

UNSUPERVISED LEARNING OF DEEP FEATURES FOR MUSIC SEGMENTATION

Matthew C. McCallum

Gracenote Inc.

ABSTRACT

Music segmentation refers to the dual problem of identifying boundaries between, and labeling, distinct music segments, e.g., the chorus, verse, bridge etc. in popular music. The performance of a range of music segmentation algorithms has been shown to be dependent on the audio features chosen to represent the audio. Some approaches have proposed learning feature transformations from music segment annotation data, although, such data is time consuming or expensive to create and as such these approaches are likely limited by the size of their datasets. While annotated music segmentation data is a scarce resource, the amount of available music audio is much greater. In the neighboring field of semantic audio unsupervised deep learning has shown promise in improving the performance of solutions to the query-by-example and sound classification tasks. In this work, unsupervised training of deep feature embeddings using convolutional neural networks (CNNs) is explored for music segmentation. The proposed techniques exploit only the time proximity of audio features that is implicit in any audio timeline. Employing these embeddings in a classic music segmentation algorithm is shown not only to significantly improve the performance of this algorithm, but obtain state of the art performance in unsupervised music segmentation.

Index Terms— Music information retrieval, Acoustic signal processing, Convolutional neural network, Deep learning

1. INTRODUCTION

Music segmentation refers to the task of labeling distinct segments of music in a way that is similar to a human annotation. For example the chorus, verse, intro, outro and bridge in popular music.. The boundary between such segments may be due to a number of factors, for example, a change in melody or chord progression, a change in rhythm, changes in instrumentation, dynamics, key or tempo. This task is generally evaluated with two classes of metrics. The first class, boundary detection, refers to the ability of the algorithm to locate the locations of such boundaries in time. The second class, segment labelling, refers to the labelling of segments where two segments that are disconnected in time are labelled as same or different based on their perceptual similarity [1–3].

1.1. Prior Work

A number of techniques exist in the literature that address either the boundary detection [4–6], segment labelling [7] problems, or both simultaneously [8–11]. Methods addressing solely the former metric typically involve the generation of a novelty function via a self similarity matrix (SSM) representation. Other approaches addressing both of the aforementioned metrics focus on clustering audio features based on characteristics that are expected to remain homogeneous within a given musical segment. Such clustering has been

performed on the basis of timbral and harmonic features in the context of spectral clustering [9], or in the context of time-translation invariant features such as HMM state histograms [8]. In addition, a number of feature transformations have been considered with respect to clustering approaches, including non-negative matrix factorizations (NMFs) of audio features [11] or self-similarity matrices [12], as well as learned features from labelled segmentation data [5].

In the context of supervised deep learning some work has addressed the problem of music segmentation [13–15], where performance is likely bound by the limited annotated music segmentation data available. The largest effort to collect labeled music segmentation data is in the SALAMI dataset [16], providing 2246 annotations of music structure in 1360 audio files. This amount of data is small in the context of deep learning, but promising results were obtained.

Recently, in the neighboring field of semantic audio, unsupervised approaches towards embedding audio features have attracted attention in the literature [17–19]. In particular, the benefits of learning audio feature embeddings in an unsupervised manner have become apparent in [17]. Because there is no requirement for labeled data in training such embeddings, they can be trained on much larger datasets and in turn, this has achieved impressive results when such pre-trained embeddings are employed in sound classification and query by example tasks [17]. Despite the promising results obtained for tasks related to the identification of events in audio, little to no investigation has been made into the use of such unsupervised audio feature embeddings for music content specifically.

1.2. Contributions

This work focuses on the unsupervised training of CNNs to obtain meaningful features for music segmentation methods. This is a natural progression of the application of modern machine learning methods to the problem of music segmentation for three reasons.

Firstly, previous literature pays careful attention to obtaining features that are representative of musical characteristics that are perceptually important in identifying segment boundaries such as timbre or harmonic repetition [4–6, 9, 10, 20–24]. Whether or not a given feature is important in identifying a segment boundary or label may be genre or song specific and so a data based approach may be promising in either producing a representation that better generalizes across genres or in the least may be arbitrarily learned for each genre specifically. Several data based approaches have been investigated for the music segmentation problem [5, 10, 13], with little to no work in the context of unsupervised deep learning.

Secondly, musical content has a structure that may be exploited to further improve the machine learning methodologies employed in works such as [17–19]. The characteristics that define music such as rhythm and harmonicity have not been exploited in these previous works, warranting some investigation in the context of music.

Finally, labeled data for the problem of music segmentation is notoriously time consuming and/or expensive to produce [25], as

such an unsupervised machine learning approach that can exploit large amounts of unlabeled music data is highly desirable. The approach in this paper investigates deep learning in an unsupervised framework, where only the time locality and a comparative analysis of time local data is exploited. Such an approach overcomes the necessity to hand annotate data, and hence may be scaled to the full extent of the available music data. No longer limited by the size of the training dataset, it is expected to generalize better than hand crafted features that rely on aspects such as timbre or repetition that may be specific to certain music genres.

2. AUDIO FEATURE EMBEDDING

The approach of here is to create an embedding for audio features by transforming an audio representation via a CNN into a domain that is representative of musical structure. These embeddings may be trained by employing a loss function, such as contrastive loss [26], or triplet loss [27], that observes positive (similar) or negative (dissimilar) pairs or triplets of examples, where a triplet represents an anchor and both a positive example and a negative example. The triplet loss function is often shown to result in superior performance to contrastive loss, which has been argued to be due to its relative nature [28]. That is, the triplet loss function is designed to provide a positive gradient with respect to increasing distance between positive pairs or the anchor and positive example of a triplet, relative to the distance between negative pairs, up to a given margin. As such, it is employed in this research.

If a transformation from an input feature, $x[q, k]$ (or equivalently x for notational simplicity) to an embedding space, e.g., via a CNN, is described as $f: \mathbb{R}^{K \times Q} \rightarrow \mathbb{R}^D$, then a Euclidean triplet loss with a given margin, α , and a mini-batch of training data consisting of the set $\mathcal{T} = \{x_a^c, x_p^c, x_n^c\}_{c=0}^{C-1}$ may be described as,

$$\mathcal{L}(\mathcal{T}) = \sum_{c=0}^{C-1} \left[\|f(x_a^c) - f(x_p^c)\|_2^2 - \|f(x_a^c) - f(x_n^c)\|_2^2 + \alpha \right]_+, \quad (1)$$

where subscript a, p and n represent an anchor, positive and negative input example respectively.

Ideally, in the context of music segmentation positive examples will be composed of audio feature representations from the same music segment as the anchor, while negative examples will represent audio features from distinct music segments. Thus, once trained, input features could be transformed via that CNN into a space where distinct clusters with respect to a Euclidean distance metric would represent distinct music segments. In an unsupervised context with no prior information about the music segmentation, exact selection of such examples is not possible, however strategies are described in Section 3 that may improve this selection.

The properties that are significant in forming clusters representing music segments in the embedded space are learned from the data at the CNN input. These embedded features may be representative of any of the musical qualities mentioned in Section 1, provided that these qualities are observable from the input features. Here, a constant-Q transform (CQT) [29] is used due to its ability to accurately represent transient and harmonic audio qualities, its translation invariant representation of harmonic structures, and the promising results observed for this feature specifically in [30].

A time window of CQT data providing K frequency bins across Q time windows forms a time frequency representation that may compose any of x_a^c, x_p^c or x_n^c . In this work it is found significantly advantageous to synchronize the CQT analysis windows with the beat of the music. That is, if a beat at index i occurs at time b_i , then

R CQT windows are analyzed centered at times $b_i + r(b_{i+1} - b_i)/R$ for integers, $r \in \{0..R-1\}$. These beat synchronized CQT representations are then aggregated across B beats providing $Q = BR$ time indices in the representation $x[q, k]$ at the input to the CNN.

3. SAMPLING

Motivated by the unreasonable effectiveness of data in deep learning [31], methods are proposed here that create noisy positive / negative examples exploiting the facts that a) musical segments form contiguous regions in a song's timeline, and b) each distinct musical segment label typically occurs for the minority of a song's timeline. Specifically, it is proposed to use the time proximity information implicit in a song's features - sampling features that occur close together or at a minimum distance apart for positive and negative examples respectively. That is, an anchor beat index, i_a , is selected via a uniform sampling of the beat indices in a given song, $\{0..L-1\}$. Thereafter, a positive beat index, i_p , is chosen uniformly sampled from beat indices $\{\max(i_a - \delta_p, 0)..\min(i_a + \delta_p, L-1)\}$, and a negative beat index i_n is chosen as uniformly sampled across two regions, $\{\max(i_a - \delta_{n,max}, 0)..\max(i_a - \delta_{n,min}, 0)\}$, and $\{\min(i_a + \delta_{n,min}, L-1)..\min(i_a + \delta_{n,max}, L-1)\}$. An example of the positive and negative sampling distributions is shown in Fig. 1. Intuitively, this results in an embedding in which clusters represent features that frequently occur close together. This is useful for the problem of the structural segmentation of music as segment boundaries are typically infrequent enough to be described as rare events, at least in comparison to lengths of the segments themselves.

It is interesting to consider the rates of false positives and negatives that result from the aforementioned sampling paradigm. Upon selection of i_a it may be denoted to fall somewhere in the n th musical segment of class s and length $l_{s,n}$. If $\delta_p < l_{sep}$, where l_{sep} is the minimum number of indices between s and any identically labeled segment, then the probability of the positive example being selected from a distinct segment, i.e., a false positive, is,

$$P(\text{FP}|l_{s,n}; \delta_p) = \begin{cases} \frac{2\delta_p - l_{s,n}}{l_{s,n}} & l_{s,n} \leq \delta_p \\ \frac{\delta_p^2}{2l_{s,n}^2} - \frac{3\delta_p}{4l_{s,n}} + \frac{1}{2} & \delta_p < l_{s,n} < 2\delta_p \\ \frac{\delta_p}{4l_{s,n}} & 2\delta_p \leq l_{s,n} \end{cases}$$

Similarly, if $\delta_{n,min} < l_{sep}$ and $\delta_{n,max} \geq L$, then the probability of a false negative is simply $\frac{\sum_l l_{s,m}}{L} - (1 - P(\text{FP}|l_{s,n}; \delta_{n,min}))$.

In practice the aforementioned assumptions are realistic in many scenarios, but do not hold under all conditions. An empirical analysis of the rate of false positives is shown in Fig. 2. It is clear that with an increasing δ_p , an increasing rate of false positives is observed, although it is important to note that smaller δ_p restricts the maximum observed time separation between the anchor and positive example. With musical phrases often lasting 16 beats, it is reasonable to set $\delta_p \geq 16$ to discourage distinct clustering of features within a single phrase. An empirical analysis false negatives is shown in Fig. 3. There it can be seen that for the datasets shown $\delta_{n,min} > 28$ and $\delta_{n,max} > 116$ result in relatively low false negative rates.

While structural annotations are not available under the scope of unsupervised learning, it is interesting to ask whether analysis of signal features may be employed to decrease the false-negative or false-positive rate. It was shown in [7] that 2D Fourier magnitude coefficients of HPCPs can be a useful feature in segment labeling. It is proposed here to use a similar feature to inform the sampling of positive and negative examples. That is, for every selected i_a ,

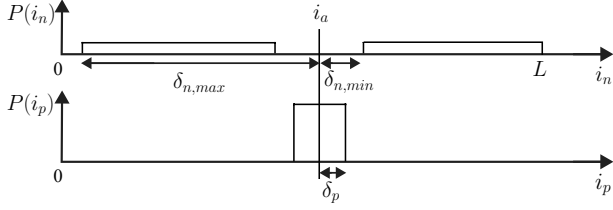


Fig. 1. Sampling distributions for i_n and i_p , for a given choice of i_a .

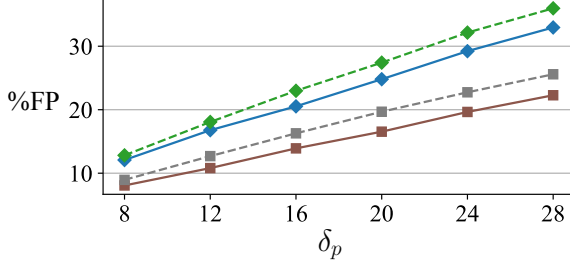


Fig. 2. Observed false positive (FP) rates in the BeatlesTUT (\diamond), and SALAMI (\square), datasets for varying δ_p . Dashed lines show results with unbiased sampling, and solid lines employ the comparative 2D FFT technique described here.

a comparison of the log-amplitude of the 2D Fourier transform of the log-amplitude of 8 beat long CQT segments is considered. The Euclidean distances between these CQT segments centered at two times before i_a , i.e., $i_a - 4$ and $i_a - 16$, and two times after i_a , i.e., $i_a + 4$ and $i_a + 16$ are considered. The side (“before” where $i < i_a$ or “after” where $i > i_a$) with minimum Euclidean distance is then assumed to be more likely in the same musical segment as i_a and is chosen from which to sample i_p while the opposite side may be chosen from which to sample i_n , in addition to the constraints, δ_p , $\delta_{n,min}$ and $\delta_{n,max}$ above. The changes in false negative and false positive rates employing this sampling paradigm can be seen in Figures 2 and 3, respectively, where some improvement is observed.

4. BOUNDARY DETECTION

To evaluate the effectiveness of the music embeddings described in this work, the problem of music segment boundary detection is considered. Perhaps the simplest and most well known method for music boundary detection is that of Foote [4]. While this has been surpassed in performance by several methods, e.g., [6, 9], its simplicity makes it effective in demonstrating the utility of the audio embeddings proposed here, as will be seen in the results in Section 5.

The SSM of the proposed features is computed as,

$$S[i, j] = \|f(x_i[q, k]) - f(x_j[q, k])\|_2^2 \quad (2)$$

where $x_i[q, k]$ and $x_j[q, k]$ correspond to beat synchronous CQT segments centered at beats i and j , respectively.

Note that this work endeavors to create features that are close with respect to Euclidean distance at any point within a music segment. If successful, the SSM of embedded features typically contains block structures as opposed to the path structures typically representative of repetition. These structures are evident in Fig. 4. It was found beneficial in practice to perform median filtering on $S[i, j]$ to produce $\bar{S}[i, j]$ which reduces noise in the distances between embedded features while maintaining the aforementioned structures.

To detect segment boundaries in $\bar{S}[i, j]$, a checkerboard kernel,

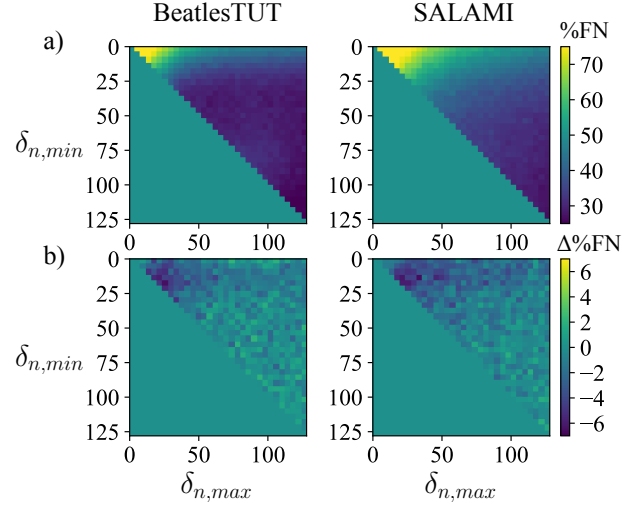


Fig. 3. Observed false negative (FN) rates for various $\delta_{n,min}$ and $\delta_{n,max}$. Row (a) shows FN rates resulting from unbiased sampling, while row (b) represents the change in FN rates when employing the comparative 2D FFT technique described in this paper.

$$g[i, j] = \text{sgn}(i)\text{sgn}(j)e^{-\frac{(i-j)^2}{\sigma}} \quad (3)$$

for $-\kappa \leq i, j \leq \kappa$ is convolved along the diagonal of the SSM,

$$\eta[\nu] = \sum_{i=-\kappa}^{\kappa} \sum_{j=-\kappa}^{\kappa} \bar{S}[\nu + i, \nu + j] g[\nu + i, \nu + j] \quad (4)$$

producing the novelty function $\eta[\nu]$. Note that it was found advantageous to set $g[i, j]$ to 0 where $|i| \leq 1$ or $|j| \leq 1$. Finally, boundaries are detected as peaks in the novelty function. Specifically, peaks at which the peak-to-mean ratio exceeds a given threshold, τ , are selected as segment boundaries, i.e., where,

$$\frac{(2T+1)\eta[\nu]}{\sum_{t=-T}^T \eta[\nu+t]} > \tau. \quad (5)$$

5. RESULTS

For evaluation, two datasets are considered: the BeatlesTUT dataset consisting of 174 hand annotated tracks from The Beatles catalogue [20], and the internet archive portion of the SALAMI dataset (SALAMI-IA) consisting of 375 hand annotated recordings [16]. The former dataset is perhaps the most widely evaluated in the music segmentation literature. The latter is employed here for two reasons, firstly the complete SALAMI dataset audio is not available to the author due to copyright restrictions, and secondly, the SALAMI-IA dataset is particularly interesting as it consists primarily of live recordings with many imperfections. It provides a dataset that is indicative of segmentation performance when there are mistakes either by musicians or recording engineers, resulting in imperfect repetitions and distorted or noisy audio in many cases.

For comparison, two baseline algorithms are included in the results as specified and measured in [30], note that these algorithms too included beat-synchronized features. Firstly, the method of [4] is included as it most closely mirrors the algorithm of Section 4. Secondly, the algorithm of [6] is widely evaluated as having the best

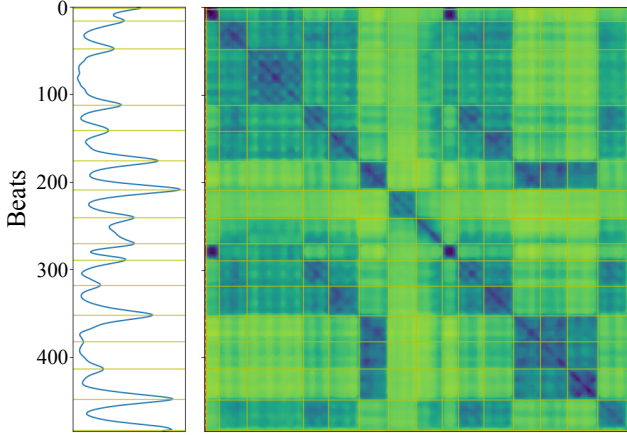


Fig. 4. Example of median filtered SSM and novelty function for the song "Starlight" by Muse. On both, detected boundaries are marked with yellow lines.

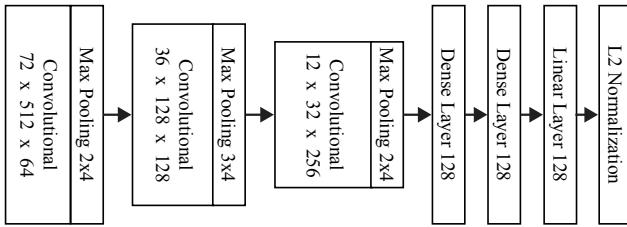


Fig. 5. Architecture of the CNN employed in experiments. For each convolutional layer, dimensions represent time, frequency and channel respectively. Each convolutional and dense layer employ a ReLU activation. All convolutional layers employ 6×4 kernels.

performance with respect to unsupervised boundary detection in music segmentation. CQT features have been evaluated to provide superior performance in [30], and are used at the input to all algorithms, proposed and benchmark, in the results presented here. For the proposed algorithms, three methods are investigated, "Unsynchronized", employing a constant hop size of 3.6 ms between successive CQT windows; "Beat-Synchronized", employing 128 CQT windows centered at times linearly interpolated between successive beat markers (estimated during training and inference by an algorithm similar to [32]); and "Biased", employing the same features as the "Beat-Synchronized" approach, but using the 2D Fourier Transform comparative sampling described in Section 3. The CQT features use a minimum frequency of 40 Hz, 12 bins per octave and 6 octaves.

The structure of CNNs employed in this paper is chosen to be simple enough so that it may be trained on most modern high performance GPUs. The architecture described in Fig. 5 was found to be effective and is employed in the experiments here. During training, mini-batches of size $C = 96$ consisting of 16 triplets from each of 6 randomly selected tracks were formed in real-time by randomly selecting from 28,345 songs, excluding all songs in the SALAMI-IA and BeatlesTUT dataset. 256 min-batches form 1 epoch, and training took place over 240 epochs taking approximately 8 hours. The triplet margin was set to $\alpha = 0.1$. Despite the observed error rates in Figures 3 and 3, [27, 28] argue that it is the difficulty of separation between examples that is important, and so not all false positives and negatives are equal. In practice it was found that $\delta_p = 16$, $\delta_{n,min} = 1$ and $\delta_{n,max} = 96$ provide optimal results.

For boundary detection, embedded features were observed at ev-

Table 1. Performance metrics for the BeatlesTUT dataset

Algorithm	F-Measure	Precision	Recall
[4]	0.503 ± 0.18	0.579 ± 0.21	0.461 ± 0.17
[6]	0.651 ± 0.17	0.622 ± 0.19	0.708 ± 0.19
Unsynchronized	0.597 ± 0.17	0.589 ± 0.19	0.625 ± 0.17
Beat-Synchronized	0.648 ± 0.17	0.647 ± 0.20	0.677 ± 0.18
Biased Sampling	0.662 ± 0.17	0.663 ± 0.20	0.691 ± 0.19

Table 2. Performance metrics for the SALAMI-A dataset

Algorithm	F-Measure	Precision	Recall
[4]	0.446 ± 0.17	0.457 ± 0.21	0.483 ± 0.19
[6]	0.493 ± 0.17	0.454 ± 0.20	0.595 ± 0.19
Unsynchronized	0.497 ± 0.16	0.429 ± 0.18	0.653 ± 0.15
Beat-Synchronized	0.535 ± 0.15	0.491 ± 0.20	0.660 ± 0.16
Biased Sampling	0.533 ± 0.16	0.491 ± 0.21	0.656 ± 0.16

ery beat for beat-synchronized approaches, or once every 0.2484 seconds for "Unsynchronized" (this is approximately twice per beat at a typical 120 BPM to ensure this method is not disadvantaged by any shortcoming in time resolution). SSM representations were median filtered using an 8×8 window. The checkerboard kernel was configured with $\sigma = 18.5$ and $\kappa = 40$ and for peak picking, the crest factor window size $T = 10$ and threshold $\tau = 1.35$ was employed.

For evaluation, the trimmed F-measure, precision and recall of the boundary detection hit rate at the 3 second tolerance level are employed [1]. The evaluation of the proposed and reference algorithms for the BeatlesTUT and SALAMI-IA dataset are shown in Table 1 and Table 2, respectively. It should be noted here that for the proposed algorithms, any selection of parameters was performed by observing results on the BeatlesTUT dataset only, and so the SALAMI-IA dataset displays the boundary detection algorithm's ability to generalize to unseen data.

It is interesting to see that simply by employing the proposed deep features in an algorithm similar to that of Foote [4], such a method becomes competitive with the state of the art in unsupervised music segmentation. Furthermore, on the SALAMI-IA dataset, a significant performance improvement over the state of the art is observed without any additional parameter adjustment. This result might be postulated to be due to the poor quality of audio / music data in this portion of the SALAMI dataset. Because the algorithm of [6] is designed to detect changes in repetition patterns, when these patterns become imperfect, or corrupted by noise, a performance drop might be expected. In the proposed embedding, clustering of features is performed simply based on the time proximity of features observed from the training data, which contains many of the aforementioned imperfections providing some robustness.

6. CONCLUSION

In this work, methods for the unsupervised training of music embeddings were investigated with respect to their utility in the task of music segmentation. In particular, it was shown that by employing such embeddings in a traditional music segmentation algorithm, the performance of this algorithm can obtain state of the art performance. It was found that a common musical feature, rhythm, may be exploited in beat-synchronized sampling (and in the 2D Fourier Transform comparative sampling of Section 3) to further improve performance.

7. REFERENCES

- [1] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel, “mir_eval: A transparent implementation of common MIR metrics,” in *ISMIR*, 2014, pp. 367–372.
- [2] Oriol Nieto, Morwaread M Farbood, Tristan Jehan, and Juan Pablo Bello, “Perceptual analysis of the F-measure for evaluating section boundaries in music,” in *ISMIR*, 2014, pp. 265–270.
- [3] Hanna M Lukashevich, “Towards quantitative measures of evaluating song segmentation,” in *ISMIR*, 2008, pp. 375–380.
- [4] Jonathan Foote, “Automatic audio segmentation using a measure of audio novelty,” in *Multimedia and Expo (ICME). IEEE International Conference on*, 2000, pp. 452–455.
- [5] Brian McFee and Daniel PW Ellis, “Learning to segment songs with ordinal linear discriminant analysis,” in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2014, pp. 5197–5201.
- [6] Joan Serra, Meinard Müller, Peter Grosche, and Josep Ll Arcos, “Unsupervised music structure annotation by time series structure features and segment similarity,” *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1229–1240, 2014.
- [7] Oriol Nieto and Juan Pablo Bello, “Music segment similarity using 2D-Fourier magnitude coefficients,” in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2014, pp. 664–668.
- [8] Mark Levy and Mark Sandler, “Structural segmentation of musical audio by constrained clustering,” *IEEE transactions on audio, speech, and language processing*, vol. 16, no. 2, pp. 318–326, 2008.
- [9] Brian McFee and Dan Ellis, “Analyzing song structure with spectral clustering,” in *ISMIR*, 2014, pp. 405–410.
- [10] Ron J Weiss and Juan Pablo Bello, “Unsupervised discovery of temporal structure in music,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1240–1251, 2011.
- [11] Oriol Nieto and Tristan Jehan, “Convex non-negative matrix factorization for automatic music structure identification,” in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2013, pp. 236–240.
- [12] Florian Kaiser and Thomas Sikora, “Music structure discovery in popular music using non-negative matrix factorization,” in *ISMIR*, 2010, pp. 429–434.
- [13] Karen Ullrich, Jan Schlüter, and Thomas Grill, “Boundary detection in music structure analysis using convolutional neural networks,” in *ISMIR*, 2014, pp. 417–422.
- [14] Jan Schlüter and Sebastian Böck, “Improved musical onset detection with convolutional neural networks,” in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2014, pp. 6979–6983.
- [15] Thomas Grill and Jan Schlüter, “Music boundary detection using neural networks on spectrograms and self-similarity lag matrices,” in *EUSIPCO*, 2015, pp. 1296–1300.
- [16] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie, “Design and creation of a large-scale database of structural annotations,” in *ISMIR*, 2011, pp. 555–560.
- [17] Aren Jansen, Manoj Plakal, Ratheet Pandya, Daniel PW Ellis, Shawn Hershey, Jiayang Liu, R Channing Moore, and Rif A Saurous, “Unsupervised learning of semantic audio representations,” in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2018, pp. 126–130.
- [18] Justin Salamon and Juan Pablo Bello, “Unsupervised feature learning for urban sound classification,” in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2015, pp. 171–175.
- [19] Justin Salamon and Juan Pablo Bello, “Feature learning with deep scattering for urban sound analysis,” in *EUSIPCO*, 2015, pp. 724–728.
- [20] Jouni Paulus and Anssi Klapuri, “Music structure analysis by finding repeated parts,” in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, 2006, pp. 59–68.
- [21] Wei Chai, “Semantic segmentation and summarization of music: methods based on tonality and recurrent structure,” *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 124–132, 2006.
- [22] Matija Marolt, “A mid-level melody-based representation for calculating audio similarity,” in *ISMIR*, 2006, pp. 280–285.
- [23] Kristoffer Jensen, “Multiple scale music segmentation using rhythm, timbre, and harmony,” *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 159–159, 2007.
- [24] Johan Pauwels, Florian Kaiser, and Geoffroy Peeters, “Combining harmony-based and novelty-based approaches for structural segmentation,” in *ISMIR*, 2013, pp. 601–606.
- [25] Cheng-i Wang, Gautham J Mysore, and Shlomo Dubnov, “Revisiting the music segmentation problem with crowdsourcing,” in *ISMIR*, 2017, pp. 738–744.
- [26] Raia Hadsell, Sumit Chopra, and Yann LeCun, “Dimensionality reduction by learning an invariant mapping,” in *In Proc. Computer Vision and Pattern Recognition Conference (CVPR)*, 2006, pp. 1735–1742.
- [27] Florian Schroff, Dmitry Kalenichenko, and James Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.
- [28] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl, “Sampling matters in deep embedding learning,” in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [29] Christian Schölkhuber and Anssi Klapuri, “Constant-Q transform toolbox for music processing,” in *7th Sound and Music Computing Conference, Barcelona, Spain*, 2010, pp. 3–64.
- [30] Oriol Nieto and Juan Pablo Bello, “Systematic exploration of computational music structure research,” in *ISMIR*, 2016, pp. 547–553.
- [31] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 843–852.
- [32] Anssi P Klapuri, Antti J Eronen, and Jaakko T Astola, “Analysis of the meter of acoustic musical signals,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 342–355, 2006.