Quality Assessment of DNN-Based Algorithms for Music Boundary Detection

Aneeka Azmat*†, Li Su*†, ChengHsin Hsu†‡

*Social Networks and HumanCentered Computing,
Taiwan International Graduate Program,
Institute of Information Science, Academia Sinica, Taiwan

†Institute of Information Systems and Applications,
National Tsing Hua University, Hsinchu City, Taiwan

‡Department of Computer Science,
National Tsing Hua University, Hsinchu City, Taiwan

Emails: aneekaazmat89@iis.sinica.edu.tw, lisu@iis.sinica.edu.tw, chsu@cs.nthu.edu.tw

***Abstract*— Music boundary detection in music structure analysis has traditionally relied on objective metrics that may not align with human perception. This paper introduces a user study driven evaluation pipeline combining three stage evaluation: objective performance, subjective listener ratings and meta evaluation of metric validity. Audio is first converted into normalized log-mel spectrograms augmented with SpecAugment. Lightweight deep neural network (DNN) algorithms: convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM), and convolutional recurrent neural network (CRNN) are trained to produce boundary-confidence curves, from which discrete timestamps are extracted by empirically tuned thresholds. Objective evaluation is measured by precision, recall, F1-score and Information Gain. Subjective evaluation is computed by mean opinion scores (MOS) collected from ten music experts using our web-based graphical user interface. Meta evaluation uses Pearson correlation to reveal which metric best detects listener judgments. On the SALAMI dataset, CNN with SpecAugment achieves precision of 0.50 and F1-score of 0.42 at a 0.5-second tolerance, and Information Gain shows the strongest listener alignment with a correlation of 0.60.**

***Index Terms*—Music structure analysis, music boundary detection, objective evaluation, subjective evaluation, and meta evaluation.**

I. INTRODUCTION

Music structure analysis (MSA), a sub-field of music information retrieval, focuses on identifying meaningful structural transitions in audio, such as changes between verses, choruses, or bridges [1]. Accurate structure boundary detection plays a critical role in enabling downstream applications such as form-aware music generation, adaptive game soundtracks, and educational music tools [2]. However, the evaluation of MSA algorithms is a complicated task because the fine-grained structure of music is music knowledge intensive yet subjective. Evaluation methods for MSA have been widely discussed [3]. In this work, we discuss the evaluation strategy of music strcuture analysis. To facilitate the discussion, we consider lightweight boundary detection algorithms are deployable to mobile and embedded devices. We focus on convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM), and convolutional recurrent neural network (CRNN), instead of computationally expensive transformer.

A key objective design of these music boundary detection algorithms is to optimize the perceptual quality of detected boundaries for listeners. While subjective evaluations with mean opinion score (MOS) are commonly used in perceptual quality [4] assessment of music boundary detection, carrying out user studies is unfortunately time-consuming, tedious, and expensive. Therefore, prior studies [5]–[9] had to resort to objective evaluations using metrics like precision, recall, and F1-score which may not always align with how listeners perceive musical structure [10]. Hence, an objective quality metric that can better predict the subjective quality results of music boundary detection is needed.

In this paper, we fill the above mentioned gap in two steps:

- First, we design and carry out a user study on multiple music boundary detection DNN-based algorithms. In particular, we recruit ten music experts, aged between 24 and 28, to evaluate the music boundaries identified by several DNN-based algorithms. The subjective evaluation results from these experts are then analyzed using MOS to quantify the quality of the detected boundaries. To the best of our knowledge, such a user study has never been done in the literature.
- Second, we conduct a meta evaluation to investigate how well each of the objective quality metrics could predict the MOS results. In particular, we consider four objective quality metrics: 1) precision, which is the fraction of predicted boundaries that lie within a tolerance window of ground-truth boundaries, 2) recall, which the fraction of ground-truth boundaries that are recovered by predicted boundaries within a tolerance window, 3) F1-score, which the harmonic mean of precision and recall, balancing both false positives and false negatives, 4) and Information Gain (IG), which is the normalized Kullback–Leibler divergence of the boundary-error histogram against a

979-8-3315-7206-8/25/$31.00

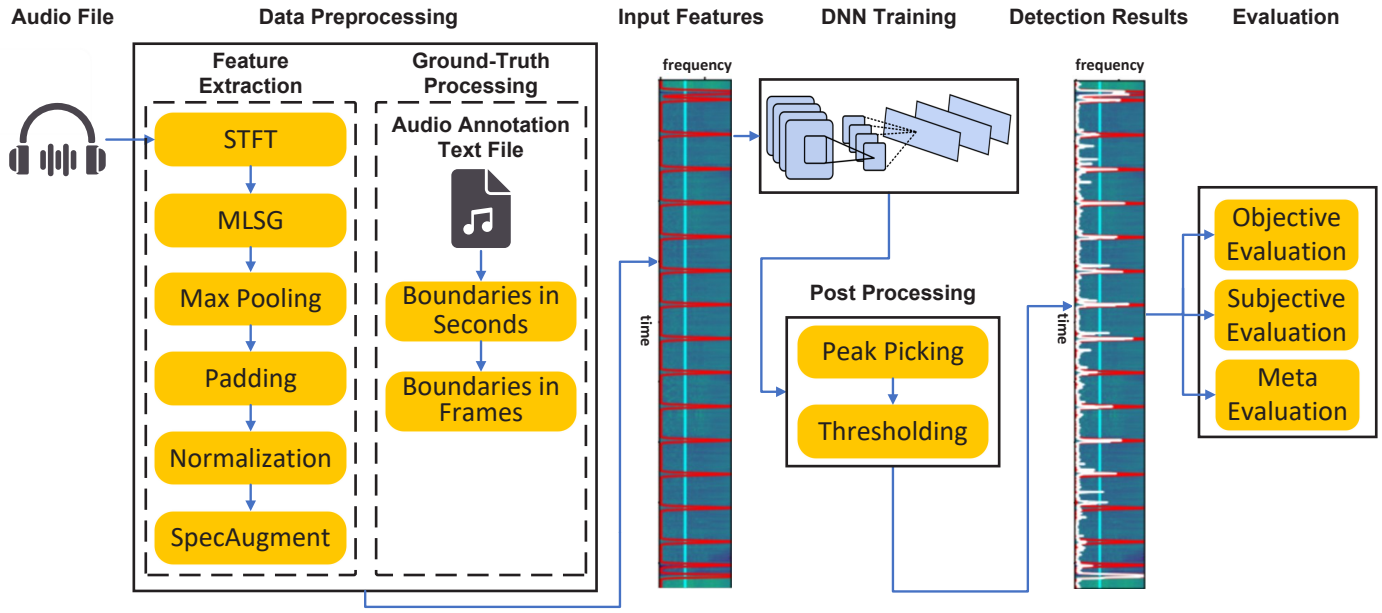

Fig. 1. Our pipeline comes with four stages of 1) data preprocessing, 2) DNN training, 3) data postprocessing, and 4) evaluation. The red line on the mel-spectrogram feature represents the labels, while the white line represents the detection results.

uniform distribution, quantifying how informative the predictions are beyond chance [3]. For the meta evaluation, Pearson's correlation coefficient is used to determine which of the above mentioned objective quality metrics predicts the perceptual quality in MOS the best.

Our evaluations reveal multiple insights: (i) CNN performs the best objectively and subjectively. It achieves the highest precision (0.50), F1 (0.42), and Pearson's r = 0.60 with MOS, surpassing BiLSTM and CRNN. (ii) Information Gain best matches expert judgment: For example, IG's correlation with MOS exceeds that of precision by 0.36, recall by 0.15, and F1 by 0.10. (iii) Our lightweight algorithms run in real time. Our DNN-based algorithms process 1 s of audio in at most 10.1 ms using a single CPU core.

II. LITERATURE REVIEW

Music boundary detection has been done using DNNs trained on log-mel-spectrograms, chroma features, or Mel-frequency cepstral coefficients (MFCCs) [5]–[7]. In particular, Ullrich *et al.* [5] adapted CNN for boundary detection using overlapped mel-spectrogram features. Subsequent works combined convolutional front-ends with LSTM layers into CRNN to capture both local and sequential structure [8]. BiLSTM models have also been explored for finer temporal transitions [7]. We also employ these lightweight DNNs which better fit the resources-constrained mobile and embedded devices.

More computationally expensive transformers have also been adopted for music boundary detection, e.g, Wang *et al.* [9] and Kim *et al.* [11] proposed transformer-based algorithms which required larger annotated datasets and higher compute demands, making them less suitable for mobile and embedded devices.

Nieto *et al.* [4] showed that F1-score does not always match listener judgments, and that precision often correlates better with perceptual accuracy than recall. Inspired by their work, we conduct a user study to identify the objective metric that best predicts the subjective quality of music boundary detection in this paper.

III. PROPOSED PIPELINE

In this section, we present our proposed pipeline, which is general and can be adopted in future research.

A. Overview

To facilitate thorough comparison among various music boundary detection algorithms, we propose a general pipeline with four stages as shown in Fig. 1. In particular, the raw audio first goes through the data preprocessing stage for mel-spectrogram features. These features serve as the input to train the DNN-based algorithms which transform mel-spectrogram features into continuous boundary signals. The data postprocessing stage then converts these signals into discrete detention results. The last stage is evaluation, consisting of 1) objective evaluations, 2) subjective evaluation, and 3) meta evaluation on various DNN-based boundary detection algorithms.

We notice that this pipeline exhibits generality and holds potential for adoption in future research efforts. Building on this, we present a reference design of the pipeline, detailing each stage in the remainder of this section.

B. Preprocessing

Our reference design of the data preprocessing stage has two branches:

Feature Extraction: We perform six steps to get the final input features 1) a short-time Fourier transform (STFT) to convert the raw waveform into the time–frequency space to make the spectral shifts at section boundaries more visible; 2) a mel-spectrogram generator (MLSG) to create mel-spectrograms consisting of bands aligned with human hearing, highlighting the frequencies people perceive more clearly; 3) max-pooling on time and frequency to shrink the feature size and emphasize broader structural patterns; 4) padding extra frames at the start and end for constant dimensions of features to prevent boundary clips caused by batching; 5) z-score normalization that centers and scales each mel band to stabilize and accelerate DNN training; and 6) SpecAugment [12] that randomly masks out bands and time spans to simulate noise for better generalization of DNN-based algorithms.

Ground-Truth Processing: We preprocess ground-truth boundary annotations given in the music structure dataset with our frame-based input features, we load each boundary timestamp given in seconds from the annotation text files and convert it into the corresponding mel-spectrogram frame index.

These two branches produce the input features needed by DNN training.

C. DNN Training

The DNN-based algorithms learn to map time–frequency features into continuous boundary signals, providing per-frame likelihoods of structural transitions. Our reference design includes three algorithms, based on: CNN, CRNN, and BiLSTM. Each of them is selected for its unique modeling strength. more specifically, CNNs focus on local spectral detail, BiLSTMs capture longer-range temporal context, and CRNNs fuse both spatial and temporal information.

D. Postprocessing

Our reference postprocessing stage consists of two steps to extract final boundary timestamps. First, we locate peaks in the curve points where the DNN's output is higher to identify the most likely boundary candidates. Second, we filter out weaker peaks by applying an empirical threshold, removing false-positive detections and ensuring only high-confidence peaks remain. The detected result is a time-ordered list of boundaries of the input audio.

E. Evaluation

In the reference evaluation stage, the quality of predicted boundaries are assessed in three ways. First, in *objective evaluation*, we calculate popular metrics, i.e., precision, recall, F1-score, and IG for quantitative benchmarks. Second, in *subjective evaluation*, we conduct a user study with several experts to understand how well the boundaries feel to experts. MOS values are computed to quantify the quality of predicted boundaries. Last, in *meta evaluation*, we correlate the objective scores with the MOS to identify which objective metric best aligns with human perception. By doing so, we aim to avoid costly and non-real-time user studies for subjective evaluation.

IV. IMPLEMENTATIONS

In this section, we give the detailed design and implementation details of a reference pipeline.

A. Preprocessing

We details our implementation of the two preprocessing braches below.

Feature Extraction: Let $x(t)$ be raw audio signals from a given music structure dataset, we first apply STFT with a window size of 4096 and a hop length of 1024, denoted as $S(f,t) = |\text{STFT}\{x(t)\}|$. This is followed by the MLSG, which essentially performs 80-band mel-filterbank conversion and logarithmic compression to obtain the mel-spectrogram, written as $M(f,t)$. Next, pink noise at -70 dB is added to avoid silent-padding artifacts, and frequency-wise z-score normalization is applied to get the normalized mel-spectrogram, denoted by:

$$Z(b_i,t) = \frac{P(b_i,t) - \mu_i}{\sigma_i}, \tag{1}$$

where $P(b_i,t)$ is the padded log-mel-spectrogram, and μ_i, σ_i are the mean and standard deviation for band b_i. To address data scarcity, we adopt SpecAugment [12] for frequency and time masking the normalized mel-spectrogram $Z(b_i,t)\ \forall i$. In particular, the frequency masking occludes 30 randomly selected bands, while time masking suppresses 32 consecutive frames. Both are applied once per sample to preserve stability.

Ground-Truth Processing: For each audio, we load boundary times annotations in seconds from the music structure dataset and convert them into frames by multiplying with a sample rate, dividing by a hop length, and rounding them to the nearest frames. We then remove any duplicate or adjacent frame indices so that each boundary is unique and add zero-label padding at the start and end to match the feature length and avoid edge clipping. Finally, we smooth each boundary impulse with a 0.1 s Gaussian window and normalize its peak to 1.0, capturing a small span of temporal uncertainty.

B. DNN Training

We design three reference DNN-based music boundary detection algorithms which are described below.

- **CNN:** Two 2D convolutional layers with kernel size of 5×31, ReLU, and batch normalization. We apply 5×3 max pooling after the first layer and 0.6 dropout before each layer. The final output is reshaped and passed through a 1D convolutional serving as a fully connected predictor.
- **BiLSTM:** Two bidirectional LSTM layers with a hidden size of 256, followed by dropout and layer normalization. The combined output is passed through a fully connected layer to produce scalar predictions for individual frames.
- **CRNN:** A CNN frontend followed by BiLSTM layers to merge local feature extraction and temporal modeling. The output passes through three fully connected layers before the final 1D convolutional predictor.

We emphasize that these algorithms are merely reference implementations. Researchers can test their own DNN-based music boundary detection algorithms using our pipeline.

C. Postprocessing

After the DNN-based algorithms produce frame-wise boundary likelihoods, we extract discrete boundary times in the following two steps.

- **Peak picking**: we use `scipy.signal.find_peaks` [13] to identify local maxima in the likelihood curve, requiring a minimum inter-peak distance of 6 seconds to avoid over-segmentation .
- **Thresholding**: we empirically select algorithm-specific decision threshold τ on the validation set to maximize F1-score to filter out low-confidence peaks. Only peaks whose confidence meets or exceeds τ are retained as final boundary predictions.

D. Evaluation

Objective Evaluation: We define a tolerate window to compute the precision, recall, and F1-score objective metrics for music boundaries detected by DNN-based algorithms. A detected boundary is counted as correct if it falls within the tolerance window of a ground-truth boundary. We also compute IG, to quantify how informative the timing errors are beyond chance [3].

Subjective Evaluation: To facilitate the user study, we have created a web-based graphical user interface (GUI) shown in Fig. 2. Using the GUI, a music expert can select a music file with detected boundaries detected by a DNN–based algorithm. The experts then interactively listens and inspects the detected section boundaries using the same GUI. To avoid bias, we randomly map each music file and corresponding detected boundaries into a "song" in the GUI. For each song, a music expert uses the GUI to:

- Validate each detected boundary section by marking its start/end as "Yes" or "No."
- Flag any boundaries whose start times are misaligned.
- Inserts any sections that were missed by the algorithm.
- Rate an overall accuracy score of detected sections from 1 (very inaccurate) to 10 (perfect).

We compute the MOS values of the overall accuracy score across all experts, which are used in the meta evaluation.

Meta Evaluation: To determine which objective metric best reflects human perception, we compute Pearson's correlation coefficient r between each DNN-based algorithm's objective metrics and the subjective MOS. The objective metric exhibiting the highest r is deemed the most perceptually aligned and used to approximate the subjective results from user studies.

V. EVALUATIONS

We give our evaluation setup and results in this section.

A. Setup

Dataset: We use the SALAMI music structure dataset [14], which contains over 1400 music files spanning diverse genres. Although SALAMI provides multi-level structural labels (fine, coarse, and functional), we leverage the boundary timestamps only. As summarized in Table I, we reserve 327 music files for testing and split the remainder into 744 for training and 100 for validation. In the preprocessing stage, we apply SpecAugment [12] on the training set, doubling it to 1488 music files.

TABLE I
THE SPLITS FOR THE SALAMI DATASET

Dataset	SALAMI
Training set	744
Training set after preprocessing	1 488
Validation set	100
Test set	327

TABLE II
THRESHOLDS AND EPOCHS FOR EACH DNN ALGORITHM

Algorithm	Threshold τ	Epochs
CNN	0.235	129
BiLSTM	0.235	129
CRNN	0.180	36

Parameters: We implement all DNN-based algorithms in `PyTorch 1.4.0+cu100` and train them with the Adam optimizer, using 1) an initial learning rate of 1×10^{-3}, 2) a decaying factor of 0.1 every 30 epochs, and 3) binary cross-entropy with logits as the loss function. To extract peaks from each DNN's output signal, we empirically select decision threshold τ using the validation set to maximize the F1-score. Table II lists the chosen thresholds and the numbers of epochs required for convergence of individual DNN-based algorithms.

Metrics: We evaluate the reference DNN-based algorithms using standard structural segmentation metrics: precision (P), recall (R), and F1 score, computed using the segment detection method of the `mir_eval` library [3]. Consistent with prior work [15], we report scores at two tolerance windows: 0.5 and 3.0 seconds. We also compute IG to assess how informative the detected boundary distribution is relative to the ground-truth distribution. Particularly, IG captures the degree to which detected boundaries structurally aligned with the annotated ones, offering a measure beyond point-wise overlap.

B. Objective Evaluation Results

We present the performance of different music boundary detection algorithms without and with SpecAugment in Table III. We make three key observations on this table. First, the CNN-based algorithm achieves the highest precision, recall, F1-score, and IG values. We believe the success of CNNs lies in their local focus: small convolutional kernels capture fine-grained spectral patterns, while pooling layers emphasize abrupt timbral shifts that often indicate section boundaries [5], [7]. Second, longer tolerance windows result in higher F1-score, precision, recall, and IG values, which is intuitive. The tolerance window shall be determined by the requirements of specific usage scenarios, e.g., for a real-time DJ or remixing tool, where producers must align samples to precise bar lines, a tight window around 0.5 second is essential to avoid audible drift [16]. In contrast, a consumer-facing music visualizer

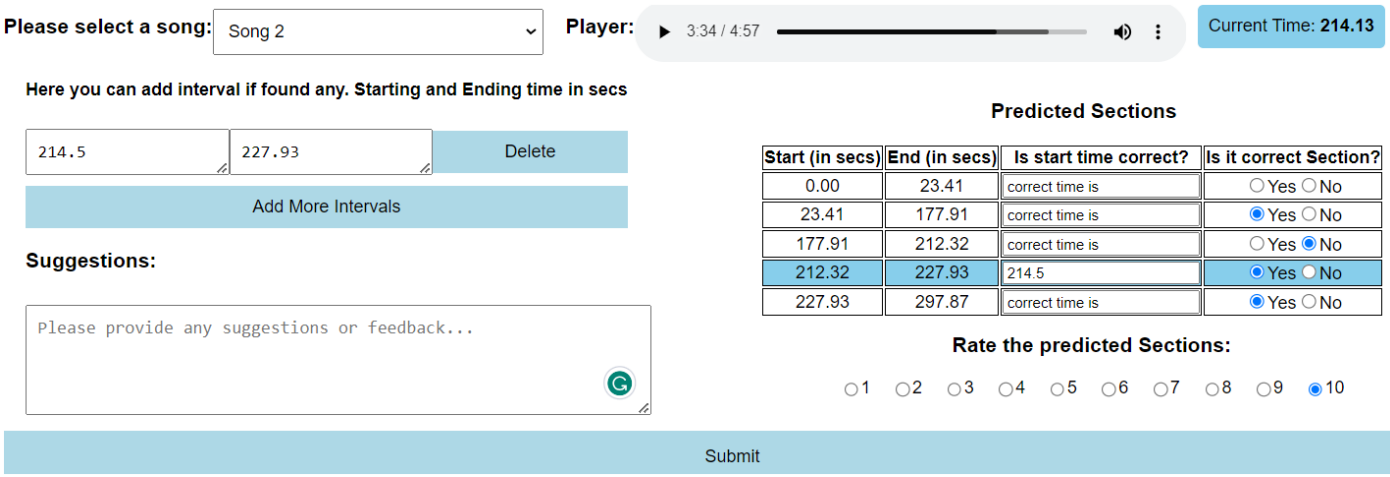

Fig. 2. Graphical user interface (GUI) for subjective evaluation. Music experts rate the accuracy of DNN-detected boundaries by validating sections and assigning perceived overall accuracy scores.

TABLE III
PERFORMANCE OF DIFFERENT DNN-BASED ALGORITHMS

	Without SpecAugment							With SpecAugment						
Algorithm	$P_{0.5}$	$R_{0.5}$	$F1_{0.5}$	$P_{3.0}$	$R_{3.0}$	$F1_{3.0}$	IG	$P_{0.5}$	$R_{0.5}$	$F1_{0.5}$	$P_{3.0}$	$R_{3.0}$	$F1_{3.0}$	IG
CNN	**0.42**	**0.36**	**0.38**	**0.74**	0.46	**0.52**	**0.41**	**0.50**	**0.38**	**0.42**	**0.70**	**0.54**	**0.58**	**0.45**
BiLSTM	0.38	0.16	0.21	0.59	0.26	0.34	0.38	0.35	0.28	0.30	0.59	0.43	0.47	0.43
CRNN	0.37	0.29	0.31	0.63	**0.51**	0.53	0.39	0.46	0.31	0.35	0.67	0.47	0.51	0.42

can tolerate a looser 3-second window without perceptible artifacts [4]. Third, we observe SpecAugment [12] helps to improve the boundary detection performance in general: as high as 0.13, 0.09, 0.17, and 0.05 boosts have been observed in F1-score, precision, recall, and IG. This confirms the merits of applying SpecAugment's time-frequency masking forces each algorithm to learn boundary cues from the surrounding context rather than memorizing spectral patterns. By randomly occluding bands and time spans during training, each algorithm must infer section transitions from more global structure and redundant cues, which reduces overfitting and improves generalization. As a result, all our considered DNN-based algorithms, especially the CNN-based one shows consistent gains in precision, recall, F1-score, and IG.

TABLE IV
ACCURACY AND MOS PER DNN-BASED ALGORITHM

Algorithm	Accuracy (%)	MOS (1–10)
CNN	89	8.87
BiLSTM	83	8.39
CRNN	87	8.92

C. Subjective Evaluation Results

Table IV reports 1) the accuracy of detected sections based on Yes/No validation given by the experts and 2) the MOS on a 1-10 scale for the overall rating of all sections. This table reveals that CNN and CRNN achieve comparable performance in accuracy and MOS, while BiLSTM perform the worst because its sequence-modeling nature smooths over rapid spectral changes, yielding flatter confidence curves that miss or blur true boundary points. The experts thus found and flagged more misaligned or missing boundaries, resulting in lower accuracy and overall ratings. We also observe that only 7% of the boundary detected sections were marked as misaligned segments and 5% of the boundary detected sections were marked as missing, which are way smaller than the inaccurate detection rate , i.e, 1-accuracy labeled by the experts.

D. Meta Evaluation Results

Next, we analyze which objective metric best aligns with expert judgments by correlating these metrics with subjective MOS over average overall accuracy ratings. Figure 3 gives the r values of individual DNN-based detection algorithms. This figure reveals that each DNN-based algorithm could require a different objective metric serving as a proxy for costly user studies. In particular, based on our user study, the best proxy metric for the CNN-, BiLSTM-, and CRNN-based algorithms are IG, F1-score, precision, and recall, respectively. Cross referring to the subjective evaluation results in Table IV, we recommend using our CNN-based detection algorithm with IG as the proxy metric, which leads to as high as 0.15 improvement in Pearson's r values, compared to the commonly used F1-score, precision, and recall.

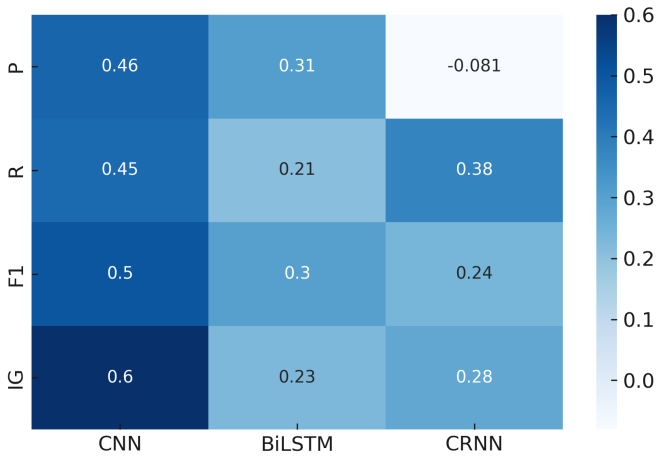

Fig. 3. Pearson's r coefficients between 50 responses for each model and evaluation metrics.

VI. Conclusion

This user study presented a compact four-stage pipeline for music boundary detection that balances efficiency and perceptual relevance. By converting raw audio into log-mel-spectrogram features, augmented with SpecAugment, and training lightweight CNN, CRNN, and BiLSTM algorithms, we achieve accurate boundary estimates at real-time speeds on a single CPU core. Our experts study confirms that Information Gain provides the strongest alignment with human judgments, outperforming precision, recall, and F1-score. This perceptually grounded metric can streamline future evaluations by reducing the reliance on labor-intensive listener tests. Together, these results demonstrate that simple algorithms, when properly augmented and validated, deliver reliable, listener-informed boundary detection suitable for mobile and embedded music applications.

References

[1] O. Nieto, G. J. Mysore, C.-i. Wang, *et al.*, "Audio-based music structure analysis: Current trends, open challenges, and applications," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[2] C. Hernandez-Olivan and J. R. Beltran, "Music composition with deep learning: A review," *Advances in speech and music technology: computational aspects and applications*, pp. 25–50, 2022.

[3] C. Raffel, B. McFee, E. J. Humphrey, *et al.*, "mir_eval: A transparent implementation of common music information retrieval metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 367–372.

[4] O. Nieto, M. M. Farbood, T. Jehan, and J. P. Bello, "Perceptual analysis of the f-measure for evaluating section boundaries in music," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 265–270.

[5] K. Ullrich, J. Schlüter, and T. Grill, "Boundary detection in music structure analysis using convolutional neural networks.," in *ISMIR*, 2014, pp. 417–422.

[6] A. Cohen-Hadria and G. Peeters, "Music structure boundaries estimation using multiple self-similarity matrices as input depth of convolutional neural networks," in *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*, Audio Engineering Society, 2017.

[7] C. Hernandez-Olivan, J. R. Beltran, and D. Diaz-Guerra, "Music boundary detection using convolutional neural networks: A comparative analysis of combined input features," *arXiv preprint arXiv:2008.07527*, 2020.

[8] T. Grill and J. Schlüter, "Music boundary detection using neural networks on combined features and two-level annotations.," in *ISMIR*, 2015, pp. 531–537.

[9] J.-C. Wang, Y.-N. Hung, and J. B. Smith, "To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 416–420.

[10] W.-T. Lu, M.-H. Wu, Y.-M. Chiu, and L. Su, "Actions speak louder than listening: Evaluating music style transfer based on editing experience," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3936–3944.

[11] T. Kim and J. Nam, "All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio," in *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, IEEE, 2023, pp. 1–5.

[12] D. S. Park, W. Chan, Y. Zhang, *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[13] S. Böck, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods.," in *ISMIR*, 2012, pp. 49–54.

[14] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations.," in *ISMIR*, Miami, FL, vol. 11, 2011, pp. 555–560.

[15] M. Müller, *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015, vol. 5.

[16] S. Balke, J. Reck, C. Weiß, J. Abeßer, and M. Müller, "Jsd: A dataset for structure analysis in jazz music," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, 2022.