MUSIC BOUNDARY DETECTION USING NEURAL NETWORKS ON SPECTROGRAMS AND SELF-SIMILARITY LAG MATRICES

Thomas Grill and Jan Schlüter

Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria

ABSTRACT

The first step of understanding the structure of a music piece is to segment it into formative parts. A recently successful method for finding segment boundaries employs a Convolutional Neural Network (CNN) trained on spectrogram excerpts. While setting a new state of the art, it often misses boundaries defined by non-local musical cues, such as segment repetitions. To account for this, we propose a refined variant of self-similarity lag matrices representing long-term relationships. We then demonstrate different ways of fusing this feature with spectrogram excerpts within a CNN, resulting in a boundary recognition performance superior to the previous state of the art. We assume that the integration of more features in a similar fashion would improve the performance even further.

Index Terms— Music information retrieval, Acoustic signal processing, Feedforward neural networks

1. INTRODUCTION

Determining the temporal structure of a piece of music, that is, decomposing it into parts known as movements, phrases, chorus and verse, etc., is a major challenge in music analysis. See [1] for an overview of the field and existing techniques. The identification of transition points, or, *boundaries* between such structural elements is often a highly ambiguous task, even for briefed human annotators. The currently by far best-performing method for boundary detection developed by Ullrich et al. [2] uses a Convolutional Neural Network (CNN), trained on a large corpus of human-annotated structural annotations. The algorithm is based on mel-scaled log-magnitude spectrograms (MLSs), taking into account a temporal context of 16 or 32 seconds, depending on the desired precision. When making the classification decision whether a boundary is present or not, the CNN sees only this relatively short local context of sequential data. It is therefore unable to account for structural information such as repeated sections which manifest themselves on a larger temporal scale. Figure 1 represents an excerpt of the piece "The Wet Spot" by "Southern Culture On The Skids" (index 1358 in the SALAMI collection, see Section 4.1). The human-annotated boundaries (*ground truth*) are depicted by vertical marks at the top. Evidently, the CNN based solely on a MLS (Figure 1a) has difficulties of identifying certain boundaries, as indicated by low probabilities in the prediction curve (Figure 1b). The mel spectrogram alone does not seem to provide all necessary information. In order to remedy this lack of information, our approach is to additionally feed an alternative representation of the underlying audio data to the CNN, in the form of recurrence information, specifically *self-similarity lag matrices* (SSLMs, see Figures 1c and 1d). Such a matrix represents similarities of certain low-level features of one point in time in relation to points in the past, up to a certain *lag time*.

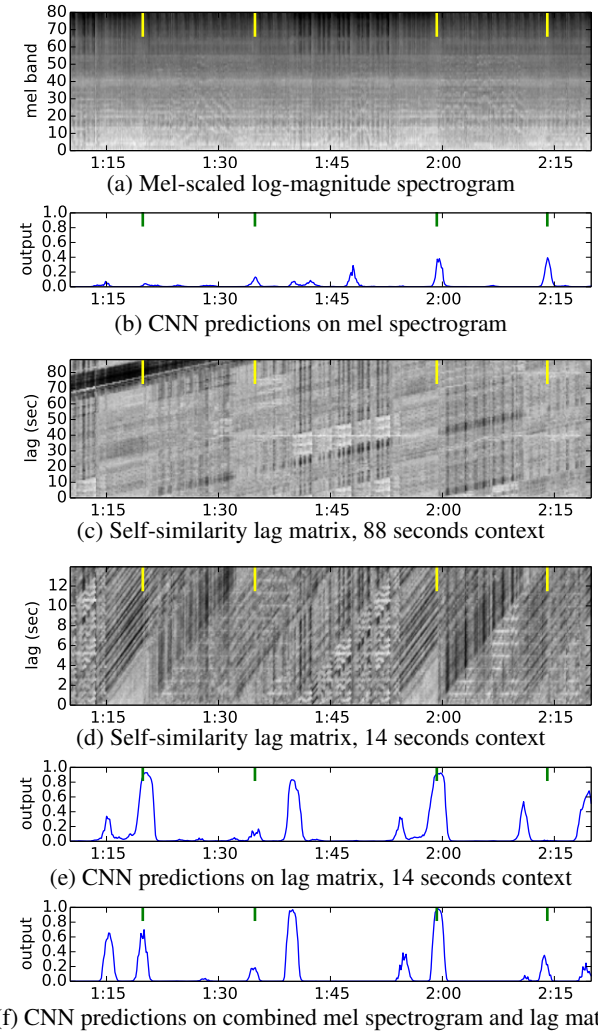
The structure of the paper is as follows: After giving an overview over related work in Section 2, we describe our proposed method in Section 3. In Section 4, we describe the experimental setup, our evaluation strategy, and our main results. We wrap up in Section 5 with a discussion and outlook.

2. RELATED WORK

Following [1], three fundamental approaches to music structure analysis can be distinguished: Novelty-based, detecting transitions between contrasting parts, homogeneity-based, identifying sections that are consistent with respect to their musical properties, and repetition-based, building on the determination of recurring patterns. Novelty is typically computed from self-similarity matrices (SSMs) or self-distance matrices (SDMs) by sliding a checkerboard kernel along the diagonal [3], building on audio descriptors like MFCCs, pitch class profiles, or rhythmic features [4]. Turnbull et al. [5] compute difference features on more complex audio feature sets and use trained Boosted Decision Stumps for boundary detection. In order to capitalize on repeated patterns, SSMs or SDMs are used with various heuristic rules and optimization schemes for structure formation [6–8]. McFee and Ellis employ spectral clustering [9], or add a supervised learning scheme using ordinal linear discriminant analysis and constrained clustering [10]. When using end-to-end neural network techniques such as Ullrich et al.'s CNNs [2], the separation between the fundamental approaches becomes blurred as the CNN infers the relationships between audio features and ground truth from

This research is funded by the Federal Ministry for Transport, Innovation & Technology (BMVIT) and the Austrian Science Fund (FWF) through project TRP 307-N23 and the Vienna Science and Technology Fund (WWTF) through project MA14-018.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE



(a) Mel-scaled log-magnitude spectrogram

(b) CNN predictions on mel spectrogram

(c) Self-similarity lag matrix, 88 seconds context

(d) Self-similarity lag matrix, 14 seconds context

(e) CNN predictions on lag matrix, 14 seconds context

(f) CNN predictions on combined mel spectrogram and lag matrix

Fig. 1: Boundary recognition using CNNs on different underlying audio features, illustrated on the piece "The Wet Spot" by "Southern Culture On The Skids". Please see `http://www.ofai.at/research/impml/projects/audiostreams/eusipco2015` for a version with audio.

the provided training data. In a similarly integral fashion, Serrà et al. [11] propose an unsupervised method explicitly combining all three domains.

3. METHOD

Our approach is derived from the work by Ullrich et al. [2]. In the following, we mainly describe our extensions to this method.

3.1. Feature extraction

For each audio file under analysis, we first compute a STFT magnitude spectrogram with a window size of 46 ms (2048 samples at 44.1 kHz sample rate) and 50% overlap, and apply a mel-scaled filterbank of $n = 80$ triangular filters from 80 Hz to 16 kHz and scale magnitudes logarithmically.

Our method of generating the SSLMs (Figures 1c and 1d) is derived from work by Serrà et al. [11]. We use the MLS $X = \{\mathbf{x}_{i=1\ldots N}\}$ of N frames as described above. The lag time to cover is given by the number of frames L. In order to reduce the amount of data and processing time, the input spectra can be max-pooled along the time axis by an integer factor p,

$$\mathbf{x}'_i = max_{j=1\ldots p}\left(\mathbf{x}_{(i-1)p+j}\right). \quad (1)$$

By performing a DCT of type II on each frame with the static 0-component omitted, we arrive at a time series of MFCCs

$$\tilde{\mathbf{x}}_i = \mathrm{DCT}^{(II)}_{1\ldots n}\left(\mathbf{x}'_i\right). \quad (2)$$

We bag several frames within a time context of length m, building a time series

$$\hat{\mathbf{x}}_i = \left[\tilde{\mathbf{x}}_i^\top, \ldots, \tilde{\mathbf{x}}_{i+m}^\top\right]^\top. \quad (3)$$

The parameter m has to be chosen such that the desired reduction of noise is not outbalanced by the resulting temporal blurring. We then use a cosine distance function $\delta_{\cos}(\mathbf{x}, \mathbf{y}) = 1 - \left\langle \frac{\mathbf{x}}{\|\mathbf{x}\|}, \frac{\mathbf{y}}{\|\mathbf{y}\|} \right\rangle$ to build a $\lfloor \frac{N}{p} \rfloor \times \lfloor \frac{L}{p} \rfloor$ recurrence matrix

$$D_{i,l} = \delta_{\cos}\left(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i-l}\right), \quad l = 1 \ldots \lfloor \tfrac{L}{p} \rfloor. \quad (4)$$

Relationships between distances across this matrix are often revealed by thresholding, yielding a binary SSLM. Extending on Serrà et al. [12], who introduced adaptive thresholding with a step function, $\Theta\left(\varepsilon_{i,l} - D_{i,l}\right)$, we resort to a smooth sigmoid transfer function $\sigma(x) = 1/\left(1 + e^{-x}\right)$ for the SSLM

$$R_{i,l} = \sigma\left(1 - \frac{D_{i,l}}{\varepsilon_{i,l}}\right). \quad (5)$$

The adaptive threshold, or, in this context, equalization factor $\varepsilon_{i,l}$ is set to a quantile κ of the distances $\delta_{\cos}\left(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_{i-j}\right)$ and $\delta_{\cos}\left(\hat{\mathbf{x}}_{i-l}, \hat{\mathbf{x}}_{i-l-j}\right)$ for $j = 1 \ldots \lfloor \frac{L}{p} \rfloor$, or

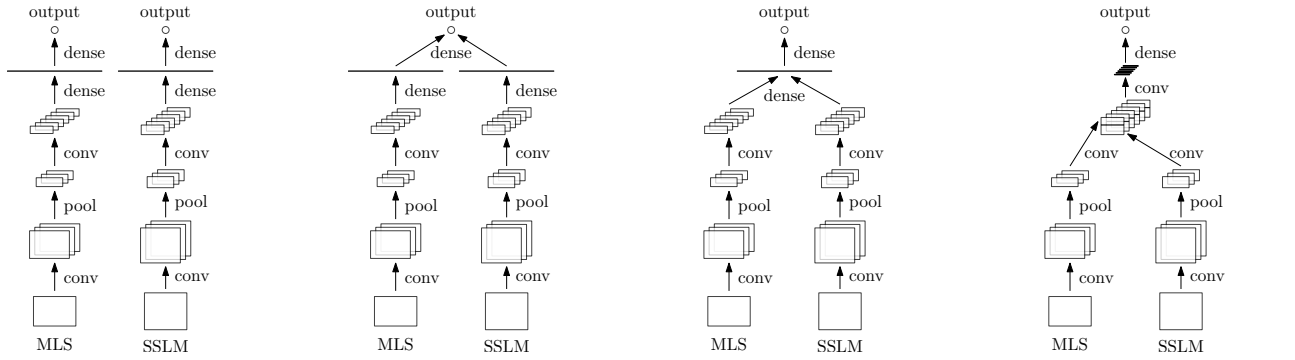
$$\varepsilon_{i,l} = Q_\kappa\left(D_{i,1}, \ldots, D_{i,\lfloor \frac{L}{p} \rfloor}, D_{i-l,1}, \ldots, D_{i-l,\lfloor \frac{L}{p} \rfloor}\right). \quad (6)$$

All indices $i < 1$ are wrapped around to $i' = i + \lfloor \frac{N}{p} \rfloor$, resulting in a time-circular SSLM R.

The use of full-length MFCCs and cosine distances was suggested by preliminary experiments, yielding better results than Euclidean metrics on the bagged original feature vectors.

3.2. Feature preprocessing

Like [2], for the MLS features, we pad the spectrogram with pink noise of -70 dB FS as needed to process the beginning and end of a piece, subsample it by taking the maximum over 6 adjacent time frames without overlap (max-pooling) and finally normalize each frequency band to zero mean and unit variance. For the SSLM features, we use circular padding and optimal pooling factors found in Section 4.3, then also normalize each lag band to zero mean and unit variance.



(a) late fusion of outputs (b) fusion in output unit (c) fusion in hidden dense layer (d) fusion in convolutional layer

Fig. 2: Four different network architectures for combining the two input features.

3.3. Convolutional Neural Network

CNNs are feed-forward networks that include *convolutional layers* computing a convolution of their input with small learned filter kernels of a given size. This allows processing large inputs with few trainable parameters, and retains the input's spatial layout. When used for binary classification, the network usually ends in one or more dense layers integrating information over the full input at once, discarding the spatial layout. Our architecture for this work is based on the one used by Ullrich et al. [2] on MLS features for their MIREX submission [13]. It has a convolutional layer of 32 8×6 kernels (8 time frames and 6 frequency bands), a max-pooling layer of 3×6, another convolution of 64 6×3 kernels, a dense layer of 128 units and a dense output layer of 1 unit.

We employ different variants of this architecture to support two input features instead of one. An obvious idea is to train two networks of the same architecture on the two features and average their output (late fusion, Fig. 2a). Instead, we can join the two output units (Fig. 2b) or even the two hidden dense layers (Fig. 2c) to obtain a single network trained on both input features. Finally, if the two features cover the same temporal context at the same resolution, we can synchronously convolve their feature maps over time (Fig. 2d).

Training is done by mini-batch gradient descent, using the same hyperparameters and tweaks as Ullrich et al. [2]. Likewise, we follow the peak-picking strategy described in [2, Section 3.4] to retrieve likely boundary locations from the network output.

4. EXPERIMENTS

4.1. Data set

For our experiments, we used the same data set as described by Ullrich et al. [2]. It is a subset of the Structural Analysis of Large Amounts of Music Information (SALAMI) database [14]. The entire data set contains over 2400 structural annotations of nearly 1400 musical recordings of different genres and origins, with about half of the annotations (779 recordings, 498 of which are doubly-annotated) being publicly available.[1] A part of this data set was also used in the "Audio Structural Segmentation" task of the annual MIREX evaluation campaign in the years 2012 through 2014.[2] Identically to [2], we used 633 musical pieces for training, 100 for validation and 487 pieces as a test set for final evaluation of our models against the published results of the various MIREX submissions.

4.2. Evaluation

For the MIREX campaign's boundary retrieval task, three different evaluation measures are used: *Hit rate* for time tolerances ± 0.5 and ± 3 seconds, and *Median deviation*. The latter computes the median time distance between each annotated boundary and its closest predicted boundary, and vice versa. The former checks which predicted boundaries fall close enough to an unmatched annotated boundary (true positives), records remaining unmatched predictions and annotations as false positives and negatives, respectively, and computes the precision, recall and F_1 scores. The Hit rate F_1 score is the measure most frequently used in the literature. In this contribution, we only evaluate for tolerances of ± 0.5 seconds where the explorable space (the distance between the lower and upper bounds exhibited in human ground-truth annotations) is greater than for ± 3 seconds, the other commonly used tolerance.

As explicated in [2], baseline scores can be estimated by using variations of regularly or randomly spaced grids as synthetic boundary estimates. For an evaluation tolerance of ± 0.5 seconds, the baseline within our test data set is $F_1 \approx 0.14$. Upper bounds, on the other hand, can be derived from the differences between two independent annotations of the same musical pieces. By analyzing the items within our test data set that have been annotated twice (439 pieces), we calculated $F_1 \approx 0.72$.

Nieto et al. [15] have identified the $F_{0.58}$ measure to be more

[1] `http://ddmal.music.mcgill.ca/datasets/salami/SALAMI_data_v1.2.zip`, accessed 2015-02-11

[2] Music Information Retrieval Evaluation eXchange, `http://www.music-ir.org/mirex`, accessed 2015-02-11

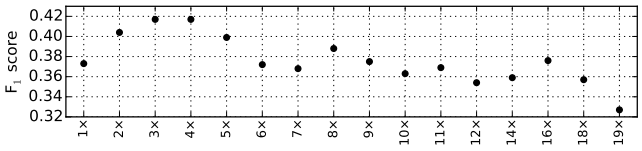


Fig. 3: F_1 scores for different lag pooling factors on SSLMs.

Pool factor / Lag bins	75 bins	100 bins	150 bins
Factor 3	0.398	**0.408**	0.399
Factor 4	0.407	0.408	0.403

Table 1: F_1 scores (means of three experiments) for different lag bin counts and pooling factors on SSLMs.

perceptually informative than the typically used F_1 measure. As this is a relatively new finding and it is not as well established as the F_1 measure (which is, e.g., used in MIREX), we base most of our evaluation, especially model selection, on the latter.

4.3. Feature optimization

To optimize parameters for the new feature, we performed a range of experiments training CNNs to predict boundaries from SSLMs alone, evaluated on the validation set.

Four parameters were fixed in advance: The block size $N = 115$, equal to the MLS of [2], input pooling $p = 2$, bagging $m = 2$, and equalization quantile $\kappa = 0.1$. Preliminary experiments showed no improvement from varying these.

As described in Section 3.1, we obtained pooled SSLMs of 21.53 fps. We computed them up to a lag of 240 s, resulting in 5168 lag bins. In comparison, the best-performing MLS of [2] are 80 frequency bins at 7.18 fps, obtained by max-pooling the original spectrograms over time. As a first measure to bring the SSLMs to manageable size, we pooled over time to match the MLS resolution. Secondly, we used a combination of max-pooling and cropping to reduce the number of lag bins to 100. We tried a range of pooling factors from 19 to 1, resulting in a lag context between 88.23 s and 4.64 s. Surprisingly, as depicted in Figure 3, best results were obtained in the regime of high resolution and a context too small to include long-term repetitions (cf. Figures 1c, 1d). To verify, we trained three networks each for the two best pooling factors (3 and 4) and 75, 100, and 150 lag bins, then averaged their evaluation scores. Table 1 shows that 100 bins seem optimal, and factors 3 and 4 are comparable. We chose factor 3 to match the temporal pooling.

With these parameters, it seems we could just compute the SSLMs with $p = 6$ up to a lag of 13.92 s instead of pooling and cropping them afterwards. However, this reduces the context for the calculation of the equalization factor ε in (6) too much, diminishing results.

4.4. Model selection

Having optimized the SSLMs, we proceeded to combine them with the MLS features. As described in Section 3.3, we devised four different network architectures that fuse the information in different processing stages. We compared these architectures in the same manner Ullrich et al. [2] obtained their final results: We trained five copies of each network and evaluated them on the test set, with the optimal peak-picking threshold found on the validation set.

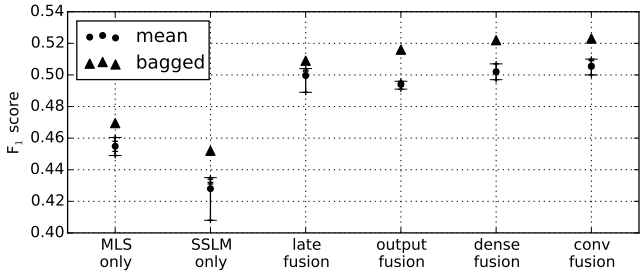


Fig. 4: Comparison of F_1 scores for MLSs and SSLMs alone, and for network architectures with different fusion modes.

Algorithm	$\mathbf{F_1}$	$\mathbf{F_{0.58}}$	**Recall**	**Precision**
Upper bound (est.)	0.72	0.72		
our result (conv fusion)	0.523	0.596	0.484	0.646
SUG1 (2014)	0.472	0.497	0.469	0.555
MP2 (2013)	0.328	0.311	0.411	0.300
MP1 (2013)	0.315	0.307	0.361	0.304
NB1 (2014)	0.301	0.273	0.421	0.254
OYZS1 (2012)	0.290	0.334	0.258	0.456
Baseline (est.)	0.14	0.19		

Table 2: Boundary recognition scores at a tolerance of ± 0.5 seconds. Comparison of our best model with the five best-performing algorithms of the MIREX campaigns 2012 through 2014.

Figure 4 shows the minimum, maximum and mean of the five F_1 scores per architecture, as well as the F_1 score obtained from bagging the networks by averaging their outputs. We can see that networks trained on MLS alone perform better than on SSLM alone, and that combining them gives a leap in performance. This shows that the two features transport different cues useful for predicting boundaries. Comparing the different fusion architectures, it seems advantageous to combine the information early on, although the differences are not significant. Further experiments have shown that also in combination with MLSs, SSLMs with a short lag context of ca. 14 s work better than those with a context of up to 88 s.

4.5. Comparison to the state of the art

Table 2 shows a comparison of our best model with the best-performing algorithms of the MIREX campaign in the years 2012 through 2014[3]. Both precision and recall rates of our model are higher than any of the state-of-the-art algorithms,

[3]For the algorithms' abbreviations, consult `http://nema.lis.illinois.edu/nema_out/mirex2012/results/struct/sal`, `.../mirex2013/...`, and `.../mirex2014/...`, accessed 2015-06-08

resulting in superior F_1 and $F_{0.58}$ scores. All scores have been calculated on our test data set.

5. DISCUSSION AND OUTLOOK

In this contribution dealing with the prediction of musically relevant structural boundaries, we have introduced a CNN model combining two different input features – mel spectrograms and our variation on self-similarity lag matrices – using four different strategies for information fusion. We have been able to show that evaluating our model on a representative subset of the SALAMI database raises the state-of-the-art scores (Table 2) from $F_1 = 0.472$ to $F_1 = 0.523$. Figure 4 illustrates that there is a significant gain from simply bagging the individual predictions to exploit information overlap within the neural network structure, especially when considering time-synchronicity of the audio features. Quite surprisingly, the network cannot take advantage of structural information contained within the lag matrices over longer time contexts. As the time lag identified as optimal is as short as the context window for the mel spectrograms, it seems that the SSLMs merely provide additional cues for novelty and homogeneity. These cues cannot easily be computed by the network from the MLS features, because it can neither perform the dot product for the cosine distances nor a long-range equalization of the similarities without considerable architectural overhead. We also suspect that annotated boundaries do not co-occur often enough with long-range structural cues such as repetitions for the network to pick these up. Artificial training data or additional preprocessing could help. Our main focus for further investigation shall lie on finding a representation of information about repetitive patterns on a more global scale that is easier to process for the network. We should also note that the space of model hyper-parameters has not been fully explored yet, concerning the audio features, network architecture and training schemes.

REFERENCES

[1] Jouni Paulus, Meinard Müller, and Anssi Klapuri, "Audio-based music structure analysis," in *Proc. of the 11th International Conference on Music Information Retrieval (ISMIR)*, 2010.

[2] Karen Ullrich, Jan Schlüter, and Thomas Grill, "Boundary Detection in Music Structure Analysis using Convolutional Neural Networks," in *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[3] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 2000.

[4] J. Paulus and A. Klapuri, "Acoustic features for music piece structure analysis," in *Proc. 11th International Conference on Digital Audio Effects (DAFx)*, 2008.

[5] Douglas Turnbull, Gert Lanckriet, Elias Pampalk, and Masataka Goto, "A supervised approach for detecting boundaries in music using difference features and boosting," in *Proc. of the 8th International Conference on Music Information Retrieval (ISMIR)*, 2007.

[6] Lie Lu, Muyuan Wang, and Hong-Jiang Zhang, "Repeating pattern discovery and structure analysis from acoustic music data," in *Proc. of the 6th ACM SIGMM international workshop on Multimedia information retrieval (MIR)*, 2004.

[7] Jouni Paulus and Anssi Klapuri, "Music structure analysis by finding repeated parts," in *Proc. of the 1st ACM workshop on Audio and music computing multimedia (AMCMM)*, 2006.

[8] Jouni Paulus and Anssi Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *Trans. Audio, Speech and Lang. Proc.*, vol. 17, no. 6, 2009.

[9] Brian McFee and Daniel P. W. Ellis, "Analyzing song structure with spectral clustering," in *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[10] Brian McFee and Daniel P. W. Ellis, "Learning to segment songs with ordinal linear discriminant analysis," in *International conference on acoustics, speech and signal processing*, 2014.

[11] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Ll. Arcos, "Unsupervised detection of music boundaries by time series structure features," in *Proc. of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[12] Joan Serrà, Xavier Serra, and Ralph G. Andrzejak, "Cross recurrence quantification for cover song identification," *New Journal of Physics*, vol. 11, no. 9, 2009.

[13] Jan Schlüter, Karen Ullrich, and Thomas Grill, "Structural segmentation with convolutional neural networks mirex submission," in *Tenth running of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2014.

[14] 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 *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[15] Oriol Nieto, Morwaread M. Farbood, Tristan Jehan, and Juan Pablo Bello, "Perceptual analysis of the f-measure for evaluating section boundaries in music," in *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.