift convergence with a risk of minimum overshoot, while a lower rate provides careful convergence but necessitates more iterations. Given that delicacy, we set it to a standard number broadly used in the literature.

2.2.3 Audio augmentation and transformation pipeline

The choice of input data is guided by the task, computational resources, and the need to balance data retention with computational efficiency.

Previous research has utilized CNNs with various features such as Mel-Scaled Log-magnitude Spectograms (MLS), Self-Similarity Matrices (SSM), and Self-Similarity Lag Matrices (SSLM) as inputs [3]. However, features derived from raw audio may lack interpretability in some scenarios [46], and raw audio presents unique advantages despite being highly computationally demanding. It ensures the preservation

of the original signal, potentially uncovering novel insights, and allows for direct feature extraction via advanced DL models [47, 48]. Nevertheless, it comes with challenges, such as high —the highest— dimensionality requiring substantial computational resources.

Time-domain processing naturally handles temporal patterns and sequences in data, thereby avoiding windowing artifacts. Although audio feature-based methods are effective for various audio-related machine learning tasks, their limitations lie in representing perceptual similarity. MLS, for example, captures frequency distribution over time. Yet, the complexity of human auditory perception, encompassing temporal patterns, phase relationships between frequencies, and higher-level musical structures means that musically similar sounds can have distinct spectrograms. This discrepancy implies that using spectrogram distance alone for measuring high-level music content may not always align with human perceptions [19, 49].

In conclusion, raw audio waveforms were our final choice for input for the anchor. Each input sequence was 15 seconds long, roughly equivalent to 8 bars of 4/4 music at 120 beats per minute. The audio was sampled at a frequency of 16 kHz, a decision influenced by computational resources. We are confident that this choice of input contains a significant amount of meaningful musical information. Although it may not encompass the complexity of a symphony, it is sufficient for the scope of our experiments.

Positive sample generation chain

The positive waveform in every triplet of input data must preserve its intelligible content when subjected to transformations, regardless of alterations in sonic qualities and processing artifacts. Maintaining the temporal structure and meaningfulness of the content allows it to present musical elements remarkably close to the original track.

While we have experimented with helpful audio augmentation tools such as [50, 51], the specific requirements of our experiments required the development of our own transformation chain using *torchaudio*'s [52] implementation of *SoX* [53]: given an

anchor audio signal $A[n]$, we generate a positive signal $P[n]$ by applying a series of amplitude, time-domain, frequency-domain, modulation, reverberation, and nonlinear effects with additive noise on top of it.

Amplitude effects: The signal's amplitude is modified by a constant factor using gain $g \in [-12, 0]$.

Time-domain effects: The signal's playback speed and duration are altered through speed change and stretching, preserving the relative perceptual musical relationships between wave points. The respective factors are $\alpha \in [0.9, 1.1]$ for speed change and $\beta \in [0.9, 1.1]$ for stretch.

Frequency-domain effects: The frequency content is adjusted through pitch-shifting, modifying the pitch by $\Delta p \in [-1200, 1200]$ cents.

Nonlinear effects: Nonlinear distortion is introduced via overdrive with a parameter $d \in [0, 30]$.

Modulation effects: Utilize a control signal or low-frequency oscillator. Six variables determine the chorus parameters. The tremolo's amplitude modulation frequency and depth are controlled by $t_s \in [0.1, 100]$ and $t_d \in [1, 101]$, respectively.

Reverberation effects simulate a physical space's acoustic reflections and reverberations by applying an impulse response.

Noise effects: A noise signal $Noise[n]$ is added with a signal-to-noise ratio (SNR) in the range $[12, 100]$.

The positive signal $P[n]$ is generated by convoluting the various impulse responses of specific effects. The noise signals are included with a set SNR ratio within the above range on top of the effect chain.

Random parameter updates within hardcoded ranges generate unique audio $P[n]$ images out of $A[n]$ anchor for each run-through. Adding random white noise and varying SNR creates countless noisy waveform variations. Although possible combinations can be estimated by multiplying discrete parameter values, the presence

of continuous parameters and randomness in noise generation effectively results in infinite unique audio versions.

Negative sample generation

Circling back to our premises and assumptions, we posit that high-level musical content unfolds over time; therefore, we argue that the temporal structure of the negative images, in contrast to our anchor, must be disrupted. While maintaining similar sonic qualities, the content should be rendered unintelligible.

The computation for the negative signal $N[n]$ per every $A[n]$ goes as follows:

1. We first calculate the minimum and maximum audio chunk lengths in samples:

$$l_{min} = t_{min} \times S, \qquad l_{max} = t_{max} \times S \tag{2.1}$$

   The minimum duration t_{min} is set to 0.05 seconds, and the maximum duration t_{max} is set to 1 second. This range is chosen thoughtfully to strike a balance between two factors: on the one hand, it is above the just noticeable difference (JND) threshold, the smallest change in a stimulus that can be perceived. On the other hand, it is short enough to maintain a reasonably-sized window to avoid discernible musical content [54].

2. We then generate random audio chunk lengths $l_1, l_2, \ldots, l_{n-1}$ from the uniform distribution on the interval $[l_{min}, l_{max}]$. Calculate the final audio chunk length as:

$$l_n = L_A - \sum_{i=1}^{n-1} l_i \tag{2.2}$$

   where L_A is the length of the anchor signal in samples.

3. The third step is to split the anchor signal A into audio chunks $C_1, C_2, \ldots, C_n$ according to the calculated audio chunk lengths in the previous step.

4. Shuffle the audio chunks randomly to get the permuted slices $C_{\sigma(1)}, C_{\sigma(2)}, \ldots, C_{\sigma(n)}$, where σ is a random permutation of indices from 1 to n.

5. We finally concatenate the shuffled audio chunks to generate the negative signal that will have similar production while the content is completely ruined:

$$N[n] = C_{\sigma(1)} \oplus C_{\sigma(2)} \oplus \ldots \oplus C_{\sigma(n)} \tag{2.3}$$

The whole purpose of this process is to disturb the content unfolding in the time domain so it becomes musically unintelligible while maintaining the production and sonic attributes.

2.2.4 Loss function

Schroff, F., Kalenichenko, D., and Philbin, J. from Google first proposed and applied triplet loss for the learning of facial recognition, catering to varied poses and angles of the same individual [55].

Contrary to the widespread contrastive loss [56], the triplet loss function directs the learning process by minimizing the distance between the anchor and positive instances and maximizing the distance between the anchor and negative instances. Including a margin parameter in the loss function guarantees a minimum separation between positive and negative instances in the embedding space.

The triplet loss function $\mathcal{L}(\mathbf{a}, \mathbf{p}, \mathbf{n})$ aims to ensure that an anchor vector $\mathbf{a}_i$ is closer in the embedding space to a positive vector $\mathbf{p}_i$ (representing an example of the same class) than to a negative vector $\mathbf{n}_i$ (representing an example of a different class) by at least a margin α. It is calculated by summing the losses overall N triplets in the dataset, where the equation gives the loss for each triplet:

$$\mathcal{L}(\mathbf{a}, \mathbf{p}, \mathbf{n}) = \sum_{i=1}^{N} \max\left(0, |\mathbf{a}_i - \mathbf{p}_i|_2^2 - |\mathbf{a}_i - \mathbf{n}_i|_2^2 + \alpha\right) \tag{2.4}$$

The $\max(0, x)$ operation ensures zero loss when the distances satisfy this condition. The final loss used for model training is then the average loss over a mini-batch of N triplets:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathbf{a}_i, \mathbf{p}_i, \mathbf{n}_i) \tag{2.5}$$

The margin is a task-dependent optimal value determined empirically based on model performance. If it's too small, the model might not differentiate classes effectively; if it's too large, it might focus on outliers.

While some packages can be found in the MIR online community [57], we wrote our own *PyTorch* [43] implementation for the sake of our experiments.

As previously stated, the goal of minimizing this loss function is to learn discriminative embeddings, where similar examples are grouped closely together. In contrast, dissimilar examples are placed farther apart in the embedding space.

2.2.5 Online triplet mining and batch normalization

Online triplet mining is beneficial for managing large datasets by dynamically selecting the most informative triplets during training, focusing on each mini-batch. This strategy makes the process memory-efficient by negating the need to store all possible triplet combinations. Still, it also enhances model performance through quicker convergence by focusing on challenging examples based on the current model state.

This hard triplet mining selects triplets (a, p, n) to maximize the Euclidean distance between the anchor and positive samples and the anchor and negative samples. These distances, D_{AP} and D_{AN}, are computed respectively:

$$D_{\text{AP}} = \sqrt{\sum_i (A_i - P_i)^2} \qquad D_{\text{AN}} = \sqrt{\sum_i (A_i - N_i)^2} \tag{2.6}$$

In implementing the batch normalization step, it is necessary to standardize the audio lengths across all elements in the minibatch. We opted to zero-pad all clips to the length of the longest clip, valuing data integrity and completeness over potential

performance trade-offs. Thus, the length of the longest array in the batch, which sets the standard for all others, is as follows:

$$L_{\max} = \max_{i \in I} \left(\max \left(|A_i|, |P_i|, |N_i| \right) \right) \tag{2.7}$$

I represents the set of all items in the batch, $|A_i|$, $|P_i|$, and $|N_i|$ denote the lengths of the anchor, positive, and negative vectors for the i-th item, respectively. The max function is applied to find the longest of these three lengths for each item, and then the maximum of these maximum lengths is taken over all items in the batch. This gives the maximum length, $L_{\max}$, of any vector in the batch.

Hardware and training strategy

The deep learning models were trained on a high-performance cloud computing setup hosted on the Google Cloud Platform. The machine was of type `n1-standard-32`, equipped with an Intel Skylake processor and four NVIDIA Tesla T4 GPUs. The multiple GPUs allowed for efficient parallel processing, significantly reducing the training time.

To tackle the computational demands of training models on extensive raw audio data, we incorporated a couple of strategies to optimize efficiency and performance.

First, we utilized 16-mixed precision training. This approach leverages the improved performance of modern GPUs for 16-bit computations, enabling the model to run faster and use less memory without sacrificing model performance [58].

Secondly, to capitalize on the computational capabilities of multiple GPUs and hasten training times, we employed the Distributed Data-Parallel (DDP) strategy [59]. DDP operates on distinct mini-batches of data across GPUs and synchronizes the gradients after each backward pass, providing a more efficient scaling than other parallel strategies.

These strategies collectively enhance the computational efficiency while maintaining the robustness of the model training on lengthy raw audio data.

2.2.6 The *embeddiogram*, a deep audio feature representation

The *embeddiogram*, a deep audio feature representation, is derived by applying our pre-trained neural network to sliding windowed segments across the audio signal, generating a sequence of embedding vectors. These relatively low-dimensional vectors collectively form a two-dimensional description of the audio signal's musical content. A visual representation can be seen in Figure 9.

Below is a detailed explanation of how we compute the *embeddiogram* from a given audio signal of length N. This process comprises loading the audio data, slicing the audio data into windowed segments, processing each window using our pre-trained model to produce a vector per window/time frame, collecting and stacking these embeddings, and finally, normalizing the resulting matrix.

1. **Load the audio data**: The audio data is loaded into memory as a one-dimensional array of length N.

2. **Slice the audio data**: The audio data is segmented into overlapping windows. Each window contains w samples, and a hop size h separates consecutive windows. This gives a total of H windows, defined as:

$$H = 1 + \left\lfloor \frac{N - w}{h} \right\rfloor \tag{2.8}$$

   We have conducted experiments using a window duration of 4 seconds ($w = 4 \times$ sampling rate). As a general rule of thumb, this duration corresponds to two 4/4 bars of a piece of music with a tempo where the quarter note equals 120 BPM.

   We find this a good starting compromise solution, allowing enough time to capture musical content while not being so large that downsampling reduces the information to an unintelligible vector.

3. **Process each window**: Each window of audio data is processed independently, passed through the pre-trained neural network, and transformed into

an embedding vector. Formally, for each window w_i of audio data, we have:

$$\text{embedding}_i = \text{model}(w_i) \tag{2.9}$$

4. **Collect the embeddings**: The embedding vectors are collected and stacked together. Each row represents a feature vector for a given time frame to form the *embeddiogram*, denoted as E:

$$\text{E} = \begin{bmatrix} \text{embedding}_1 \\ \text{embedding}_2 \\ \vdots \\ \text{embedding}_H \end{bmatrix} \tag{2.10}$$

5. **Normalize the *embeddiogram***: The *embeddiogram* is normalized to have a minimum value of 0 and a maximum value of 1 per dimension. The normalization process is given by:

$$E'_{ij} = \frac{E_{ij} - \min(E)}{\max(E) - \min(E)} \tag{2.11}$$

These deep audio features can be a foundation for current state-of-the-art methods, which are expected to receive as input standard traditional audio features. Consequently, they can be processed and manipulated like conventional features as described by [60] and displayed in Figures 10, 11, 12, 13, and 14. Employing such embeddings in a traditional music segmentation algorithm can achieve state-of-the-art performance [1].

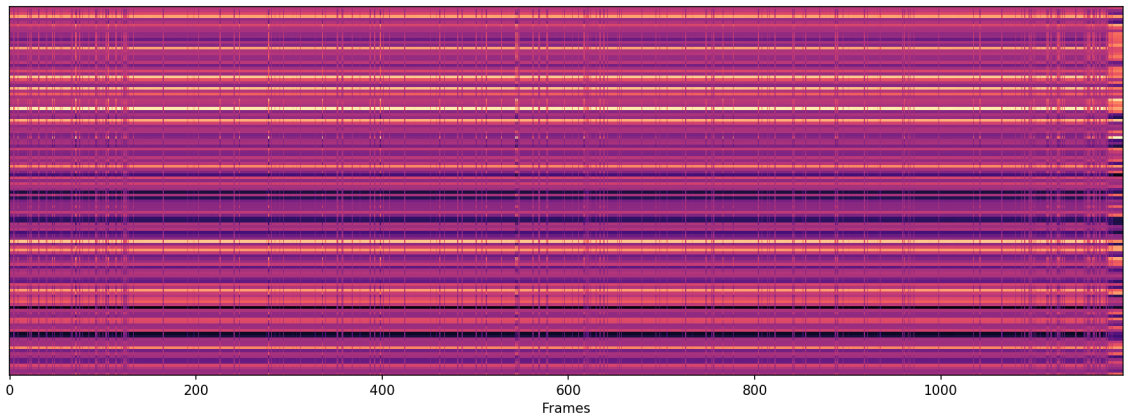

Figure 9: Embeddiogram. Track 355 (SALAMI dataset).

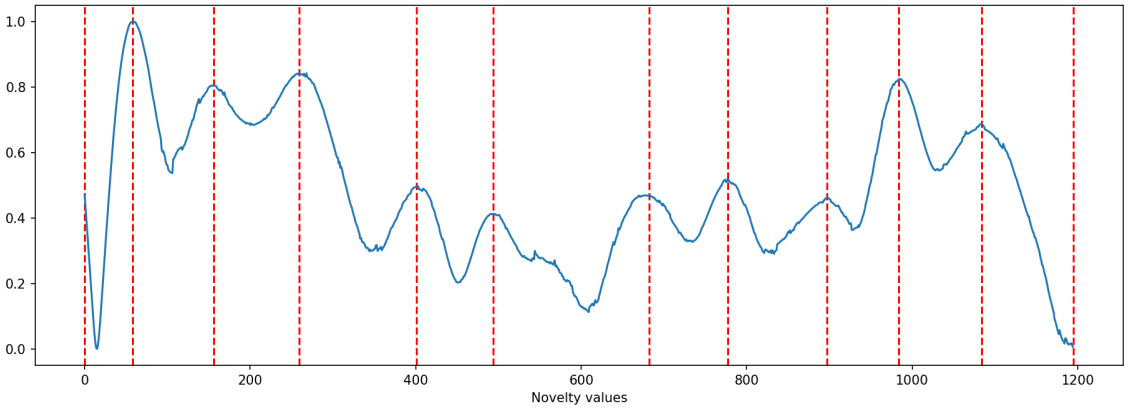

Figure 10: Novelty curve and peak detection. Track 355 (SALAMI dataset).

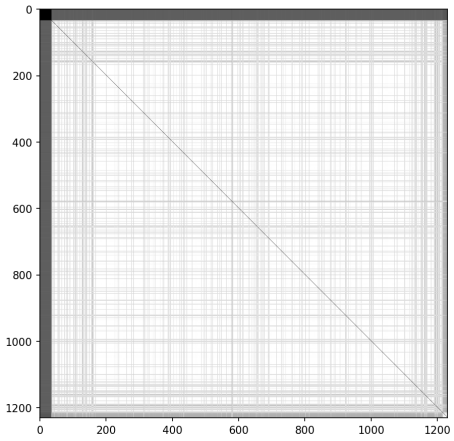

Figure 11: Self-similarity matrix computation of the embeddiogram. Track 355 (SALAMI dataset).

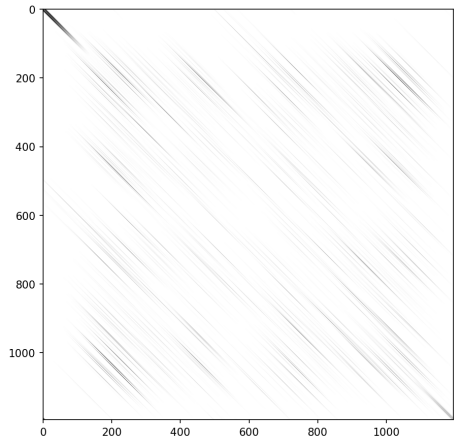

Figure 12: Self-similarity lag matrix. Track 355 (SALAMI dataset).

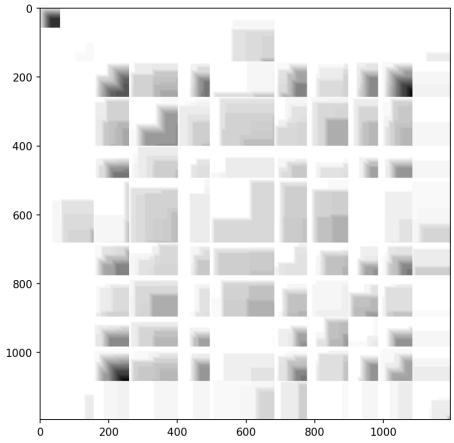

Figure 13: Cumulative matrix computation of the embeddiogram. Track 355 (SALAMI dataset) .

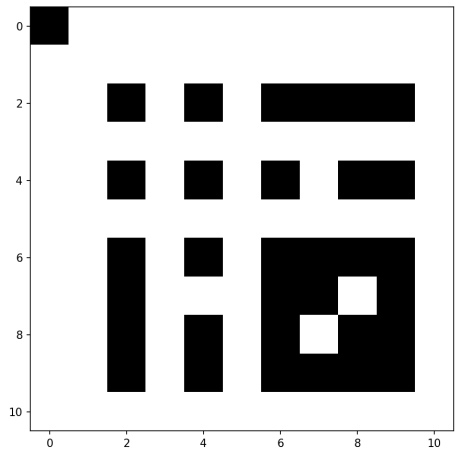

Figure 14: Transitive Binary Matrix computation of the embeddiogram. Track 355 (SALAMI dataset).

Chapter 3

Evaluation

3.1 Datasets and metrics

The developed model has been trained on GTZAN [61], the Million Song Dataset (MSD) [62] and evaluated on the SALAMI dataset [63].

3.1.1 The GTZAN dataset

The GTZAN Genre Collection is a widely-used dataset for music genre classification tasks. It comprises 1,000 audio tracks that are each 30 seconds long. These tracks are evenly distributed across ten genres: blues, classical, country, disco, hip hop, jazz, metal, pop, reggae, and rock, each containing precisely 100 songs. The audio files are stored as WAV files with a sample rate of 22,050 Hz [61].

3.1.2 The Million Song Dataset

The Million Song Dataset (MSD) is a publicly available audio and metadata collection for a million contemporary popular music tracks.

The MSD encourages research on large-scale recommendation systems, exploration of musicological properties, and general research on large datasets. It provides a massive scale and diversity of data, making it an excellent resource for complex,

large-scale music-related machine-learning tasks [62].

3.1.3 The SALAMI dataset

The Structural Analysis of Large Amounts of Music Information (SALAMI) project conducts extensive structural analyses on a wide variety of music. SALAMI segments music pieces into distinct sections, integrating different analyses, including perceptual, functional, and transcriptional. While there are some limitations, this approach offers a nuanced and thorough understanding of musical structure.

SALAMI covers an extensive range of music genres and styles, including but not limited to classical, jazz, popular, and world music. These pieces originate from diverse sources, including Codaich, the Internet Archive's Live Music Archive, the RWC Music Database, and the Isophonics database.

Each piece of music in SALAMI is accompanied by detailed metadata such as title, artist, duration, names of the annotators, and the time taken for the annotation process. This metadata is provided in multiple formats, catering to each source database's needs.

While SALAMI does not directly distribute audio, it directs users to corresponding audio files on streaming platforms. This makes it a valuable resource for researchers working on music structure analysis, genre classification, music summarization, and other related fields [63].

Note: We used SALAMI's original release from 2011, featuring 1,359 tracks.

3.1.4 Metrics

Three primary metrics have been utilized to assess the model performance: Precision, Recall, and F-measure.

Precision

Precision quantifies the proportion of accurately identified boundaries relative to all estimated boundaries to indicate the algorithm's accuracy in boundary detection.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.1}$$

Recall

On the other hand, recall measures the proportion of accurately detected boundaries against all reference boundaries, indicating the completeness or sensitivity of the algorithm.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.2}$$

F-measure

Lastly, the F-measure provides a harmonized measure of precision and recall. As a widely adopted metric for boundary detection, it compares predicted boundaries with ground truth ones, yielding a score between 0 and 1. This score, calculated as the harmonic mean of Precision and Recall, effectively accounts for both under-segmentation and over-segmentation. Given the inherent inaccuracies in human annotations and prediction errors, the F-measure allows for minor deviations between predicted and actual boundaries. The F-measure can adjust the tolerance threshold, permitting a predicted boundary to be considered correct if it falls within a predefined window of a ground truth boundary [31, 64]. In this work, we opted for an F-measure tolerance of 0.5 seconds, a decision largely guided by established norms in existing literature and music information retrieval research. This 0.5-second tolerance balances precision and flexibility, demanding accurate boundary predictions while accommodating slight variations inevitable due to the subjective nature of music segmentation.

We computed the F-measure for each track and then calculated the average rather than aggregating all tracks into a single dataset and calculating the F-measure on this combined set.

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.3}$$

Chapter 4

Results

As mentioned in Section 3.1, the evaluation results were obtained using standard and commonly used MIR tools, frameworks, and metrics. These include the MSAF [65] implementation for segmentation algorithms, the SALAMI dataset [63] for evaluation ground truth, and *mireval* Python package [66] for metric computation.

Table 2 provides a comparative analysis across four features using three segmentation algorithms. The results reveal that the performance of the algorithms varies depending on the feature type. For instance, the CQT feature exhibits the highest precision (0.570), recall (0.339), and f-measure (0.353) when processed with the Foote algorithm. Nevertheless, this table proves that our embeddiograms can achieve competitive performance compared to traditional handcrafted signal processing methods by learning only from unlabeled audio files.

These results are also represented in Figure 15 to understand the metrics' distribution for each feature visually. The high outliers suggest that there is significant variability in the performance of the model. While the model performs exceptionally well in some instances, it performs average in most cases.

Table 4 and Figure 16 compare the current study's boundary detection F-measure results with those of previous studies using unsupervised methods. The most accurate result was reported in 2019 by McCallum [1], who used a CNN on CQT and

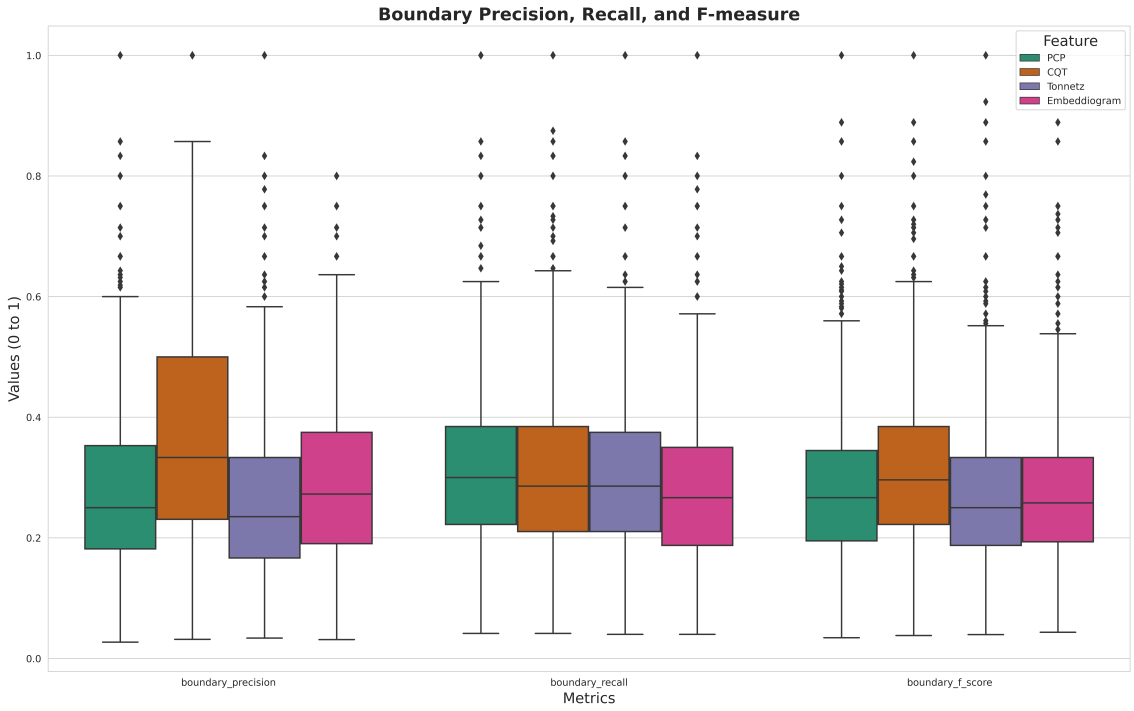

Figure 15: Boxplot visual comparison of different features' average precision, recall, and f-measure. Sliding windowed segments across the audio signal input signal is 4 seconds.

achieved an F-measure of 0.535. On the other hand, our study's approach yielded an F-measure of 0.288. Numbers show that our unsupervised method is competitive with research conducted a decade ago, trailing behind the current state of the art. However, this is a promising starting point, as the unsupervised nature offers ample and almost effortless opportunities for enhancement.

Feature	**SF**			**Foote**			**CNMF**		
PCP	0.311	0.324	0.305	0.288	**0.331**	0.295	0.228	0.310	0.250
Tonnetz	0.312	0.312	0.300	0.272	0.317	0.280	0.212	0.306	0.237
CQT	0.311	**0.339**	**0.312**	**0.570**	0.311	**0.353**	**0.296**	**0.311**	**0.287**
Embeddiogram	**0.333**	0.280	0.288	0.275	0.318	0.280	0.248	0.296	0.254

Table 2: Comparison of precision (left column), recall (middle column), and f-measure (right column) metrics for different features using the Structural Feature (SF)[67], Checkerboard-like Kernel (Foote) [68], and Convex Non-negative Matrix Factorization (CNMF) [69] algorithms on the SALAMI dataset. The sliding windowed segments across the input signal is 4 seconds long.

Table 3 compares the performance of the Structural Feature (SF) algorithm on the SALAMI dataset, utilizing two distinct sets of embeddiograms as input features.

Training dataset [Ref]	**Precision**	**Recall**	**F-measure**
GTZAN [61]	0.228	0.171	0.185
MSD [62]	**0.333**	**0.280**	**0.288**

Table 3: Comparison of precision, recall, and F-measure for GTZAN-trained versus MSD-trained embeddiograms on SALAMI dataset computed using the Structural Feature (SF)[67] algorithm.

These embeddiograms were generated with the neural network trained using the GTZAN and MSD datasets, respectively, and the results show a clear advantage when trained on the MSD dataset compared to the GTZAN dataset. Specifically, the precision, recall, and f-measure are all higher for the MSD-trained algorithm.

The GitHub repository containing all the code needed to run the experiments can be found HERE.

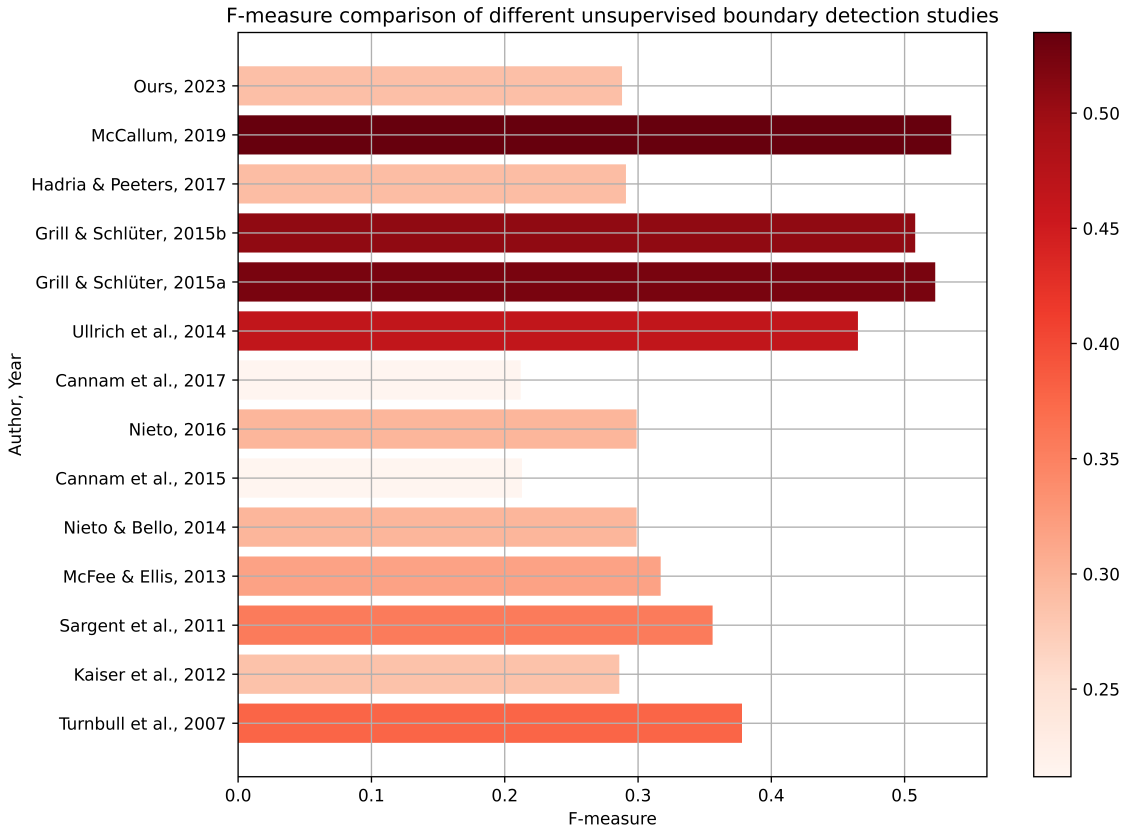

Figure 16: Previous studies' boundary detection f-measure results using unsupervised methods for a 0.5s time-window tolerance. Only the top-performing algorithm for each year on the SALAMI dataset is displayed. This figure has been extended from [3].

Authors [Ref], Year	**Input**[1]	**Method**	**F-measure**
Turnbull et al. [64], 2007	MFCCs, chromas, spectrogram	Boosted Decision Stump	0.378
Kaiser et al. [70], 2012	SSM	Novelty measure	0.286
Sargent et al. [71], 2011	MFCCs, chromas	Viterbi	0.356
McFee & Ellis [72], 2013	MLS	Fisher's Linear Discriminant	0.317
Nieto & Bello [73], 2014	MFCCs, chromas	Checkerboard-like kernel	0.299
Cannam et al. [74], 2015	Timbre-type histograms	HMM	0.213
Nieto [75], 2016	CQT Spectrogram	Linear Discriminant Analysis	0.299
Cannam et al. [74], 2017	Timbre-type histograms	HMM	0.212
Ullrich et. al [76], 2014	MLS	CNN	0.465
Grill & Schlüter [77], 2015	MLS, SSLMs	CNN	0.523
Grill & Schlüter [2], 2015	MLS, PCPs, SSLMs	CNN	0.508
Hadria & Peeterss [78], 2017	MLS, SSLMs	CNN	0.291
McCallum [1], 2019	CQT	CNN	**0.535**
Ours, 2023	Raw waveforms	CNN	0.288

Table 4: Previous studies' boundary detection f-measure results using unsupervised methods for a 0.5s time-window tolerance. Only the top-performing algorithm for each year on the SALAMI dataset is displayed. This table has been extended from [3].

[1] **Legend:** SSM: Self-Similarity Matrix, MLS: Mel Spectrogram, MFCC: Mel-Frequency Cepstral Coefficient, CQT: Constant Q-Transform, PCP: Pitch Class Profile, SSLM: Self-Similarity Lag Matrix.

Chapter 5

Conclusions

5.1 Conclusions

In this work, we introduced a method that leverages self-supervised deep neural networks to learn low-dimensional music latent representations with applications to music boundary detection tasks. Building on existing approaches and architectures, we replaced traditional features with deep embeddings trained to represent high-level musical information to analyze its performance in music segmentation tasks.

While our musically-informed technique does not yet outperform the existing state-of-the-art baselines, it exhibits significant potential in boundary detection tasks, particularly when expanding the training set. The improvement we've observed between datasets is noteworthy, and when compared with two of the acoustic features, our results are highly competitive.

We have also managed to circumvent the typical issues associated with dataset enlargements, such as the need for extensive supervision or human annotation, which gives our method an edge in terms of practicality and scalability, effectively turning what is usually seen as an expensive hurdle into a much more manageable task.

5.2 Discussion

One of the main reasons why [1, 32] might outperform our method might be their task-tailored and MSA-focused designs specifically oriented towards music segmentation. In contrast, our method aims to be broader and more abstract, which may present a downside when evaluating specific tasks.

Whether our model effectively decodes underlying high-level musical content remains open to scientific investigation. It's plausible that our technique possesses significant potential for nearly all content-based MIR downstream tasks, given its intention to be both sound-agnostic and content-sensitive. Queries persist about whether such a general-purpose audio representation can mimic human hearing [4, 5], or if it can accurately decode high-level musical content. Such questions remain unresolved due to the current lack of evaluative measures.

We argue whether factors such as the size of the representation layer and the size of the model [48] might be insufficient. Furthermore, we posit that the loss function could decrease further with additional time to iterate repeatedly over the stochastic and never-ending dataset.

The proposed embeddiograms are a promising new approach for music boundary detection. While they do not yet rival state-of-the-art methods, they are competitive with traditional handcrafted signal processing methods and can be trained on unlabeled audio files. This makes them a cost-effective and scalable solution for music boundary detection tasks.

Chapter 6

Future Work

This research merely scratches the surface, and vast territories are yet uncharted. In future studies, the following areas will be explored to extend the current research further:

- Investigate the effect of different transformations and augmentation pipelines. Incorporating track stems and different takes of the same piece as natural, human-made augmentations could yield exciting results.
- Expand input data to include dB-scale Mel-spectrum magnitude and CQT of audio. This approach has been widely used in music-related tasks with CNNs [1, 19]. Though raw audio provides a rich representation, dB-scale Mel-spectrum offers a frequency-domain summarization that is not only grounded in psycho-acoustics but is also computationally efficient and hard to reproduce solely through data-driven methods. Therefore, this trade-off is worth exploring.
- Implement k-fold cross-validation to improve the robustness of the model's performance.
- Experiment with different hyperparameters. For instance, setting the kernel size to 0.005 times the sample rate could match the Just Noticeable Difference

(JND). As for the loss function margin, a starting point of 0.2 has proven effective, but different values should be explored to optimize performance on the validation set.

- Increase the size of the representation layer to 512 or 1024 dimensions.
- Apply easy triplet mining to improve the model's performance [79].
- Implement visual and auditory evaluations: 2D or 3D visualization of the latent space as displayed in Figure 1, coupled with a Graphical User Interface (GUI) that enables playback for evaluation, can help assess the extent to which the model considers sonic attributes. It would also facilitate the understanding of the clustering of complex musical content.

This represents a 'thorn in our side' that we intend to address in future research.

List of Figures

1 Dimensionality reduction and latent space representation [80]. 3
2 Excerpt of *Wandrers Nachtlied, Op. 4, D. 224* by Franz Schubert. . . 6
3 Mahler's 9th Symphony, 2nd movement excerpt. 7
4 Giant Steps excerpt. 8

5 Backpropagation. 11
6 Neural network graph [80]. 11
7 Activation function [80]. 12
8 SampleCNN filters 14
9 Embeddiogram | Track 355 (SALAMI dataset). 24
10 Novelty curve and peaks | Track 355 (SALAMI dataset). 24
11 Self-similarity matrix | Track 355 (SALAMI dataset) 25
12 Self-similarity lag matrix | Track 355 (SALAMI dataset) 25
13 Q-matrix | Track 355 (SALAMI dataset) 25
14 Transitive Binary Similarity Matrix | Track 355 (SALAMI dataset) . 25

15 Metric comparison for different audio features. 31
16 Baseline. State-of-the-art graph. 33

List of Tables

1 SampleCNN layer specifications 15

2 Metric comparison: audio features and segmentation algorithms . . . 31

3 GTZAN-trained versus MSD-trained embeddiograms 32

4 Baseline. State-of-the-art comparison table. 34

Bibliography

[1] McCallum, M. C. Unsupervised learning of deep features for music segmentation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 346–350 (2019).

[2] Grill, T. & Schï, J. Music boundary detection using neural networks on combined features and two-level annotations. Tech. Rep. (2015). URL `http://www.ofai.at/research/impml/`.

[3] Hernandez-Olivan, C., Beltran, J. R. & Diaz-Guerra, D. Music boundary detection using convolutional neural networks: A comparative analysis of combined input features. *International Journal of Interactive Multimedia and Artificial Intelligence* **7**, 78–88 (2021).

[4] Li, Y. *et al.* MERT: Acoustic music understanding model with large-scale self-supervised training (2023). URL `http://arxiv.org/abs/2306.00107`.

[5] Turian, J. *et al.* HEAR: Holistic evaluation of audio representations (2022). URL `http://arxiv.org/abs/2203.03022`.

[6] Cramer, J., Wu, H.-H., Salamon, J. & Bello, J. P. Look, listen, and learn more: design choices for deep audio embeddings. Tech. Rep. URL `https://github.com/marl/l3embedding`.

[7] Kim, K. L., Lee, J., Kum, S. & Nam, J. Learning a cross-domain embedding space of vocal and mixed audio with a structure-preserving triplet loss. Tech. Rep. (2021).

[8] Hung, Y.-N. & Lerch, A. Feature-informed embedding space regularization for audio classification (2022). URL http://arxiv.org/abs/2206.04850.

[9] Thomé, C., Piwell, S. & Utterbäck, O. Musical audio similarity with self-supervised convolutional neural networks (2022). URL http://arxiv.org/abs/2202.02112.

[10] Koh, E. & Dubnov, S. Comparison and analysis of deep audio embeddings for music emotion recognition. Tech. Rep. URL https://openl3.readthedocs.io/en/latest/index.html.

[11] Phukan, O. C., Buduru, A. B. & Sharma, R. Transforming the Embeddings: A lightweight technique for speech emotion recognition tasks (2023). URL http://arxiv.org/abs/2305.18640.

[12] Cleveland, J., Cheng, D., Zhou, M., Joachims, T. & Turnbull, D. Content-based music similarity with triplet networks (2020). URL http://arxiv.org/abs/2008.04938.

[13] Won, M., Salamon, J., Bryan, N. J., Mysore, G. J. & Serra, X. Emotion embedding spaces for matching music to stories. Tech. Rep. URL https://multimediaeval.github.io/2020-Emotion-and-Theme-.

[14] Stoller, D., Vatolkin, I. & Müller, H. Intuitive and efficient computer aided music rearrangement with optimised processing of audio transitions. Tech. Rep. (2018).

[15] Plachouras, C. & Miron, M. Music rearrangement using hierarchical segmentation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5 (IEEE, 2023). URL https://ieeexplore.ieee.org/document/10097212/.

[16] Hamel, P., Davies, M. E. P., Yoshii, K. & Goto, M. Transfer learning in MIR: sharing learned latent representations for music audio classification and similarity. Tech. Rep., National Institute of Advanced Industrial Science and Technology (AIST), Japan (2013).

[17] Cífka, O. Deep learning methods for music style transfer. Tech. Rep. URL `https://tel.archives-ouvertes.fr/tel-03499991`.

[18] Ding, Y. & Lerch, A. Audio embeddings as teachers for music classification (2023). URL `http://arxiv.org/abs/2306.17424`.

[19] Kim, J., Urbano, J., Liem, C. C. & Hanjalic, A. One deep music representation to rule them all? A comparative analysis of different representation learning strategies. *Neural Computing and Applications* **32**, 1067–1093 (2020).

[20] Yuan, R. *et al.* MARBLE: Music audio representation benchmark for universal evaluation (2023). URL `http://arxiv.org/abs/2306.10548`.

[21] Komar, A. Schenker's Conception of Musical Structure. Tech. Rep. (1959).

[22] Samson, J. Schoenberg's 'Atonal' Music. *Tempo* 16–25 (1974). URL `http://www.jstor.org/stable/944097`.

[23] Lydian Chromatic Concep Of Tonal Organization (George Russell) .

[24] Levy, E. A Theory of Harmony URL `https://books.google.se/books/about/A_Theory_of_Harmony.html?id=cTAWsmYzwDwC&redir_esc=y`.

[25] Vaswani, A. *et al.* Attention Is All You Need (2017). URL `http://arxiv.org/abs/1706.03762`.

[26] OpenAI. GPT-4 Technical Report (2023). URL `http://arxiv.org/abs/2303.08774`.

[27] Koffka, K. *Principles of Gestalt psychology*, vol. 44 (2013).

[28] Lerdahl, F. & Jackendoff, R. S. *A Generative Theory of Tonal Music* (The MIT Press, Boston, 1985).

[29] Smith, J. B. L., Chew, E. & Mary, Q. A meta-analysis of the MIREX structure segmentation task. Tech. Rep. (2013).

[30] Nieto, O. *et al.* Audio-based music structure analysis: current trends, open challenges, and applications. *Transactions of the International Society for Music Information Retrieval* (2020).

[31] Nieto, O., Farbood, M. M., Jehan, T. & Bello, J. P. Perceptual analysis of the f-measure for evaluating section boundaries in music. Tech. Rep. URL http://www.music-ir.org/mirex/wiki/MIREX.

[32] Salamon, J., Nieto, O. & Bryan, N. J. Deep embeddings and section fusion improve music segmentation. Tech. Rep.

[33] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015). URL https://doi.org/10.1038/nature14539.

[34] Sarker, I. H. Deep Learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science* **2**, 420 (2021). URL https://doi.org/10.1007/s42979-021-00815-1.

[35] Rosenblatt, F. The perceptron - a perceiving and recognizing automaton. Tech. Rep. 85-460-1, Cornell Aeronautical Laboratory, Ithaca, New York (1957).

[36] Huitt, W. & Hummel, J. Piagets Theory of Cognitive Development. *Educational Psychology Interactive* (2003).

[37] Liu, S. *et al.* Audio self-supervised learning: A survey (2022).

[38] Spijkervet, J. & Burgoyne, J. A. Contrastive learning of musical representations (2021). URL http://arxiv.org/abs/2103.09410.

[39] Balestriero, R. *et al.* A cookbook of self-supervised learning (2023). URL http://arxiv.org/abs/2304.12210.

[40] Sikaroudi, M. *et al.* Offline versus online triplet mining based on extreme distances of histopathology patches (2020). URL http://arxiv.org/abs/2007.02200http://dx.doi.org/10.1007/978-3-030-64556-4_26.

[41] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. & Shah, R. Signature verification using a "Siamese" time delay neural network. In Cowan, J., Tesauro, G. & Alspector, J. (eds.) *Advances in Neural Information Processing Systems*, vol. 6 (Morgan-Kaufmann, 1993). URL `https://proceedings.neurips.cc/paper_files/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf`.

[42] Lee, J., Park, J., Kim, K. L. & Nam, J. SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification. *Applied Sciences (Switzerland)* **8** (2018).

[43] Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 32 (Curran Associates, Inc., 2019). URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf`.

[44] PyTorch Lightning 2.0.1 documentation. URL `https://lightning.ai/docs/pytorch/stable/`.

[45] Loshchilov, I. & Hutter, F. Decoupled weight decay regularization (2017). URL `http://arxiv.org/abs/1711.05101`.

[46] Schindler, A., Lidy, T. & Böck, S. Deep Learning for MIR Tutorial (2020). URL `http://arxiv.org/abs/2001.05266`.

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

[48] Dai, W., Dai, C., Qu, S., Li, J. & Das, S. Very deep convolutional neural networks for raw waveforms (2016). URL `http://arxiv.org/abs/1610.00087`.

[49] Vahidi, C. *et al.* Mesostructures: beyond spectrogram loss in differentiable time-frequency analysis (2023). URL `http://arxiv.org/abs/2301.10183`.

[50] Spijkervet, J. Spijkervet/torchaudio-augmentations: v1.0 (2021). URL `https://doi.org/10.5281/zenodo.4748582#.ZATdOBUmob8.mendeley`.

[51] Kharitonov, E. *et al.* Data augmenting contrastive learning of speech representations in the time domain (2020). URL http://arxiv.org/abs/2007.00991.

[52] Yang, Y.-Y. *et al.* TorchAudio: building blocks for audio and speech processing (2021). URL http://arxiv.org/abs/2110.15018.

[53] SoX - Sound eXchange | HomePage. URL https://sox.sourceforge.net/.

[54] Fastl, H. & Zwicker, E. Just-Noticeable Sound Changes. In Fastl, H. & Zwicker, E. (eds.) *Psychoacoustics: Facts and Models*, 175–202 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007). URL https://doi.org/10.1007/978-3-540-68888-4_7.

[55] Schroff, F., Kalenichenko, D. & Philbin, J. FaceNet: A unified embedding for face recognition and clustering (2015). URL http://arxiv.org/abs/1503.03832http://dx.doi.org/10.1109/CVPR.2015.7298682.

[56] Khosla, P. *et al.* Supervised contrastive learning (2020). URL http://arxiv.org/abs/2004.11362.

[57] Steinmetz, C. J. & Reiss, J. D. auraloss: Audio-focused loss functions in PyTorch. In *Digital Music Research Network One-day Workshop (DMRN+15)*, vol. 2021-June (Institute of Electrical and Electronics Engineers Inc., 2020).

[58] Das, D. *et al.* Mixed precision training of convolutional neural networks using integer operations (2018). URL http://arxiv.org/abs/1802.00930.

[59] Li, S. *et al.* PyTorch Distributed: experiences on accelerating data parallel training (2020). URL http://arxiv.org/abs/2006.15704.

[60] Serrà, J., Müller, M., Grosche, P. & Ll Arcos, J. Unsupervised music structure annotation by time series structure features and segment similarity. Tech. Rep. (2013). URL http://www.music-ir.org/mirex/wiki/MIREX.

[61] Tzanetakis, G., Essl, G. & Cook, P. *Automatic musical genre classification of audio signals* (The International Society for Music Information Retrieval). URL http://ismir2001.ismir.net/pdf/tzanetakis.pdf.

[62] Bertin-Mahieux, T., Ellis, D. P. W., Whitman, B. & Lamere, P. The Million Song Dataset.

[63] Smith, J. B. L., Ashley Burgoyne, J., Fujinaga, I., De Roure, D. & Downie, J. S. Design and creation of a large-scale database of structural annotations. Tech. Rep. (2011).

[64] Turnbull, D., Lanckriet, G., Pampalk, E. & Goto, M. A Supervised approach for Detecting boundaries in music using difference features and boosting. Tech. Rep. (2007).

[65] Nieto, O. & Bello, J. P. MSAF: Music Structure Analysis Framework. Tech. Rep. URL `https://github.com/urinieto/msaf`.

[66] Raffel, C. *et al.* mir_eval: A transparent implementation of common MIR metrics. Tech. Rep.

[67] Serrà, J., Müller, M., Grosche, P. & Arcos, J. L. Unsupervised detection of music boundaries by time series structure features. *Proceedings of the AAAI Conference on Artificial Intelligence* **26**, 1613–1619 (2021). URL `https://ojs.aaai.org/index.php/AAAI/article/view/8328`.

[68] Foote, J. Automatic audio segmentation using a measure of audio novelty. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, vol. 1, 452–455 (2000).

[69] Nieto, O. & Jehan, T. Convex non-negative matrix factorization for automatic music structure identification. Tech. Rep. URL `http://developer.echonest.com`.

[70] Kaiser, F., Sikora, T. & Peeters, G. MIREX 2012-music structural segmentation tasks: IRCAMstructure submission. Tech. Rep.

[71] Grill, T. & Schlüter, J. Structural segmentation with convolutional neural networks MIREX submission. Tech. Rep. URL `http://docs.scipy.org/doc/scipy/reference/`.

[72] Sargent, G., Bimbot, F. & Vincent, E. Supplementary material to the article: Estimating the structural segmentation of popular music pieces under regularity constraints. Tech. Rep. URL https://inria.hal.science/hal-01368683.

[73] Nieto, O. & Bello, J. P. MIREX 2014 ENTRY: 2D fourier magnitude coefficients. Tech. Rep. URL https://github.com/urinieto/SegmenterMIREX2014.

[74] Cannam, C. *et al.* MIREX 2015: VAMP PLUGINS FROM THE CENTRE FOR DIGITAL MUSIC. Tech. Rep. URL http://code.soundsoftware.ac.uk/projects/beatroot-vamp/.

[75] Nieto, O. MIREX: MSAF V0.1.0 SUBMISSION. Tech. Rep. URL https://github.com/urinieto/msaf/releases/.

[76] Schlüter, J. & Böck, S. Improved musical onset detection with Convolutional Neural Networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 6979–6983 (Institute of Electrical and Electronics Engineers Inc., 2014).

[77] Grill, T. & Schluter, J. Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, 1296–1300 (2015).

[78] Cohen-Hadria, A. & Peeters, G. Music structure boundaries estimation using multiple self-similarity matrices as input depth of convolutional neural networks. Tech. Rep. (2017). URL http://www.aes.org/e-lib.

[79] Xuan, H., Stylianou, A. & Pless, R. Improved embeddings with easy positive triplet mining. Tech. Rep. URL https://github.com/littleredxh/EasyPositiveHardNegative.

[80] TikZ.net – Graphics with TikZ in LaTeX. URL https://tikz.net/.