BOUNDARY DETECTION IN MUSIC STRUCTURE ANALYSIS USING CONVOLUTIONAL NEURAL NETWORKS

Karen Ullrich **Jan Schlüter** **Thomas Grill**
Austrian Research Institute for Artificial Intelligence, Vienna
firstname.lastname@ofai.at

ABSTRACT

The recognition of boundaries, e.g., between chorus and verse, is an important task in music structure analysis. The goal is to automatically detect such boundaries in audio signals so that the results are close to human annotation. In this work, we apply Convolutional Neural Networks to the task, trained directly on mel-scaled magnitude spectrograms. On a representative subset of the SALAMI structural annotation dataset, our method outperforms current techniques in terms of boundary retrieval F-measure at different temporal tolerances: We advance the state-of-the-art from 0.33 to 0.46 for tolerances of ± 0.5 seconds, and from 0.52 to 0.62 for tolerances of ± 3 seconds. As the algorithm is trained on annotated audio data without the need of expert knowledge, we expect it to be easily adaptable to changed annotation guidelines and also to related tasks such as the detection of song transitions.

1. INTRODUCTION

The determination of the overall structure of a piece of audio, often referred to as *musical form*, is one of the key tasks in music analysis. Knowledge of the musical structure enables a variety of real-world applications, be they commercially applicable, such as for browsing music, or educational. A large number of different techniques for automatic structure discovery have been developed, see [16] for an overview. Our contribution describes a novel approach to retrieve the boundaries between the main structural parts of a piece of music. Depending on the music under examination, the task of finding such musical boundaries can be relatively simple or difficult, in the latter case leaving ample space for ambiguity. In fact, two human annotators hardly ever annotate boundaries at the exact same positions. Instead of trying to design an algorithm that works well in all circumstances, we let a Convolutional Neural Network (CNN) learn to detect boundaries from a large corpus of human-annotated examples.

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 introduce the data set used for training and testing. After presenting our main results in Section 5, we wrap up in Section 6 with a discussion and outlook.

2. RELATED WORK

In the overview paper to audio structure analysis by Paulus et al. [16], three fundamental approaches to segmentation are 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. Many segmentation algorithms follow mixed strategies. Novelty is typically computed using Self-Similarity Matrices (SSMs) or Self-Distance Matrices (SDMs) with a sliding checkerboard kernel [4], building on audio descriptors like timbre (MFCC features), pitch, chroma vectors and rhythmic features [14]. Alternative approaches calculate difference features on more complex audio feature sets [21]. In order to achieve a higher temporal accuracy in rhythmic music, audio features can be accumulated beat-synchronously. Techniques capitalizing on homogeneity use clustering [5] or state-modelling (HMM) approaches [1], or both [9, 11]. Repeating pattern discovery is performed on SSMs or SDMs [12], and often combined with other approaches [13, 15]. Some algorithms combine all three basic approaches [18].

Almost all existing algorithms are hand-designed from end to end. To the best of our knowledge, only two methods are partly learning from human annotations: Turnbull et al. [21] compute temporal differences at three time scales over a set of standard audio features including chromagrams, MFCCs, and fluctuation patterns. Training Boosted Decision Stumps to classify the resulting vectors into boundaries and non-boundaries, they achieved significant gains over a hand-crafted boundary detector using the same features, evaluated on a set of 100 pop songs. McFee et al. [13] employ Ordinal Linear Discriminant Analysis to learn a linear transform of beat-aligned audio features (including MFCCs and chroma) that minimizes the variance within a human-annotated segment while maximizing the distance across segments. Combined with a repetition feature, their method defines the current state of the art in boundary retrieval, but still involves significant manual engineering.

For other tasks in the field of Music Information Retrieval, supervised learning with CNNs has already proven

to outperform hand-designed algorithms, sometimes by a large margin [3, 6, 8, 10, 17]. In this work, we investigate whether CNNs are effective for structural boundary detection as well.

3. METHOD

We propose to train a neural network on human annotations to predict likely musical boundary locations in audio data. Our method is derived from Schlüter and Böck [17], who use CNNs for onset detection: We also train a CNN as a binary classifier on spectrogram excerpts, but we adapt their method to include a larger input context and respect the higher inaccuracy and scarcity of segment boundary annotations compared to onset annotations. In the following, we will describe the features, neural network, supervised training procedure and the post-processing of the network output to obtain boundary predictions.

3.1 Feature Extraction

For each audio file, we compute a magnitude spectrogram with a window size of 46 ms (2048 samples at 44.1 kHz) and 50% overlap, apply a mel filterbank of 80 triangular filters from 80 Hz to 16 kHz and scale magnitudes logarithmically. To be able to train and predict on spectrogram excerpts near the beginning and end of a file, we pad the spectrogram with pink noise at -70 dB as needed (padding with silence is impossible with logarithmic magnitudes, and white noise is too different from the existing background noise in natural recordings). To bring the input values to a range suitable for neural networks, we follow [17] in normalizing each frequency band to zero mean and unit variance. Finally, to allow the CNN to process larger temporal contexts while keeping the input size reasonable, we subsample the spectrogram by taking the maximum over 3, 6 or 12 adjacent time frames (without overlap), resulting in a frame rate of 14.35 fps, 7.18 fps or 3.59 fps, respectively. We will refer to these frame rates as `high`, `std` and `low`.

We also tried training on MFCCs and chroma vectors (descriptors with less continuity in the 'vertical' feature dimension to be exploited by convolution), as well as fluctuation patterns and self-similarity matrices derived from those. Overall, mel spectrograms proved the most suitable for the algorithm and performed best.

3.2 Convolutional Neural Networks

CNNs are feed-forward neural networks usually consisting of three types of layers: Convolutional layers, pooling layers and fully-connected layers. A convolutional layer computes a convolution of its two-dimensional input with a fixed-size kernel, followed by an element-wise nonlinearity. The input may consist of multiple same-sized channels, in which case it convolves each with a separate kernel and adds up the results. Likewise, the output may consist of multiple channels computed with distinct sets of kernels. Typically the kernels are small compared to the input, allowing CNNs to process large inputs with few learnable parameters. A pooling layer subsamples its two-dimensional input, possibly by different factors in the two dimensions, handling each input channel separately. Here, we only consider max-pooling, which introduces some translation invariance across the subsampled dimension. Finally, a fully-connected layer discards any spatial layout of its input by reshaping it into a vector, computes a dot product with a weight matrix and applies an element-wise nonlinearity to the result. Thus, unlike the other layer types, it is not restricted to local operations and can serve as the final stage integrating all information to form a decision.

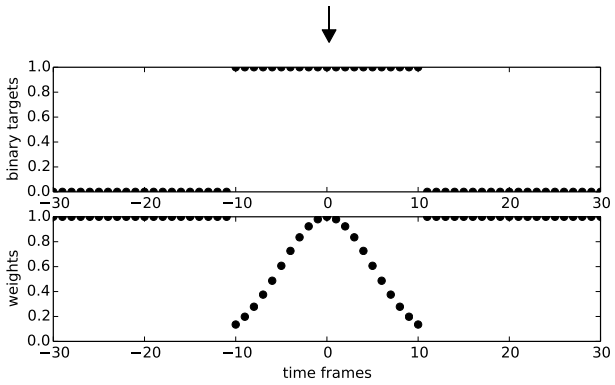

Figure 1. The arrow at the top signifies an annotated segment boundary present within a window of feature frames. As seen in the upper panel, the target labels are set to one in the environment of this boundary, and to zero elsewhere. The lower panel shows how positive targets far from the annotation are given a lower weight in training.

In this work, we fix the network architecture to a convolutional layer of 16 8×6 kernels (8 time frames, 6 mel bands, 16 output channels), a max-pooling layer of 3×6, another convolution of 32 6×3 kernels, a fully-connected layer of 128 units and a fully-connected output layer of 1 unit. This architecture was determined in preliminary experiments and not further optimized for time constraints.

3.3 Training

The input to the CNN is a spectrogram excerpt of N frames, and its output is a single value giving the probability of a boundary in the center of the input. The network is trained in a supervised way on pairs of spectrogram excerpts and binary labels. To account for the inaccuracy of the ground truth boundary annotations (as observable from the disagreement between two humans annotating the same piece), we employ what we will refer to as *target smearing*: All excerpts centered on a frame within $\pm E$ frames from an annotated boundary will be presented to the network as positive examples, weighted in learning by a Gaussian kernel centered on the boundary. Figure 1 illustrates this for $E = 10$. We will vary both the spectrogram length N and smearing environment E in our experiments. To compensate for the scarceness of positive examples, we increase their chances of being randomly selected for a training step by a factor of 3.

Training is performed using gradient descent on cross-

entropy error with mini-batches of 64 examples, momentum of 0.95, and an initial learning rate of 0.6 multiplied by 0.85 after every mini-epoch of 2000 weight updates. We apply 50% dropout to the inputs of both fully-connected layers [7]. Training is always stopped after 20 mini-epochs, as the validation error turned out not to be robust enough for early stopping. Implemented in Theano [2], training a single CNN on an Nvidia GTX 780 Ti graphics card took 50–90 minutes.

3.4 Peak-picking

At test time, we apply the trained network to each position in the spectrogram of the music piece to be segmented, obtaining a boundary probability for each frame. We then employ a simple means of peak-picking on this boundary activation curve: Every output value that is not surpassed within ± 6 seconds is a boundary candidate. From each candidate value we subtract the average of the activation curve in the past 12 and future 6 seconds, to compensate for long-term trends. We end up with a list of boundary candidates along with strength values that can be thresholded at will. We found that more elaborate peak picking methods did not improve results.

4. DATASET

We evaluate our algorithm on a subset of the Structural Analysis of Large Amounts of Music Information (SALAMI) database [20]. In total, this dataset contains over 2400 structural annotations of nearly 1400 musical recordings of different genres and origins. About half of the annotations (779 recordings, 498 of which are doubly-annotated) are publicly available. [1] A part of the dataset was also used in the "Audio Structural Segmentation" task of the annual MIREX evaluation campaign in 2012 and 2013. [2] Along with quantitative evaluation results, the organizers published the ground truth and predictions of 17 different algorithms for each recording. By matching the ground truth to the public SALAMI annotations, we were able to identify 487 recordings. These serve as a test set to evaluate our algorithm against the 17 MIREX submissions. We had another 733 recordings at our disposal, annotated following the SALAMI guidelines, which we split into 633 items for training and 100 for validation.

5. EXPERIMENTAL RESULTS

5.1 Evaluation

For boundary retrieval, the MIREX campaign uses two evaluation measures: *Median deviation* and *Hit rate*. The former measures the median distance between each annotated boundary and its closest predicted boundary or vice versa. The latter 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, then computes the precision, recall and F-measure. Since not only the temporal distance of predictions, but also the figures of precision and recall are of interest, we opted for the Hit rate at as our central measure of evaluation, computed at a temporal tolerance of ± 0.5 seconds (as in [21]) and ± 3 seconds (as in [9]). For accumulation over multiple recordings, we follow the MIREX evaluation by calculating F-measure, precision and recall per item and averaging the three measures over the items for the final result. Note that the averaged F-measure is not necessarily the harmonic mean of the averaged precision and recall. Our evaluation code is publicly available for download. [3]

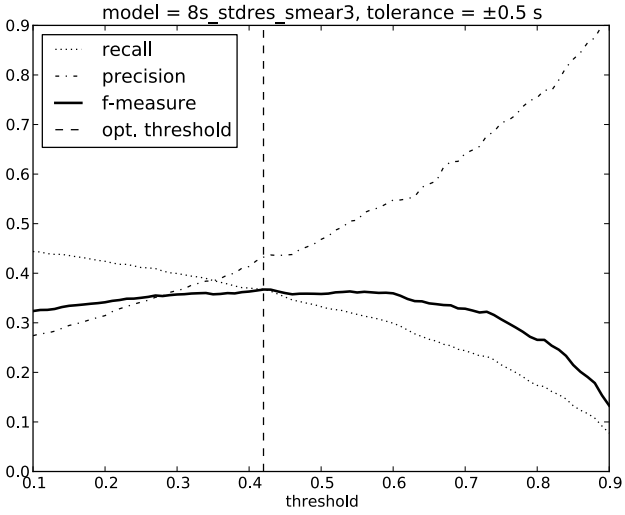

Figure 2. Optimization of the threshold shown for model `8s_std_3s` at tolerance ± 0.5 seconds. Boundary retrieval precision, recall and F-measure are averaged over the 100 validation set files.

5.2 Baseline and upper bound

Our focus for evaluation lies primarily on the F-measure. Theoretically, the F-measure is bounded by $F \in [0, 1]$, but for the given task, we can derive more useful lower and upper bounds to compare our results to. As a baseline, we use regularly spaced boundary predictions starting at time 0. Choosing an optimal spacing, we obtain an F-measure of $F_{\text{inf},3} \approx 0.33$ for ± 3 seconds tolerance, and $F_{\text{inf},0.5} \approx 0.13$ for a tolerance of ± 0.5 seconds. Note that it is crucial to place the first boundary at 0 seconds, where a large fraction of the music pieces has annotated segment boundaries. Many pieces have only few boundaries at all, thus the impact can be considerable. An upper bound F_{sup} can be derived from the insight that no annotation will be perfect given the fuzzy nature of the segmentation task. Even though closely following annotation guidelines, [4] two annotators might easily disagree on the existence or exact po-

[1] `http://ddmal.music.mcgill.ca/datasets/salami/SALAMI_data_v1.2.zip`, accessed 2014-05-02

[2] Music Information Retrieval Evaluation eXchange, `http://www.music-ir.org/mirex`, accessed 2014-04-29

[3] `http://ofai.at/research/impml/projects/audiostreams/ismir2014/`

[4] cf. the SALAMI Annotator's Guide: `http://www.music.mcgill.ca/~jordan/salami/SALAMI-Annotator-Guide.pdf`, accessed 2014-04-30

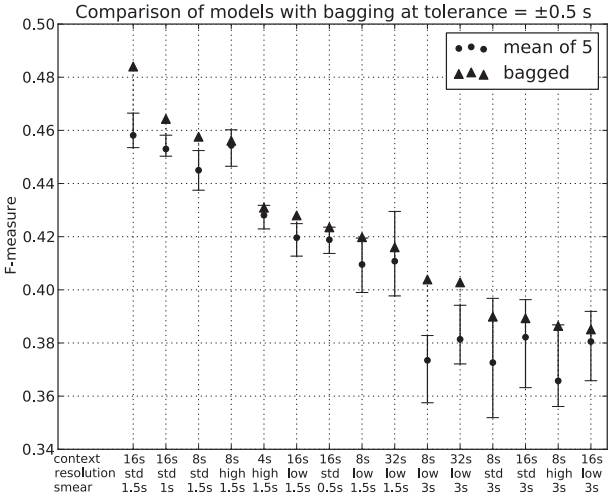

Figure 3. Comparison of different model parameters (context length, resolution and target smearing) with respect to mean F-measure on our validation set at ± 0.5 seconds tolerance. Mean and minimum-maximum range of five individually trained models for each parameter combination are shown, as well as results for bagging the five models.

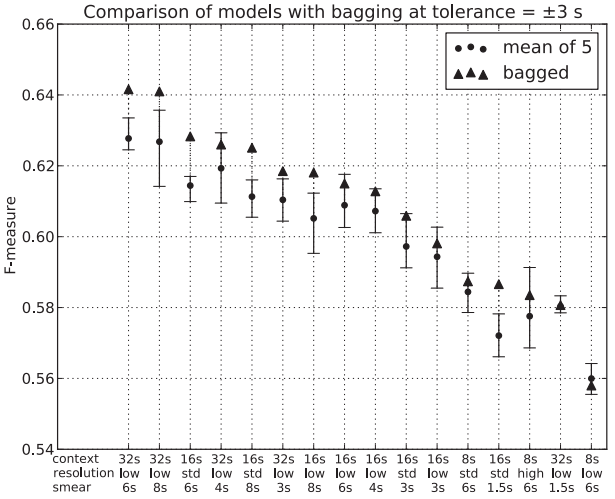

Figure 4. Comparison of different model parameters (context length, resolution and target smearing) with respect to mean F-measure on our validation set at ± 3 seconds tolerance. Mean and minimum-maximum range of five individually trained models for each parameter combination are shown, as well as results for bagging the five models.

sitions of segment boundaries. By analyzing the items in the public SALAMI dataset that have been annotated twice (498 pieces in total), we calculated $F_{\text{sup},3} \approx 0.76$ for ± 3 seconds tolerance, and $F_{\text{sup},0.5} \approx 0.67$ for ± 0.5 seconds tolerance. Within our evaluation data subset (439 double-annotations), the results are only marginally different with $F_{\text{sup},0.5} \approx 0.68$.

5.3 Threshold optimization

Peak-picking, described in Section 3.4, delivers the positions of potential boundaries along with their probabilities, as calculated by the CNN. The application of a threshold to those probabilities rejects part of the boundaries, affecting the precision and recall rates and consequently the F-measure we use for evaluation. Figure 2 shows precision and recall rates as well as the F-measure as a function of the threshold for the example of the `8s_std_3s` model (8 seconds of context, standard resolution, target smearing 3 seconds) at ± 0.5 seconds tolerance, applied to the 100 files of the validation data set. By locating the maximum of the F-measure we retrieve an estimate for the optimum threshold which is specific for each individual learned model. Since the curve for the F-measure is typically flat-topped for a relatively wide range of threshold values, the choice of the actual value is not very delicate.

5.4 Temporal context investigation

It is intuitive to assume that the CNN needs a certain amount of temporal context to reliably judge the presence of a boundary. Furthermore, the temporal resolution of the input spectra (Section 3.1) and the applied target smearing (Section 3.3) is expected to have an impact on the temporal accuracy of the predictions. See Figure 3 and Figure 4 for comparisons of these model parameters, for tolerances ± 0.5 seconds and ± 3 seconds, respectively. Each bar in the plots represents the mean and minimum-maximum range of five individual experiments with different random initializations. For the case of only ± 0.5 seconds of acceptable error, we conclude that target smearing must also be small: A smearing width of 1 to 1.5 seconds performs best. Low temporal spectral resolution tends to diminish results, and the context length should not be shorter than 8 seconds. For ± 3 seconds tolerance, context length and target smearing are the most influential parameters, with the F-measure peaking at 32 seconds context and 4 to 6 seconds smearing. Low temporal resolution is sufficient, keeping the CNN smaller and easier to train.

5.5 Model bagging

As described in Section 5.4, for each set of parameters we trained five individual models. This allows us to improve the performance on the given data using a statistical approach: *Bagging*, in our case averaging the outputs of multiple identical networks trained from different initializations before the peak-picking stage, should help to reduce model uncertainty. After again applying the above described threshold optimization process on the resulting boundaries, we arrived at improvements of the F-measure of up to 0.03, indicated by arrow tips in Figures 3 and 4. Tables 1 and 2 show our final best results after model bagging for tolerances ± 0.5 seconds and ± 3 seconds, respectively. The results are set in comparison with the algorithms submitted to the MIREX campaign in 2012 and 2013, and the lower and upper bounds calculated from the annotation ground-truth (see Section 5.2).

Algorithm	F-measure	Precision	Recall
Upper bound (est.)	0.68		
16s_std_1.5s	**0.4646**	0.5553	0.4583
MP2 (2013)	0.3280	0.3001	0.4108
MP1 (2013)	0.3149	0.3043	0.3605
OYZS1 (2012)	0.2899	0.4561	0.2583
32s_low_6s	0.2884	0.3592	0.2680
KSP2 (2012)	0.2866	0.2262	0.4622
SP1 (2012)	0.2788	0.2202	0.4497
KSP3 (2012)	0.2788	0.2202	0.4497
KSP1 (2012)	0.2788	0.2201	0.4495
RBH3 (2013)	0.2683	0.2493	0.3360
RBH1 (2013)	0.2567	0.2043	0.3936
RBH2 (2013)	0.2567	0.2043	0.3936
RBH4 (2013)	0.2567	0.2043	0.3936
CF5 (2013)	0.2128	0.1677	0.3376
CF6 (2013)	0.2101	0.2396	0.2239
SMGA1 (2012)	0.1968	0.1573	0.2943
MHRAF1 (2012)	0.1910	0.1941	0.2081
SMGA2 (2012)	0.1770	0.1425	0.2618
SBV1 (2012)	0.1546	0.1308	0.2129
Baseline (est.)	0.13		

Table 1. Boundary recognition results on our test set at ± 0.5 seconds tolerance. Our best result is emphasized and compared with results from the MIREX campaign in 2012 and 2013.

Algorithm	F-measure	Precision	Recall
Upper bound (est.)	0.76		
32s_low_6s	**0.6164**	0.5944	0.7059
16s_std_1.5s	0.5726	0.5648	0.6675
MP2 (2013)	0.5213	0.4793	0.6443
MP1 (2013)	0.5188	0.5040	0.5849
CF5 (2013)	0.5052	0.3990	0.7862
SMGA1 (2012)	0.4985	0.4021	0.7258
RBH1 (2013)	0.4920	0.3922	0.7482
RBH2 (2013)	0.4920	0.3922	0.7482
RBH4 (2013)	0.4920	0.3922	0.7482
SP1 (2012)	0.4891	0.3854	0.7842
KSP3 (2012)	0.4891	0.3854	0.7842
KSP1 (2012)	0.4888	0.3850	0.7838
KSP2 (2012)	0.4885	0.3846	0.7843
SMGA2 (2012)	0.4815	0.3910	0.6965
RBH3 (2013)	0.4804	0.4407	0.6076
CF6 (2013)	0.4759	0.5305	0.5102
OYZS1 (2012)	0.4401	0.6354	0.4038
SBV1 (2012)	0.4352	0.3694	0.5929
MHRAF1 (2012)	0.4192	0.4342	0.4447
Baseline (est.)	0.33		

Table 2. Boundary recognition results on our test set at ± 3 seconds tolerance. Our best result is emphasized and compared with results from the MIREX campaign in 2012 and 2013.

6. DISCUSSION AND OUTLOOK

Employing Convolutional Neural Networks trained directly on mel-scaled spectrograms, we are able to achieve boundary recognition F-measures strongly outperforming any algorithm submitted to MIREX 2012 and 2013. The networks have been trained on human-annotated data, considering different context lengths, temporal target smearing and spectrogram resolutions. As we did not need any domain knowledge for training, we expect our method to be easily adaptable to different 'foci of annotation' such as, e.g., determined by different musical genres or annotation guidelines. In fact, our method is itself an adaption of a method for onset detection [17] to a different time focus.

There are a couple of conceivable strategies to improve the results further: With respect to the three fundamental approaches to segmentation described in Section 1, the CNNs in this work can only account for novelty and homogeneity, which can be seen as two sides of the same medal. To allow them to leverage repetition cues as well, the vectorial repetition features of McFee et al. [13] might serve as an additional input. Alternatively, the network could be extended with recurrent connections to yield a Recurrent CNN. Given suitable training data, the resulting memory might be able to account for repeating patterns. Secondly, segmentation of musical data by humans is not a trivially sequential process but inherently hierarchical. The SALAMI database actually provides annotations on two levels: A coarse one, as used in the MIREX campaign, but also a more fine-grained variant, encoding subtler details of the temporal structure. It could be helpful to feed both levels to the CNN training, weighted with respect to the significance. Thirdly, we leave much of the data preprocessing to the CNN, very likely using up a considerable part of its capacity. For example, the audio files in the SALAMI collection are of very different loudness, which could be fixed in a simple preprocessing step, either on the whole files, or using some dynamic gain control. Similarly, many of the SALAMI audio files start or end with noise or background sounds. A human annotator easily recognizes this as not belonging to the actual musical content, ignoring it in the annotations. The abrupt change from song-specific background noise to our pink noise padding may be mistaken for a boundary by the CNN, though. Therefore it could be worthwhile to apply some intelligent padding of appropriate noise or background to provide context at the beginnings and endings of the audio. And finally, we have only explored a fraction of the hyperparameter space regarding network architecture and learning, and expect further improvements by a systematic optimization of these.

Another promising direction of research is to explore the internal processing of the trained networks, e.g., by visualization of connection weights and receptive fields [19]. This may help to understand the segmentation process as well as differences to existing approaches, and to refine the network architecture.

7. ACKNOWLEDGMENTS

This research is funded by the Federal Ministry for Transport, Innovation & Technology (BMVIT) and the Austrian Science Fund (FWF): TRP 307-N23. Many thanks to the anonymous reviewers for your valuable feedback!

8. REFERENCES

[1] J.-J. Aucouturier and M. Sandler. Segmentation of musical signals using hidden markov models. In *Proc. AES 110th Convention*, May 2001.

[2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proc. of the Python for Scientific Computing Conference (SciPy)*, June 2010.

[3] S. Dieleman, P. Braken, and B. Schrauwen. Audio-based music classification with a pretrained convolutional network. In *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Miami, FL, USA, October 2011.

[4] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME'00)*, volume 1, pages 452–455 vol.1, 2000.

[5] J. T. Foote and M. L. Cooper. Media segmentation using self-similarity decomposition. In *Proc. of The SPIE Storage and Retrieval for Multimedia Databases*, volume 5021, pages 167–175, San Jose, California, USA, January 2003.

[6] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, October 2011.

[7] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, 2012.

[8] E. J. Humphrey and J. P. Bello. Rethinking automatic chord recognition with convolutional neural networks. In *Proc. of the 11th Int. Conf. on Machine Learning and Applications (ICMLA)*, volume 2, Boca Raton, FL, USA, December 2012. IEEE.

[9] M. Levy and M. Sandler. Structural segmentation of musical audio by constrained clustering. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):318–326, Feb 2008.

[10] T. LH. Li, A. B. Chan, and A. HW. Chun. Automatic musical pattern feature extraction using convolutional neural network. In *Proc. of the Int. MultiConf. of Engineers and Computer Scientists (IMECS)*, Hong Kong, March 2010.

[11] Beth Logan and Stephen Chu. Music summarization using key phrases. In *In Proc. IEEE ICASSP*, pages 749–752, 2000.

[12] Lie Lu, Muyuan Wang, and Hong-Jiang Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 275–282, New York, NY, USA, 2004. ACM.

[13] B. McFee and D. P. W. Ellis. Learning to segment songs with ordinal linear discriminant analysis. In *International conference on acoustics, speech and signal processing*, ICASSP, 2014.

[14] Jouni Paulus and Anssi Klapuri. Acoustic features for music piece structure analysis. In *Conference: 11th International Conference on Digital Audio Effects (Espoo, Finland)*, 2008.

[15] 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*, AMCMM '06, pages 59–68, New York, NY, USA, 2006. ACM.

[16] Jouni Paulus and Anssi Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *Trans. Audio, Speech and Lang. Proc.*, 17:12, 2009.

[17] J. Schlüter and S. Böck. Improved musical onset detection with convolutional neural networks. *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[18] Joan Serra, Meinard Müller, Peter Grosche, and Josep Ll. Arcos. Unsupervised detection of music boundaries by time series structure features. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1613–1619. Association for the Advancement of Artificial Intelligence, 2012.

[19] Karen Simonyan and Andrea Vedaldi and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *CoRR*, abs/1312.6034, 2013.

[20] 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 *Proceedings of the International Society for Music Information Retrieval Conference*, pages 555–560, 2011.

[21] Douglas Turnbull and Gert Lanckriet. A supervised approach for detecting boundaries in music using difference features and boosting. In *In Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 42–49, 2007.