MUSIC BOUNDARY DETECTION USING FULLY CONVOLUTIONAL NEURAL NETWORKS

Jeongmin Liu[1] **Kyeongseok Jeong**[2] **Hyunjun Lee**[3]
jeongmin96@kaist.ac.kr chungks603@kaist.ac.kr hjlee9605@kaist.ac.kr
Korea Advanced Institute of Science and Technology[123]

ABSTRACT

Music boundary detection is a basic level task of the music structure analysis in the field of music information retrieval. Deep neural network-based approach is widely used for the task in recent. We propose a method using fully convolutional neural networks (FCNs) that don't require fixed-size input and output. The FCN takes the mel-scaled log magnitude spectrogram of a full song in order to consider long temporal information, and estimates the probability that boundaries occur for every time frames. From the estimated boundary score, peak-picking algorithm predicts which time frames include boundaries. The FCN we proposed is trained and validated by a subset of SALAMI v2.0 dataset. We evaluate the proposed method with our own test set consists of 23 songs, which are Korean pop songs mostly. Test result is mainly given with F1 scores and analyzed according to genres. The result is not strictly compared with other methods, but the feasibility of the proposed method is illustrated in this paper.

1. INTRODUCTION

Music structure analysis is one of the major tasks in the music information retrieval. A song can be segmented into functional sections, such as verse and chorus, or smaller sections than that. To accomplish those tasks, musical boundary detection should be preceded. With an advent of deep neural networks (DNNs), algorithms have achieved high performance.

In Grill's paper [2], convolutional neural networks (CNNs) containing fully-connected layers are employed. Its input features are mel-scale log magnitude spectrogram (MLS) and self-similarity lag matrices, and its output is the probability that the middle frame is a boundary. Because of the fully-connected layers, the input features are only from fixed-size context frames, not from a full song.

Unlike Grill's method [2], our proposed method uses the fully convolutional neural networks (FCNs) for taking a full song as an input. It can make the model detects musical boundaries considering long temporal information. This paper doesn't include any strict comparison with other existing methods, but shows the feasibility of the proposed method by roughly comparing its performance with that of Grill's method [2].

The structure of the paper is as followings: our overall proposed method is illustrated in Section 2. Section 3 presents the experiment setup and result analysis. Finally, the entire content of the paper is wrapped up in Section 4.

2. PROPOSED METHOD

2.1 Feature Extraction And Data Augmentation

To calculate an MLS from the audio signal, the Hann window of 46 ms length and 2048-points FFT are used. The hop size is 1024, and 128 mel-scaled filterbanks are used.

All possible combinations of three augmentation types are applied to original features. First, +1 and -1 step pitch-shifting is applied. Second, -24 dBFS of the white noise is added. Third, SpecAugment [4] is applied: up to 15 randomly chosen mel bins are filled with zero.

2.2 Fully Convolutional Neural Network (FCN)

We utilize a FCN based on U-Net [6] as shown in Figure 1. There are three differences between the original U-Net and our model. First, because our desired output is 1D, our model has global average pooling with respect to the mel axis after the last convolutional layer. Second, skipped features are summed to decoded features in our model because that shows better results than they are concatenated. Third, our model uses convolutional kernels with long time-axis lengths. 3×11 kernels and 1×5 kernels are used to capture long temporal features.

2.3 Learning Target and Loss Function

The DNN is optimized to estimate boundary scores from given MLS. At every training epoch, one annotation is randomly selected from two annotation data in SALAMI. From the annotation, the frame-wise labels are created: the label is 1 if boundary occurs in the frame, and 0 if boundary does not occur. Boundary scores are the frame-wise labels smeared by the 31-length Gaussian kernel [9] as shown in Figure 2.

The loss function is weighted binary cross entropy. In the target boundary score y's, zero values occur more frequently than non-zero values. Hence, the following weight

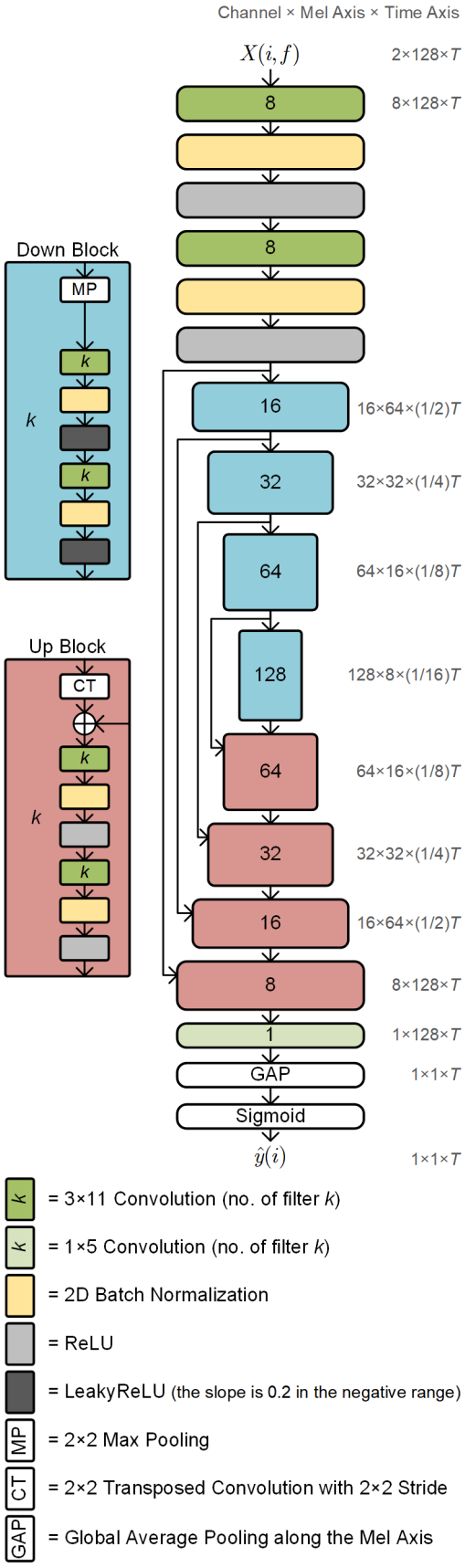

Figure 1: The FCN for the proposed method based on U-Net [6].

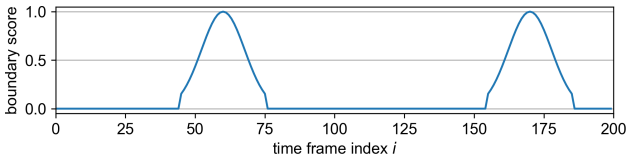

Figure 2: An example of boundary score.

function is needed:

$$W(a) = \begin{cases} 1, & a > 0 \\ \frac{K_0}{K - K_0}, & a = 0 \end{cases}, \tag{1}$$

where K is the total number of all time frames in the training set and K_0 is the total number of all frame index i's that satisfy $y(i) = 0$ in all y's of the training set. For predicted boundary score $\hat{y}$, the loss function is defined by

$$L(\hat{y}, y) = -\frac{1}{T} \sum_{i=0}^{T-1} W(y(i)) \left[(1 - y(i)) \ln(1 - \hat{y}(i)) + y(i) \ln \hat{y}(i) \right], \tag{2}$$

where T is the number of time frames in y.

2.4 Peak-Picking Algorithm

Peak-picking from estimated boundary score is performed to decide which frame is a boundary. First, the set of boundary candidate indices $\tilde{\mathcal{B}}$ consists of frame indices whose estimated score values are local maximums within ± 6 seconds [9]. For the sample rate f_s, the hop size L, and the round operator $\lfloor \cdot \rceil$, the number of time frames in 6 seconds is expressed as $N = \lfloor 6 f_s / L \rceil - 1$. Hence, $\tilde{\mathcal{B}}$ is defined by

$$\tilde{\mathcal{B}} = \left\{ i \,\middle|\, i = \underset{i-N \le k \le i+N}{\operatorname{argmax}} \hat{y}(k) \right\}. \tag{3}$$

Second, to predict which candidates are boundary, we propose defining the threshold $\bar{y}$ by the mean of candidates' estimated score values. It can be expressed as $\bar{y} = \sum_{i \in \tilde{\mathcal{B}}} \hat{y}(i) / |\tilde{\mathcal{B}}|$. Finally, the set of predicted boundaries $\hat{\mathcal{B}}$ consists of the candidates whose boundary scores are greater than the threshold $\bar{y}$:

$$\hat{\mathcal{B}} = \left\{ i \,\middle|\, i \in \tilde{\mathcal{B}},\ \hat{y}(i) \ge \bar{y} \right\}. \tag{4}$$

3. EXPERIMENTS AND RESULTS

3.1 Dataset

We use a part of SALAMI v2.0 dataset [7] whose audio data is publicly opened. 373 songs are used for training and 66 songs are used for validation. For the test set, we annotate 23 songs from 8 different genres following the annotator's guide [1] provided by SALAMI. 44.1 kHz sample rate is used for all the audio data.

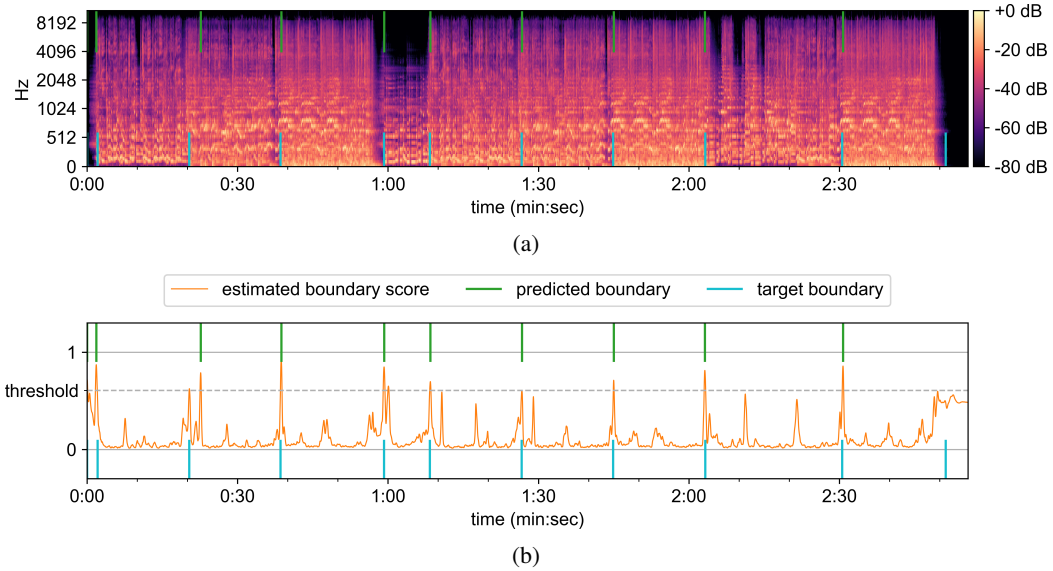

Figure 3: The mel spectrogram (a) and the estimated boundary score (b) of "Take The Dive by Jonghyun", which is one of the test set. Both (a) and (b) show its target boundary and predicted boundary.

3.2 Evaluation Metrics

The evaluation metric is hit-rate with time tolerances of $\pm$ 0.5 s [8] and $\pm$ 3.0 s [3]. It checks whether predicted boundaries are matched to the target within a time tolerance. Matched and unmatched predictions are considered as true positives and false positives respectively, and unmatched true boundaries are considered as false negatives. Precision, recall, and F1 scores are calculated with these above. The evaluation is proceeded with the Python package `mir_eval` [5].

3.3 Results

Figures 3a and 3b illustrates the mel spectrogram and the estimated boundary score of a song in the test set. Both figures show the target boundary and the predicted boundary. The DNN predicts only two boundaries incorrectly.

The F1 scores with time tolerances $\pm$ 0.5 s and $\pm$ 3.0 s according to genres are shown in Figures 4a and 4b respectively. The circle marker means the average, the upper error marker means the maximum, and the lower error marker means the minimum. The total mean of F1 scores with time tolerance $\pm$ 0.5 s is 0.432, and that with time tolerance $\pm$ 3.0 s is 0.609. The strict comparison between the proposed method and Grill's method [2] is impossible because the dataset is different. However, the proposed method can be thought of as showing similar performance to Grill's method [2]. Considering the small training set is used and the proposed method uses only MLS feature, the result is quite meaningful. The table for precision, recall, and F1 score are presented in Appendix.

There are two problems that make it difficult for the DNN to detect boundaries. One is the incomplete-bar problem. As shown in Figures 5a and 5c, we annotate the first beat of the bar right after an incomplete bar as a boundary, but the DNN predicts the beginning of an incomplete bar as a boundary. It occurs a lot in "Gravity by John Mayer" and

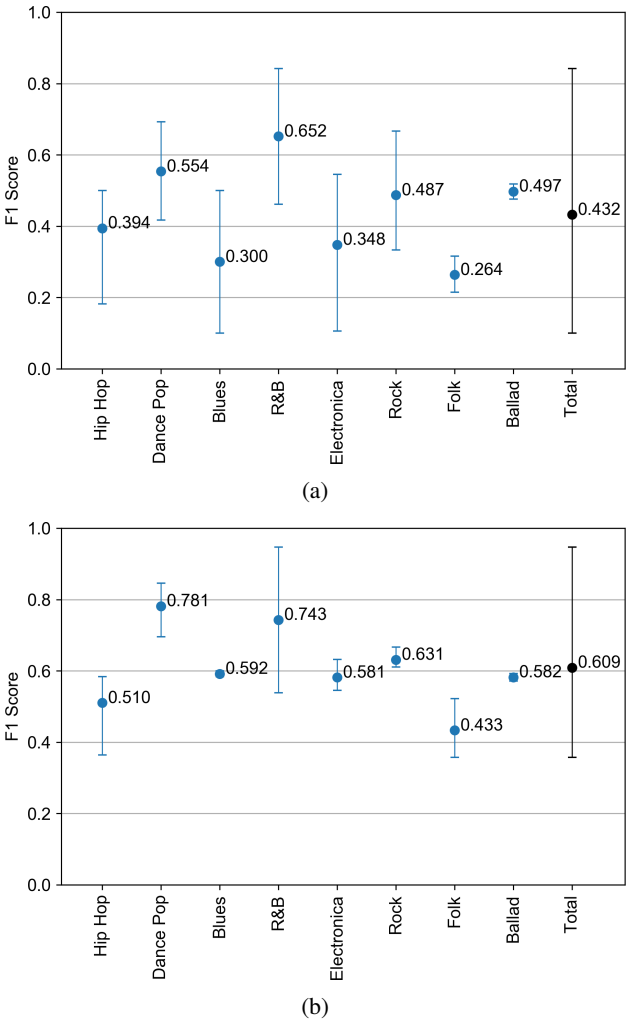

Figure 4: F1 scores of test set according to genres. Tolerance is 0.5 s [8] for (a) and 3.0 s [3] for (b).

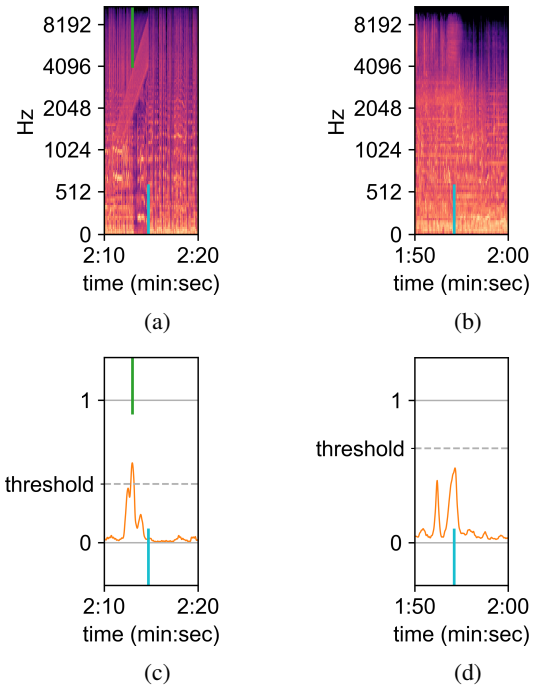

Figure 5: Unmatched predicted boundary and target plotted on MLS and estimated boundary score, each in column-wise. (a) and (c) are for "Amor Fati by Yonja Kim." (b) and (d) are for "Gwanggyeonshidae (狂犬時代) by Jaurim."

"Amor Fati by Kim Yonja", so their F1 scores with the time tolerance $\pm$ 0.5 s are the minimum in Blues and Electronica respectively. Because the prediction difference caused by incomplete bars doesn't exceed 3.0 s usually, the time tolerance $\pm$ 3.0 s makes those cases the correct predictions. Therefore, F1 scores with the time tolerance $\pm$ 3.0 s for Blues and Electronica show less deviations than F1 scores with the time tolerance $\pm$ 0.5 s. The other problem is about smooth changes of musical ideas. The DNN tends not to be able to detect boundaries where smooth changes occur. An example is a part of "Gwanggyeonshidae (狂犬時代) by Jaurim" shown in Figures 5b and 5d. To relieve those problems, the annotators should annotate boundaries only at the first beat of bars and the DNN should be trained with the musical tempo information.

4. CONCLUSION

We use an FCN based on U-Net for music boundary detection. MLS of a full song is used for the input, the frame-wise boundary scores with Gaussian kernel of a full song is used for the desired output. Training and validation is processed with small dataset: only publicly opened audio and annotation data in SALAMI. Even though the proposed DNN model is trained by small dataset and doesn't require input features other than MLS feature, it shows meaningful F1 scores roughly compared to Grill's method [2]. Incomplete bars in music and smooth change of musical ideas make the DNN confused to detect boundaries. To solve those problems, boundary detection synchronized to the musical tempo might be needed.

5. REFERENCES

[1] Salami: Structural analysis of large amounts of music information - annotator's guide. accessed 2019-06-01.

[2] Thomas Grill and Jan Schluter. Music boundary detection using neural networks on spectrograms and self-similarity lag matrices. In *the 23rd European Signal Processing Conference*, pages 1296–1300, 2015.

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

[4] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

[5] 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 *Proc. of the 15th International Society for Music Information Retrieval Conference*. Citeseer, 2014.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention 2015*, volume 9351, pages 234–241. Springer International Publishing, 2015.

[7] 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*, volume 11, pages 555–560. Miami, FL, 2011.

[8] Douglas Turnbull, Gert RG 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 Society for Music Information Retrieval Conference*, pages 51–54, 2007.

[9] Karen Ullrich, Jan Schluter, and Thomas Grill. Boundary Detection in Music Structure Analysis Using Convolutional Neural Networks. Number Ismir, pages 417–422, 2014.

6. APPENDIX

GENRE	Precision	Recall	F1
Hip-hop	0.380	0.421	0.394
Dance pop	0.558	0.551	0.554
Blues	0.300	0.300	0.300
R&B	0.694	0.614	0.652
Electronica	0.322	0.379	0.348
Rock	0.470	0.507	0.487
Folk	0.229	0.319	0.264
Ballad	0.461	0.542	0.497
TOTAL	0.420	0.451	0.432

(a)

GENRE	Precision	Recall	F1
Hip-hop	0.490	0.546	0.510
Dance pop	0.791	0.775	0.781
Blues	0.592	0.592	0.592
R&B	0.792	0.700	0.743
Electronica	0.547	0.622	0.581
Rock	0.606	0.660	0.631
Folk	0.371	0.537	0.433
Ballad	0.539	0.633	0.582
TOTAL	0.591	0.638	0.609

(b)

Table 1: Mean value of precision, recall, and F1 scores according to genres and in total. Tolerance is 0.5 s [8] for (a) and 3.0 s [3] for (b).