U-BEAT: A MULTI-SCALE BEAT TRACKING MODEL BASED ON WAVE-U-NET

Tian Cheng and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST), Japan

ABSTRACT

In this paper, we propose a multi-scale model for beat tracking based on the Wave-U-Net model. The proposed model learns multi-scale features by repeatedly resampling feature maps via a series of downsampling blocks and upsampling blocks. With the U-shape structure, we observe that global features are summarized at the bottom blocks. Then, these global features guide feature upsampling for predicting beats with a steady tempo. The local features learned in the downsampling blocks are combined with the upsampled features for predicting beats precisely. Besides the features learned from the waveform, we also combine spectral features at a middle level in the model. Experimental results show that beat tracking performance is improved by combining spectral features.

***Index Terms*—** Multi-scale structure, beat tracking, combining waveform and spectral inputs

1. INTRODUCTION

Beat tracking is to determine a periodic sequence of time instants with which people tap along a music piece. Although the task has a long history [1], beat tracking still gains a great attention in the Music Information Retrieval (MIR) research field because it produces basic time units for musical content analysis. Beat tracking has been applied as an intermediate processing step for time regulation in other music analysis tasks, such as music transcription [2, 3], chord estimation [4–6], structure analysis [7, 8], and so on. It is also important for music-synchronized applications [1, 9, 10].

Beat tracking relies on local features for precise timing prediction and features at larger timescale for constraining global consistency. Current beat tracking models employ deep learning techniques to learn beats from labeled annotations, and feed the deep networks with spectral features. In these models, Convolutional Neural Networks (CNNs) are used to detect the local spectral events [11–14], and beats are predicted in a larger timescale by using sequence models, such as Recurrent Neural Network (RNN) [15–18] and Temporal Convolutional Network (TCN) [19–21].

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917, Japan.

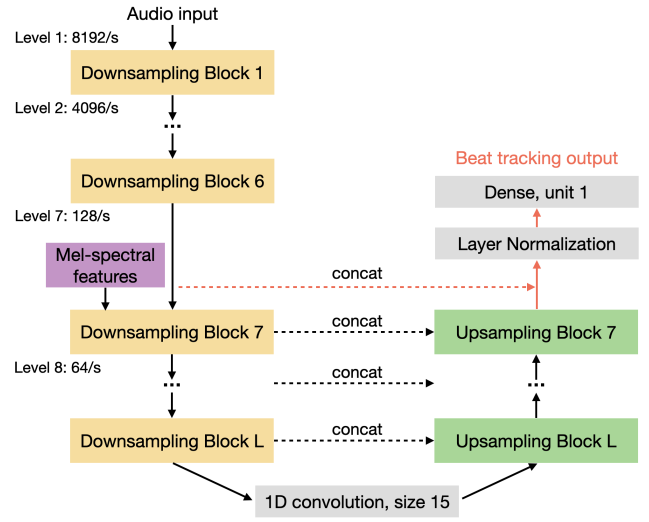
To learn and use features at different time scales for more flexible beat tracking, we propose U-Beat, a multi-scale model for beat tracking based on the Wave-U-Net model [22]. The model consists of a series of downsampling and upsampling blocks and provides a multi-scale structure with the time resolution halved at each successive level. We can make use of features at top levels for beat tracking with a desired time resolution. Besides the multi-scale structure, the U-shape structure is also important since it enables features to be upsampled from the bottom global features and concatenated with the local features from the downsampling block.

For the purpose of beat tracking, we modify the original Wave-U-Net model in several aspects. First, we replace the downsampling layers in the original model with maxpooling layers in order to avoid missing important beat-related information during downsampling. The second difference from the original model is the flexible output layers of the proposed model. In the original model, the model output is at the top level with a sample-rate resolution for audio source separation. In comparison, the proposed model has more flexible output layers, which makes full use of the multi-scale structure. We can produce outputs for beat tracking at multiple levels with the beat annotations sampled to the corresponding time resolutions, which is similar to the lyrics alignment model with the output from an intermediate upsampling block [23]. Third, we combine Mel-spectral features in the proposed model as a complement of the learned features from the waveform, which improves beat tracking performance.

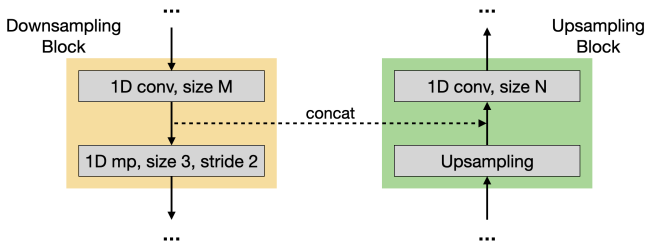
In the experiment, we find that the proposed model works on both waveform and spectral inputs, and we gain better results by combining both inputs. Although there are other multi-scale models, such as the TCN model which can also utilize the features from different scales by using the dilated connections, it is hard to combine input features at different scales in the models. In fact, TCN-based beat tracking models perform on either the spectral input [19–21] or the wave input [24]. To the best of our knowledge, the proposed model is the first model that leverages the Wave-U-Net model for beat tracking and combines both the waveform input and spectral features for this task.

2. U-BEAT MODEL

We build a multi-scale model for beat tracking based on the Wave-U-Net model. The overall framework of the proposed



(a) The overall framework.



(b) Details of the downsampling and upsampling blocks.

Fig. 1: The framework of the proposed model.

model is shown in Figure 1(a). We see that the waveform first goes through a series of downsampling blocks, and then connects to the upsampling part with a convolutional layer at the bottom level. We made some modifications to the original Wave-U-Net in order to apply the model for beat tracking, with details described as follows.

2.1. Downsampling with Maxpooling Layers

In each downsampling block, the input features are first convolved with a 1D filter of a size of M and then downsampled by a factor of 2. We load audios with a sampling rate of 8192 Hz. At Level l, there are $8192/2^{(l-1)}$ frames per second, as shown in Figure 1(a).

In our preliminary experiment, we found that direct downsampling in the downsampling block loses important information. For example, if peaks at beat times are discarded during downsampling, these peaks are difficult to recover during upsampling. In order to fix this problem and preserve the peaks, we use a maxpooling layer with a stride of 2 and a size of 3 instead of the downsampling layer used in the original Wave-U-Net model. Details of the downsampling block are shown in Figure 1(b).

2.2. Upsampling

The upsampling blocks start from the bottom to the top. As shown in Figure 1(b), in the unsampling block, the input features are first upsampled by a factor of 2 and concatenated with the features from the downsampling block at the same level. Then, the concatenated features are convolved with a 1D filter of a size of N. In the convolutional layers (for both the downsampling and upsampling blocks), we choose the same padding to preserve the dimension so that the model does not lose the beats at the beginning and end of each musical piece.

2.3. Beat Tracking Output Layer

For beat tracking, we concatenate features from the downsampling block and the upsampling block at the same level, and stack a layer normalization layer and a dense layer on top of the concatenated features, as shown in Figure 1(a). The dense layer has an output dimension of 1 and uses the sigmoid activation function. Since the time resolution of each level is different, we can choose the output layers flexibly and train beat tracking at different time scales. Theoretically, we can train beat tracking for every level, and obtain the sample-rate time resolution at the top level.[1] However, the time resolution of most existing beat tracking datasets cannot support the output for the topmost levels (i.e., the time resolution of existing beat annotations is too low to train those top levels). We show an example of the outputs from the beat tracking output layers for Level 7 to 10 of the same piece in Figure 2.

For a comparison to other state-of-the-art methods (usually with a time resolution of 10 ms), we train the model with the beat tracking output layer at Level 7. In the beat tracking output, there are 128 frames per second, which correspond to a time resolution of 7.812 ms. To track beats from this beat tracking output (output activation), we adapt the madmom's post-processing method [16] that is based on a Dynamic Bayesian Network (DBN). Readers are referred to [16, 25] for more details.

2.4. Combining Mel-spectral Features

With the multi-scale structure of the proposed model, we can combine spectral features at different levels as a complement to the waveform input. In this paper, we concatenate Mel-spectral features at Level 7 for beat tracking. The Mel-spectrogram is computed with a FFT size of 512 in a frequency range from 0 Hz to 4000 Hz. We choose a hop size of 64 to match the dimension of the downsampling features. Inspired by [26] which applies attention models on different

[1] According to this sample-rate time resolution at the top level, we use the sample rate of 8192 Hz, which is much higher than typical time resolutions for beat annotations. Preparing beat annotations for such high resolution in the future will further exploit the potential of the proposed U-Beat.

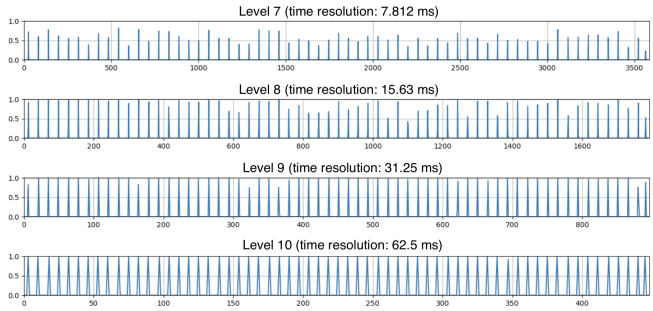


Fig. 2: Example of multi-resolution beat-tracking outputs given a musical piece from the GTZAN dataset.

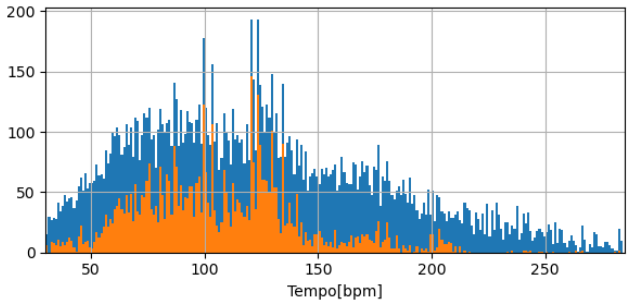


Fig. 3: Tempo distribution before (orange) and after (blue) data augmentation.

frequency ranges for tempo estimation, we also test the Mel-spectral features obtained by concatenating low-frequency and high-frequency Mel-spectrograms, which are computed in frequency ranges of 0-500 Hz and 500-4000 Hz, respectively.

In the experiment, we train and evaluate five models: one model with the waveform input, two models combining waveform and Mel-spectral inputs, and two models with only Mel-spectral inputs for comparison.

2.5. Network Parameters

We choose the same filter sizes for the convolutional layers as those used in the original Wave-U-Net model, with a size of $M = 15$ in the downsampling block and a size of $N = 5$ in the upsampling block. In Block l, there are $24 + 12 * l$ filters [22]. Because we observe some redundancy in the learned features, we set a maximum of 60 filters for all blocks. In our preliminary experiment, we obtained no improvement by increasing the filter sizes and the number of filters. On the basis of validation results, we decided to use a model with 11 levels ($L = 11$). We also tested adding spectral features at Level 6 and 8, respectively, which yielded similar results without significant differences.

3. TRAINING

The beat tracking model is trained and tested by using ten standard datasets. The Beatles dataset [27] and five RWC datasets [28, 29] are used for training. The Ballroom (685 items) [30, 31], Hainsworth (222 items) [32], and SMC (217 items) [33] datasets are trained and tested in an 8-fold cross-validation manner. The GTZAN (1000 items) [34, 35] is used for testing only.

We augment the training set by speeding up/slowing down clips with factors from 0.7-1.4 without altering the pitch [26]. The clips and the corresponding factors for speeding up/slowing down are carefully selected in order to produce balanced data for all tempos. We make sure that each piece is not selected more than 12 times for data augmentation. The validation and testing sets are left untouched. All data used for training are segmented into 30-second clips. After data augmentation, the number of clips in the training set increases from 4974 to 15202. The tempo distributions before and after augmentation are shown in Figure 3.

The model is trained with the binary cross-entropy loss on the beat tracking output. We apply the RMSprop optimizer with a learning rate of 10^{-4} and a batch size of 32. The training is stopped when there is no improvement in the validation loss for 20 epochs.

4. EVALUATION AND RESULT ANALYSIS

We use standard metrics for beat tracking evaluation: the *F-measure* and the continuity-based metrics *CMLt* and *AMLt* [27]. All metrics are computed with a ± 70 ms tolerance.

4.1. Comparing models with different inputs

We compare results obtained by models trained with five different inputs, as mentioned in Section 2.4. The models in this section are trained with the training data without data augmentation. From the results in Figure 4, we find that the spectral models (`mel` and `mel_l_h`) work better than the wave model (`wave`) for all four datasets. Dividing Mel-spectrogram into low- and high-frequency ranges (`mel_l_h`) works better than the single Mel-spectrogram input (`mel`), except for the SMC dataset. In general, combining wave and spectral inputs provide better results. The model `wave+mel_l_h` outperforms the model `wave` by 1.1-5.8 percentage points. It also outperforms the spectral models for all cross-validation results, and provides similar results as the spectral models for the testing-only data (the GTZAN dataset).

4.2. A comparison to state-of-the-art

With all above results considered, we choose the model (`wave+mel_l_h`) and train it again with the augmented

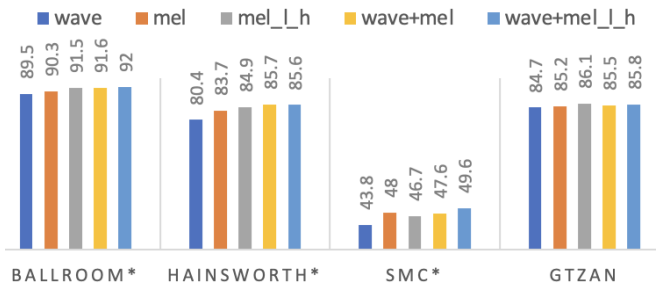


Fig. 4: F-measures of models with different inputs. $*$ denotes datasets used for 8-fold cross-validation.

training data (denoted by `wave+mel_l_h+aug`) to compare with state-of-the-art results. We show beat tracking results on testing data in Table 1. We can see that our results are further improved by data augmentation.

Although the goal of this paper is not to show the superiority of the U-Beat over the state-of-the-art results, we briefly explain some differences and show that the U-Beat can produce comparable results. For the GTZAN dataset (the testing-only data), the best *F-measure* was 88.3%, achieved by `Spec_TCN`, a TCN model trained on the spectrogram [21]. The proposed method (`wave+mel_l_h+aug`) provides the second best results with a 87.1% F-measure. For cross-validation results, our results are still worse than other spectral based models. Although the proposed model also uses the spectral features, the convolution is operated in 1D rather than 2D. Using 2D convolution layers can be a direction for further improvement. Our results are generally better than the other wave based model (`Wave_TCN`, which also used 1D convolution), except on the Hainsworth dataset. Note that the results of the `Wave_TCN` model are produced with a train/vali/test (80%/10%/10%) split without using cross-validation, so its results are not directly comparable since they depend more on the random split in comparison to other cross-validation results.

5. DISCUSSION ON POTENTIAL EXTENSIONS

As the mentioned in Section 4, our model uses only 1D convolutional layers. It is surprising to see that our model also works for the spectral input without the waveform input because the spectral input usually needs 2D convolutional layers to learn its latent features. We could further extend the model by using those 2D layers. The results also demonstrate that the model can be benefited by combining spectral features at level 7. We expect the model can be further explored by adding more features at different levels simultaneously based on the flexible multi-scale structure. The third way of exploring the ability of the model is to use a multi-task learning to train related tasks together, such as downbeat tracking [13, 14, 21, 36], tempo estimation [37], and so on.

	F-measure	*CMLt*	*AMLt*
	Ballroom		
`Spec_RNN_2` [16] $*$	93.8	89.2	95.3
`Spec_TCN` [21] $*$	**95.6**	**93.5**	**95.8**
`Wave_TCN` [24] †	92.5	82.9	93.7
`wave+mel_l_h+aug` $*$	93.1	87.8	94.6
`wave+mel_l_h` $*$	92	85.3	94.5
	Hainsworth		
`Spec_RNN_1` [17] $*$	88.4	80.8	91.6
`Spec_TCN` [21] $*$	90.4	85.1	93.7
`Wave_TCN` [24] †	**97.3**	**97.6**	**97.6**
`wave+mel_l_h+aug` $*$	86	78.7	89.3
`wave+mel_l_h` $*$	85.6	78.1	89
	SMC		
`Spec_RNN_1` [17] $*$	52.9	42.8	56.7
`Spec_TCN` [21] $*$	**55.2**	**46.5**	**64.3**
`Wave_TCN` [24]	41.8	28.0	41.9
`wave+mel_l_h+aug` $*$	51.4	42.4	56.6
`wave+mel_l_h` $*$	49.6	40.3	54.7
	GTZAN		
`Spec_RNN_1` [17]	86.4	76.8	92.7
`Spec_TCN` [21]	**88.3**	**80.8**	**93.0**
`Wave_TCN` [24]	82.8	71.9	86.0
`wave+mel_l_h+aug`	87.1	78	91.8
`wave+mel_l_h`	85.8	75.7	91.1

Table 1: Beat tracking results. $*$ and † denote results with 8-fold cross-validation and one single train/vali/test split, respectively; and the rest are results on testing-only data.

6. CONCLUSIONS

In this paper, we proposed the U-Beat model that uses the multi-scale structure of the Wave-U-Net model for beat tracking. Our contributions include 1) replacing the downsampling layers with the maxpooling layers to reserve the temporal features for beat tracking; 2) providing a model that works on both waveform and spectral inputs and obtaining better results by combining both inputs; and 3) demonstrating a beat tracking model with multi-resolution outputs (Figure 2). The advantage that distinguishes the proposed model from other beat tracking models is its flexibility of adding input features and output layers at different levels. We have already shown that the performance was improved by combining waveform and spectral inputs. We are also interested in multi-resolution beat tracking outputs, especially in results at higher resolutions from top levels. Our future work includes producing datasets with more precise beat annotations and predicting beats at a higher time resolution that is needed by some real-world music synchronization applications.

7. REFERENCES

[1] M. Goto and Y. Muraoka, "A Beat Tracking System for Acoustic Signals of Music," in *Proc. ACM Multimedia*, 1994, pp. 365–372.

[2] K. Ochiai, H. Kameoka, and S. Sagayama, "Explicit Beat Structure Modeling for Non-Negative Matrix Factorization-based Multipitch Analysis," in *Proc. IEEE ICASSP*, 2012, pp. 133–136.

[3] E. Nakamura, E. Benetos, K. Yoshii, and S. Dixon, "Towards Complete Polyphonic Music Transcription: Integrating Multi-Pitch Detection and Rhythm Quantization," in *Proc. IEEE ICASSP*, 2018, pp. 101–105.

[4] M. Mauch and S. Dixon, "Simultaneous Estimation of Chords and Musical Context from Audio," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 6, pp. 1280–1289, 2010.

[5] M. McVicar, R. Santos-Rodriguez, Y. Ni, and T. D. Bie, "Automatic Chord Estimation from Audio: A Review of the State of the Art," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 2, pp. 556–575, 2014.

[6] J. Pauwels, K. O'Hanlon, E. Gómez, and M. B. Sandler, "20 Years of Automatic Chord Recognition from Audio," in *Proc. ISMIR*, 2019.

[7] M. Levy and M. B. Sandler, "Structural Segmentation of Musical Audio by Constrained Clustering," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 2, pp. 318–326, 2008.

[8] O. Nieto *et al.*, "Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 3, no. 1, pp. 246–263, 2020.

[9] J. L. Oliveira, M. E. P. Davies, F. Gouyon, and L. P. Reis, "Beat Tracking for Multiple Applications: A Multi-Agent System Architecture With State Recovery," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 10, pp. 2696–2706, 2012.

[10] J. Kato, M. Ogata, T. Inoue, and M. Goto, "Songle Sync: A Large-Scale Web-based Platform for Controlling Various Devices in Synchronization with Music," in *Proc. ACM Multimedia*, 2018, pp. 1697–1705.

[11] S. Durand, J. P. Bello, B. David, and G. Richard, "Robust Downbeat Tracking Using an Ensemble of Convolutional Networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 1, pp. 72–85, 2017.

[12] A. Gkiokas and V. Katsouros, "Convolutional Neural Networks for Real-Time Beat Tracking: A Dancing Robot Application," in *Proc. ISMIR*, 2017, pp. 286–293.

[13] M. Fuentes *et al.*, "Analysis of Common Design Choices in Deep Learning Systems for Downbeat Tracking," in *Proc. ISMIR*, 2018.

[14] T. Cheng, S. Fukayama, and M. Goto, "Joint Beat and Downbeat Tracking Based on CRNN Models and a Comparison of Using Different Context Ranges in Convolutional Layers," in *Proc. ICMC*, 2020.

[15] S. Böck and M. Schedl, "Enhanced Beat Tracking with Context-Aware Neural Networks," in *Proc. DAFx*, 2011.

[16] S. Böck, F. Krebs, and G. Widmer, "A Multi-Model Approach to Beat Tracking Considering Heterogeneous Music Styles," in *Proc. ISMIR*, 2014.

[17] ——, "Joint Beat and Downbeat Tracking with Recurrent Neural Networks," in *Proc. ISMIR*, 2016.

[18] F. Krebs, S. Böck, and G. Widmer, "Downbeat Tracking Using Beat-Synchronous Features and Recurrent Networks," in *Proc. ISMIR*, 2016.

[19] M. E. P. Davies and S. Böck, "Temporal Convolutional Networks for Musical Audio Beat Tracking," in *Proc. EUSIPCO*, 2019.

[20] S. Böck, M. E. P. Davies, and P. Knees, "Multi-Task Learning of Tempo and Beat: Learning One to Improve the Other," in *Proc. ISMIR*, 2019, pp. 486–493.

[21] S. Böck and M. E. Davies, "Deconstruct, Analyse, Reconstruct: How to Improve Tempo, Beat, and Downbeat Estimation," in *Proc. ISMIR*, 2020.

[22] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation," in *Proc. ISMIR*, 2018.

[23] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *Proc. IEEE ICASSP*, 2019, pp. 181–185.

[24] C. J. Steinmetz and J. D. Reiss, "WaveBeat: End-to-end beat and downbeat tracking in the time domain," in *151st AES Convention*, 2021.

[25] F. Krebs, S. Böck, and G. Widmer, "An Efficient State-Space Model for Joint Tempo and Meter Tracking," in *Proc. ISMIR*, 2015, pp. 72–78.

[26] X. Sun, Q. He, Y. Gao, and W. Li, "Musical Tempo Estimation Using a Multi-scale Network," in *Proc. ISMIR*, 2021.

[27] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation Methods for Musical Audio Beat Tracking Algorithms," Queen Mary University of London, London, United Kingdom, Tech. Rep. C4DM-TR-09-06, 2009.

[28] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical, and Jazz Music Databases," in *Proc. ISMIR*, 2002, pp. 287–288.

[29] ——, "RWC Music Database: Music Genre Database and Musical Instrument Sound Database," in *Proc. ISMIR*, 2003, pp. 229–230.

[30] F. Gouyon *et al.*, "An Experimental Comparison of Audio Tempo Induction Algorithms," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 5, pp. 1832–1844, 2006.

[31] F. Krebs, S. Böck, and G. Widmer, "Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio," in *Proc. ISMIR*, 2013, pp. 227–232.

[32] S. W. Hainsworth and M. D. Macleod, "Particle Filtering Applied to Musical Tempo Tracking," *EURASIP Journal on Applied Signal Process.*, vol. 15, 2004.

[33] A. Holzapfel *et al.*, "Selective Sampling for Beat Tracking Evaluation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 9, pp. 2539–2548, 2012.

[34] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Speech Audio Process.*, vol. 10, no. 5, 2002.

[35] U. Marchand and G. Peeters, "Swing Ratio Estimation," in *Proc. DAFx*, 2015.

[36] B. D. Giorgi, M. Mauch, and M. Levy, "Downbeat Tracking with Tempo Invariant Convolutional Neural Networks," in *Proc. ISMIR*, 2020, pp. 216–222.

[37] H. Schreiber, J. Urbano, and M. Müller, "Music tempo estimation: Are we done yet?" *Trans. Int. Soc. Music. Inf. Retr.*, vol. 3, no. 1, pp. 111–125, 2020.