

DEEPCHORUS: A HYBRID MODEL OF MULTI-SCALE CONVOLUTION AND SELF-ATTENTION FOR CHORUS DETECTION

Qiqi He¹, Xiaoheng Sun¹, Yi Yu² and Wei Li^{1,3}

¹ School of Computer Science and Technology, Fudan University, Shanghai, China

² National Institute of Informatics (NII), Tokyo, Japan

³ Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China

{heqq20, 19210240112, weili-fudan}@fudan.edu.cn, yiyu@nii.ac.jp

ABSTRACT

Chorus detection is a challenging problem in musical signal processing as the chorus often repeats more than once in popular songs, usually with rich instruments and complex rhythm forms. Most of the existing works focus on the receptiveness of chorus sections based on some explicit features such as loudness and occurrence frequency. These pre-assumptions for chorus limit the generalization capacity of these methods, causing misdetection on other repeated sections such as verse. To solve the problem, in this paper we propose an end-to-end chorus detection model *DeepChorus*, reducing the engineering effort and the need for prior knowledge. The proposed model includes two main structures: **i)** a Multi-Scale Network to derive preliminary representations of chorus segments, and **ii)** a Self-Attention Convolution Network to further process the features into probability curves representing chorus presence. To obtain the final results, we apply an adaptive threshold to binarize the original curve. The experimental results show that *DeepChorus* outperforms existing state-of-the-art methods in most cases.

Index Terms— Chorus detection, Self-Attention, Multi-Scale Network, Music structure analysis, Music information retrieval

1. INTRODUCTION

Music songs are highly structured, especially for popular music [1]. Among the sections in a pop song such as intro, verse, chorus, and bridge, the chorus can always best reflect the “most catchy” section of a piece of music [2]. Chorus detection is such a task that aims to find a short, continuous segment of a piece of music that can nicely represent the whole piece. In music information retrieval (MIR) field, the automatic detection of the chorus is a challenging problem, while it has extensive application value in various scenarios, such as extracting audition for commercial, and helping the user to quickly and efficiently preview selections from a large music database [3].

Many previous studies have assumed that the melody patterns with the most repetition correspond to the chorus sections. Based on the assumption, methods with manual features or traditional machine learning have been attempted for chorus detection: two approaches for finding “key phrases” based on clustering or hidden Markov model (HMM) was proposed in [4]; the other method considered the audio signal as several “state” parts, using k-means and HMM to derive different sections automatically [5]. Self-similarity matrix (SSM) is also widely used [6, 7], by which some significant change points can be derived and therefore the most-repeated sections are detected as the chorus. As a related research field, more research focused on generating “thumbnails” from music songs, which is also known as “thumbnailing”. A chroma-based thumbnailing

framework was proposed in [8]. A distance matrix representation was used [9] to produce music thumbnails. [1] proposed a fitness value that expresses the repetitiveness of the segments. In addition, many studies on music structure analysis involving the chorus sections also achieved certain results [10, 11].

With the development of deep learning in these years, the attempts of using deep neural networks (DNN) for chorus detection began to increase. [12] and [13] used music genre/emotion classification as surrogate tasks to establish unsupervised models for music highlight extraction. [14] proposed a hybrid generative model for music structure analysis, in which a recurrent neural network (RNN) is used for pre-processing. However, the lack of datasets with structure annotation limited the development of the supervised method for chorus detection. It is only in recent years that more audio datasets are annotated into chorus sections [15], alleviating the problem of insufficient training data. Using these datasets, [16] proposed a supervised multi-task approach to detecting chorus in popular music. Combining boundary detection, the method demonstrated a great improvement in F-measure.

Though many studies have tried the DNN-based approach for chorus detection, the existing methods either rely on some surrogate tasks [12, 13], or introduce some additional information [14, 16]. Most of these methods require complex post-processing. However, with the excessive involvement of hand-craft engineering, generalization ability may be hindered in such models.

In this paper we propose *DeepChorus*, an end-to-end chorus detection system based on a hybrid network of multi-scale convolution and self-attention, differentiating and extracting choruses by learning the latent features (such as harmonics from different instruments, or complex performance in low frequency caused by various rhythm patterns) of musical sections. Highlighting the similarity among different sections by learnable parameters, the required engineering effort and the need for prior knowledge can be reduced. To extract embedding vectors suitable for these characteristics from input Mel-spectrograms, a *Multi-Scale Network* is introduced to focus on the latent chorus information both in high and low resolution. For further processing to emphasize and summarize the similarity of chorus sections, self-attention layers combined with convolutional layers are used in the *Self-Attention Convolution Network* to compute the correlation between each vector. Moreover, the binarization process to obtain final chorus section results is implemented by adaptive thresholds and requires no prior knowledge, meaning that *DeepChorus* can work end-to-end from input-output pairs. Experimental results on three public datasets show that *DeepChorus* outperforms existing state-of-the-art methods in most cases.

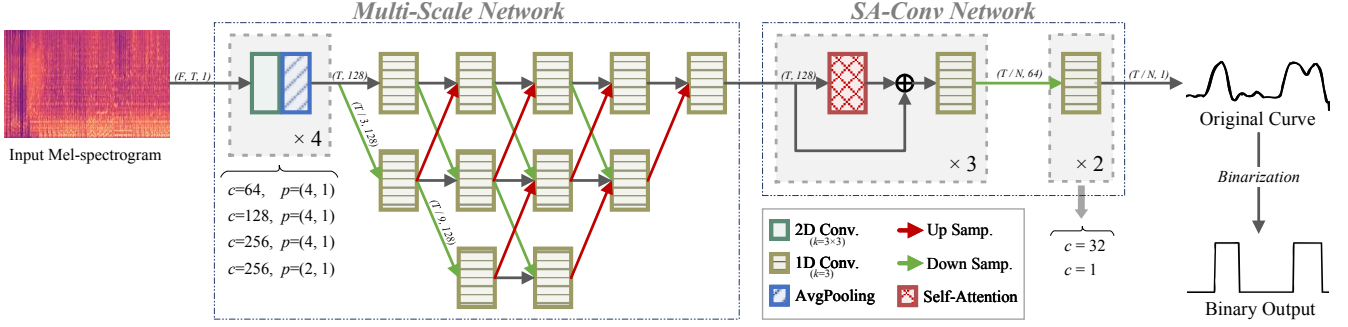


Fig. 1: The overall architecture of *DeepChorus*. (F, T, C) denotes a feature map with length F, T and C on frequency axis, time axis, and channel axis respectively. k, c and p denote kernel size, channel number and pooling size respectively. Downsampling / upsampling are implemented by 1D convolutional / transposed convolutional layers with strides of the same size as kernels.

2. PROPOSED METHOD

The overall framework of *DeepChorus* is illustrated in Fig. 1. As shown in the figure, the output of the model is a binary vector, representing chorus or non-chorus. Because the Mel-scaled frequency matches closely the human auditory perception, the model is designed to take Mel-spectrograms as inputs. Note that the input song can be of arbitrary length, leading to a flexible model. In the rest of this section, we detail the main modules of *DeepChorus*.

2.1. Multi-Scale Network

For chorus detection, global information is particularly important for detecting homogeneity and repetitions, while local information is essential for finding an accurate position of change points. Considering the efficiency and flexibility of convolutional layers in MIR field [13, 16], we adopt a CNN-based architecture as the backbone. However, kernels are specifically designed to capture short-range information, a single convolutional layer cannot model dependencies that extend beyond the receptive field, causing unsatisfied results, and stacking multiple convolutional layers lead to an excess of parameters. The visualization of feature maps and output curves (as shown in Fig. 2) also demonstrate convincingly that a traditional CNN structure is not such a suitable design for capturing global information for chorus detection. As a simple solution, downsampling operation is introduced in some network structures for segmentation, known as U-Net [17]. The dilated convolution [18] is also considered as an alternative to gain larger receptive fields. However, these solutions lead to the loss of details to some extent, which are important for locating chorus boundaries accurately.

In order to tackle these issues, we introduce a multi-scale resolution network to our framework, which was originally designed for computer vision tasks [19]. The key idea of the presented strategy is: the input feature should be first downsampled to low resolution to facilitate the extraction of global information, and then merged to the high resolution. By downsampling / upsampling the feature to different scales and exchanging information repeatedly, embedding vectors capable of distinguishing chorus from non-chorus can be derived and region information can be highlighted for further processing after just a few layers.

The architecture of the Multi-Scale Network is composed of several branches for different scales. To convert the spectral information into the channel domain, we first process the spectrogram input $(F \times T \times 1)$ into a sequence with the shape of $(T, 128)$ by

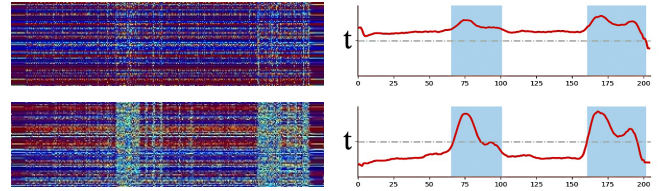


Fig. 2: Features after several layers (left) and the corresponding output curves (right) for “Tenshi no utatane” in RWC, derived by a single-scale (upper) and a multi-scale (bottom) CNN model respectively. “ t ” is the adaptive threshold and blue regions indicate the ground truth.

2D convolutional layers with average pooling. Then in each branch, the features are rescaled over the time axis using 1D convolutional layers and transposed convolutional layers [20]. The Multi-Scale Network conducts fusion repeatedly by concatenation along channel dimension such that representations in each branch receive information from different resolutions, leading to embedding vectors expressing chorus structure clearly.

Fig. 2 shows the visualization of the output features of a traditional CNN without multi-scale structure and the Multi-Scale Network in *DeepChorus*. The original curve before binarization is also shown in the figure. Note that the two models have a similar number of parameters. It is demonstrated that *DeepChorus* with multi-scale structure obtains a clearer representation of chorus sections, while the output feature of traditional CNN network lacks obvious boundaries. The output curve of *DeepChorus* also shows a more obvious change trend along the time axis, proving that the multi-scale structure is more suitable for chorus detection.

2.2. Self-Attention Convolution Network

Success has been made in traditional methods by using SSM to detect chorus sections based on the homogeneity and repetitiveness of acoustic features [1]. However, a drawback in such methods is that additional hand-craft rules are often needed to form segments [6, 7], which leads to unsatisfying of robustness. Instead, we argue that the self-attention layer [21] is a suitable alternative because of its strong ability to build pairwise relations using learnable parameters. Besides, embodying the principle of local processing, the convolutional layer can learn the local spatial features such as music struc-

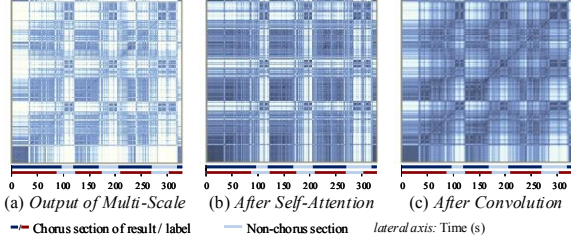


Fig. 3: SSMs of SA-Conv Network of “J-Song(Vagabond)” in SL dataset. Each point represents the distance between two vectors, and the darker the color, the closer the distance.

ture edges and instrumental details within a segment. Adopting convolutional layers to further process the sequence helps to highlight the chorus boundary clearer. Therefore, we present a Self-Attention Convolution (SA-Conv) Network stacking self-attention layers and convolutional layers alternatively for the further processing of embedding sequence produced by the Multi-Scale Network.

We design an SA-Conv Block as the basic module. In the block, a self-attention layer and a convolutional layer are used. Three SA-Conv Blocks are stacked successively to form the main structure. Then a 1D convolutional layer whose kernel length and stride are N is followed to downsample sequence length to $1/N$, where N denotes the number of frames per second (fps) in the input Mel-spectrograms (in our experiments, N is 43). Finally, two convolutional layers are adopted to process the sequence into a probability curve, representing the presence of chorus.

For an input sequence with n elements $\mathbf{x} = (x_1, \dots, x_n), x_i \in \mathbb{R}^{d_x}$, the self-attention layer produces a new sequence $\mathbf{s} = (s_1, \dots, s_n)$ of the same length where $s_i \in \mathbb{R}^{d_s}$, and \mathbf{s} is fused with \mathbf{x} by concatenation. Each element of \mathbf{s} is computed as a weighted sum of the linearly transformed input elements, and the results are fed to a softmax computation:

$$s_i = \text{softmax} \left[\sum_{k=1}^n (p_{ik})(x_k W^V) \right] \quad (1)$$

where p_{ik} is computed using a scaled dot-product to measure the similarities between x_i and x_k :

$$p_{ik} = \frac{(x_i W^Q)(x_k W^K)^T}{\sqrt{d_s}} \quad (2)$$

$W^V, W^Q, W^K \in \mathbb{R}^{d_x \times d_s}$ are trainable parameter matrices in each layer. $\sqrt{d_s} = \|\mathbf{s}\mathbf{s}^T - \mathbf{I}\|$ is a factor to prevent the result of dot-product being too large. Using scaled dot-product computation in Eq. 2 enables linear transformations of the inputs of sufficient expressive power.

To intuitively show the mechanism, we demonstrate the SSMs for the outputs of different layers in the SA-Conv Networks, as shown in Fig. 3. It can be observed in Fig. 3b that the self-attention layer summarizes the similarity to produce a more explicit representation, and therefore clusters the embeddings in the high-dimensional space for further processing. Furthermore, Fig. 3c exhibits that the convolutional layer works as a supplement to the self-attention layer to enhance the local boundaries.

2.3. Binarization

In experiments we observed that impulse noise makes the resulted curves serrated, hindering the threshold selection for binarization.

Therefore, we first apply a median filter to smooth the output curves. Supposing $\mathbf{y} = (y_1, \dots, y_n)$ is the original curve delivered by the SA-Conv Network, the smoothed curve $\mathbf{m} = (m_1, \dots, m_n)$ is given by:

$$m_i = \begin{cases} \text{Med}(y_{i-4}, y_{i+4}), & 4 < i \leq n-4, \\ y_i, & \text{otherwise.} \end{cases} \quad (3)$$

where $\text{Med}(y_{i-4}, y_{i+4})$ denotes the trimmed mean in the window.

The probability curves for different songs are erratic and a fixed threshold cannot produce satisfactory results. To solve the problem, we apply adaptive thresholds for different curves to perform binarization. The adaptive threshold t for \mathbf{m} is defined by:

$$t(\mathbf{m}) = 0.5 \times (m_{\max} - m_{\min}) \quad (4)$$

where m_{\max} and m_{\min} correspond to the maximum and minimum of \mathbf{m} respectively.

3. EXPERIMENT

3.1. Experiment Setup

3.1.1. Datasets & Metrics

For training and validation, we select 886 tracks with “chorus” label from Harmonix [15] and 102 tracks by The Beatles and Michael Jackson from Isophonics [22] and regard “refrain” labels as “chorus”. Finally, a total of 988 tracks are used, among which 20 tracks are randomly chosen for validation, and the rest 968 tracks are adopted for training.

For testing, we used following public datasets: a) RWC: 100 popular songs from [23] with structural labels provided by [24]; b) SP: 210 songs tagged as “Popular” from SALAMI [25]; and c) SL: 198 songs tagged as “Live” class from SALAMI. Tracks are carefully selected to ensure that there is no overlap between SP and SL.

To compare with the latest work [16], F1 score and the area under curve (AUC) are chosen as evaluation metrics in this paper. We use AUC [26] to study the model’s ability to produce sensitive probability curves, while the F1 metric is a standard measurement for music structure analysis [27]. For our ablation study, we supplement recall rate and precision rate as additional metrics, which are widely used to evaluate the effectiveness of information retrieval systems [27].

3.1.2. Training Details

To extract the input Mel-spectrograms, we first resample the original audio data to 22,050 Hz, then we use 512 overlapping windows of 2,048 frames and transform each window into a 128-band Mel-scaled magnitude spectrum. For training, the batch size we set is 4. In each epoch, 3,096 consecutive frames of each track (roughly 72 seconds long) are randomly selected for training. We choose the mean squared error as the loss function, and an Adam optimizer [28] is applied with a learning rate of 0.0001. We evaluate F1 on the validation set every 50 iterations and save the model with the highest F1 score. The training is not stopped until F1 score has not improved for 500 iterations.

3.2. Ablation Study

To demonstrate how much the proposed structures contribute to the model, we conduct a set of experiments. Note that all the models in this section experiment have similar parameters (about 5M).

Method	F1		AUC		Recall		Precision	
	RWC	SP	RWC	SP	RWC	SP	RWC	SP
<i>DeepChorus</i>	.675	.611	.842	.780	.582	.844	.873	.523
<i>w/o SA-Conv</i>	.469	.430	.743	.709	.689	.694	.408	.356
<i>w/o Multi-scale</i>	.593	.555	.833	.763	.760	.766	.576	.518
<i>Fully CNN</i>	.546	.541	.693	.689	.739	.799	.507	.472

Table 1: Results of Ablation Study on RWC and SP dataset, “w/o SA-Conv.”, and “w/o Multi-scale” denote for without SA-Conv and Multi-Scale Network respectively. “*Fully CNN*” stands for a traditional single-scale fully convolution network.

Method	RWC		SP		SL	
	AUC	F1	AUC	F1	AUC	F1
<i>DeepChorus</i>	.842	.675	.780	.611	.765	.501
<i>Multi2021</i> [16]	.819	.643	.675	.473	.633	.380
<i>Highlighter</i> [13]	.804	.407	.703	.303	.671	.251
<i>SCluster</i> * [11]	.533	.427	.545	.448	.551	.392
<i>CNMF</i> * [10]	.526	.403	.543	.422	.478	.340

Table 2: Results of the proposed and baseline methods on RWC, SP, and SL dataset. Traditional methods are marked with asterisks (*).

First, we replace the SA-Conv Network with a set of convolutional layers to justify the effectiveness of the SA-Conv Network. As shown in Table 1, the performances on RWC and SP datasets are both dropped to a large extent. When focusing on F1 metric, the model without the SA-Conv architecture shows significantly worse performance, decreased by 2.6% on RWC and by 18.1% on SP. Then, we remove the Multi-Scale Network and only keep the SA-Conv Network, using only “single-scale” convolutional layers for producing embedding vectors. The performance on the two datasets is decreased by 8.2% and by 5.6% respectively. For remaining metrics, *DeepChorus* also performs significantly better than other models in general. It is noteworthy that as shown in Fig. 2, the original probability curves of the single-scale model (*w/o Multi-scale*) are much flatter and mostly above the adaptive thresholds, which leads to a high recall rate but poor performance on AUC and precision.

To investigate the overall effectiveness of *DeepChorus* structure, we remove both the SA-Conv Network and the Multi-Scale Network, and a traditional fully convolution network is adopted. The results on F1 are decreased by 11.3% and 7.7% on the two datasets respectively. For AUC, recall rate, and precision rate, there is also a certain degree of decline. The results indicate the obvious effect of the SA-Conv and Multi-Scale Network in *DeepChorus*, proving them effective structures to detect the chorus more accurately.

3.3. Comparison with Previous Work

To compare with previous work, we choose four models as baseline methods: *CNMF* [10], *SCluster* [11], *Highlighter* [13], and *Multi2021* [16]. Among them, *CNMF* and *SCluster* are traditional methods, and the implementation by MSAF [29] (a framework contains various algorithms for music structure analysis) is used. For *Highlighter*, an unsupervised CNN-based model, the pre-trained model provided by the author is used for evaluation. *Multi2021* is a supervised CNN-based model with state-of-the-art performance among the chosen systems. For a fair comparison, we train the model on our training datasets. The model consists of nearly 7M

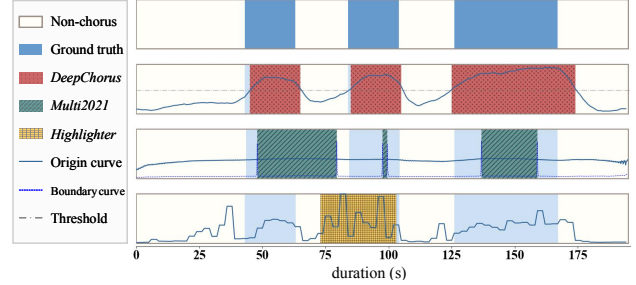


Fig. 4: Chorus detection results for “The Spy” in SP dataset using different methods.

parameters. Due to the limitations of experimental conditions, the batch size 256 in the original experiment is not achievable, and therefore we set batch to 32 as an alternative (roughly 600 seconds per mini-batch, about twice as long as ours).

The experimental results on the three test datasets are shown in Table 2. The results clearly confirm that *DeepChorus* outperforms four baseline methods significantly for both AUC and F1. When focusing on the RWC dataset, which is widely used in music structure analysis, *DeepChorus* exhibits an improved performance by 2.5% higher than the other methods on F1 on average. For *Multi2021*, which we consider as the state-of-the-art in chorus detection, *DeepChorus* outperforms it by 8.67% on AUC and by 9.7% on F1 on average. This gap may be largely due to the heavy dependency on prior information. For example, an average number of chorus is used as prior knowledge in *Multi2021*, which can be considered as a reason for the large performance gap between *DeepChorus* and *Multi2021* on SP. Because in SP, the repeat times of chorus varies notably among. The SL dataset is made up of live performance tracks, which leads to the degraded quality of audio signals, making chorus detection more difficult compared to the other datasets but also closer to the real-world scenarios.

To investigate how *DeepChorus* improves the performance in chorus detection, a case study is performed on a randomly selected song. We choose *Multi2021* and *Highlighter* to compare with because both of them are DNN-based methods. As depicted in Fig. 4, we can observe that there are fewer misdetections on chorus sections and more accurate boundaries for *DeepChorus*. From the visualization result of *Multi2021*, we can find that its original curves are too flat that nearly all values are below the fixed threshold 0.5, and therefore a boundary curve is indispensable to provide additional information for chorus section. Compared with *Highlighter*, we can say that the performance gains of *DeepChorus* can be attributed to the ability to detect multiple chorus sections at a time and to locate them with better precision.

4. CONCLUSION

In this paper, we have proposed *DeepChorus*, an end-to-end system for chorus detection, mainly consisting of a Multi-Scale Network and a Self-Attention Convolution Network. Trained on mass data, latent chorus information in Mel-spectrograms can be learned to differentiate and highlight choruses, and therefore the model gains higher generalization capacity. Experimental results show *DeepChorus* outperforms the existing state-of-the-art methods on three datasets significantly. Designing a method to improve the performance of chorus detection remains an avenue for future work.

The code of *DeepChorus* and the reproduced model is available at <https://github.com/Qqi-HE/DeepChorus>.

5. ACKNOWLEDGEMENT

This work was supported by National Key R&D Program of China (2019YFC1711800) and NSFC (62171138).

6. REFERENCES

- [1] Meinard Müller, Nanzhu Jiang, and Peter Grosche, “A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing,” *IEEE Transactions on audio, speech, and language processing*, pp. 531–543, 2012.
- [2] JMH van Balen, John A Burgoyne, Frans Wiering, and Remco C Veltkamp, “An analysis of chorus features in popular song,” in *International Society for Music Information Retrieval*, 2013, pp. 107–112.
- [3] Chia-Hung Yeh and Hung-Hsuan Lin, “The extraction of popular music chorus via structural content analysis,” in *IEEE Industrial Electronics Society*, 2007, pp. 2532–2536.
- [4] Beth Logan and Stephen Chu, “Music summarization using key phrases,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000, pp. 749–752.
- [5] Geoffroy Peeters, Amaury La Burthe, and Xavier Rodet, “Toward automatic music audio summary generation from signal analysis,” in *International Society for Music Information Retrieval*, 2002, pp. 1–1.
- [6] Sheng Gao and Haizhou Li, “Popular song summarization using chorus section detection from audio signal,” in *International Workshop on Multimedia Signal Processing*, 2015, pp. 1–6.
- [7] Matthew Cooper and Jonathan Foote, “Automatic music summarization via similarity analysis,” in *International Society for Music Information Retrieval*, 2002, pp. 81–85.
- [8] Mark A Bartsch and Gregory H Wakefield, “To catch a chorus: Using chroma-based representations for audio thumbnailing,” in *The IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, 2001, pp. 15–18.
- [9] Antti Eronen, “Chorus detection with combined use of mfcc and chroma features and image processing filters,” in *International Conference on Digital Audio Effects*, 2007, pp. 229–236.
- [10] Oriol Nieto and Tristan Jehan, “Convex non-negative matrix factorization for automatic music structure identification,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 236–240.
- [11] Brian McFee and Dan Ellis, “Analyzing song structure with spectral clustering,” in *International Society for Music Information Retrieval*, 2014, pp. 405–410.
- [12] Jung-Woo Ha, Adrian Kim, Chanju Kim, Jangyeon Park, and Sunghun Kim, “Automatic music highlight extraction using convolutional recurrent attention networks,” *arXiv:1712.05901*, 2017.
- [13] Yu-Siang Huang, Szu-Yu Chou, and Yi-Hsuan Yang, “Pop music highlighter: Marking the emotion keypoints,” *Transactions of the International Society for Music Information Retrieval*, pp. 68–78, 2018.
- [14] Go Shibata, Ryo Nishikimi, and Kazuyoshi Yoshii, “Music structure analysis based on an lstm-hsmm hybrid model,” in *International Society for Music Information Retrieval*, 2020, pp. 15–22.
- [15] Oriol Nieto, Matthew McCallum, Matthew EP Davies, Andrew Robertson, Adam M Stark, and Eran Egozy, “The harmonix set: Beats, downbeats, and functional segment annotations of western popular music,” in *International Society for Music Information Retrieval*, 2019, pp. 565–572.
- [16] Ju-Chiang Wang, Jordan BL Smith, Jitong Chen, Xuchen Song, and Yuxuan Wang, “Supervised chorus detection for popular music using convolutional neural network and multi-task learning,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2021, pp. 566–570.
- [17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and Computer-assisted intervention*, 2015, pp. 234–241.
- [18] Fisher Yu and Vladlen Koltun, “Multi-scale context aggregation by dilated convolutions,” in *International Conference on Learning Representations*, 2016, pp. 1–10.
- [19] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Minghui Tan, Xinggang Wang, et al., “Deep high-resolution representation learning for visual recognition,” *IEEE transactions on pattern analysis and machine intelligence*, pp. 3349–3364, 2020.
- [20] Vincent Dumoulin and Francesco Visin, “A guide to convolution arithmetic for deep learning,” *arXiv:1603.07285*, 2016.
- [21] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong, “Attention is all you need in speech separation,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2021, pp. 21–25.
- [22] Matthias Mauch, Chris Cannam, Matthew Davies, Simon Dixon, Christopher Harte, Sefki Kolozali, Dan Tidhar, and Mark Sandler, “Omr2 metadata project,” in *International Society on Music Information Retrieval*, 2009.
- [23] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, “Rwc music database: Popular, classical and jazz music databases,” in *International Society for Music Information Retrieval*, 2002, pp. 287–288.
- [24] Masataka Goto et al., “Aist annotation for the rwc music database,” in *International Society for Music Information Retrieval*, 2006, pp. 359–360.
- [25] Jordan BL Smith, John A Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie, “Design and creation of a large-scale database of structural annotations,” in *International Society for Music Information Retrieval*, 2011, pp. 555–560.
- [26] Foster Provost and Tom Fawcett, “Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions,” in *International Conference on Knowledge Discovery and Data Mining*, 1997, pp. 57–63.
- [27] Oriol Nieto, Gautham J Mysore, Cheng-i Wang, Jordan BL Smith, Jan Schlüter, Thomas Grill, and Brian McFee, “Audio-based music structure analysis: Current trends, open challenges, and applications,” *Transactions of the International Society for Music Information Retrieval*, pp. 246–263, 2020.
- [28] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980*, 2014.
- [29] Oriol Nieto and Juan P Bello, “Systematic exploration of computational music structure research,” in *International Society for Music Information Retrieval*, 2016, pp. 547–553.