Joint Beat and Downbeat Tracking Based on CRNN Models and a Comparison of Using Different Context Ranges in Convolutional Layers

Tian Cheng, Satoru Fukayama, Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{tian.cheng, s.fukayama, m.goto}@aist.go.jp

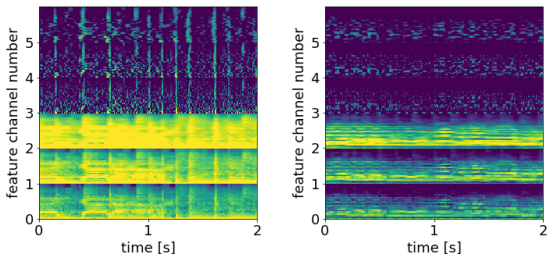
ABSTRACT

In this paper, we address joint beat and downbeat tracking by using Convolutional-Recurrent Neural Networks (CRNNs). The model consists of four convolutional layers and four bi-directional recurrent layers. In order to deal with music in various styles, we propose to increase the convolution filter sizes in the convolutional layers, which helps obtain more context information. We compare four different filter sizes (covering 3 to 9 frames) to analyse the context effect on ten individual datasets. The mean cross validation results of eight datasets show that using context ranges of 5 and 7 frames perform better on downbeat tracking than other context ranges. The comparison results on two testing-only datasets (an in-house pop dataset and the SMC dataset) show the proposed CRNN model outperforms a previous state-of-the-art method with a context range of 7 frames.

1. INTRODUCTION

Beat tracking and downbeat tracking are two fundamental tasks for defining the metrical structure of a music piece [1]. They detect a hierarchical beat structure in two rhythmic levels: beat-level and bar-level [2]. Beat tracking is 'to determine the periodic sequence of beat positions' from a music piece [3], while downbeat tracking is to detect the first beat of each bar. Beats are basic time units for analysing many other musical contents, and downbeats are often related to changes of chords or rhythm patterns. Therefore, beat and downbeat tracking is useful for various tasks, such as music transcription [4], chord estimation [5, 6], structure analysis [7, 8], and so on.

In the current decade, Recurrent Neural Networks (RNNs) have been used for beat tracking with the ability of modelling data in sequences, in which bi-directional Long Short-Term Memory (LSTM) is usually used [9, 10, 11]. For downbeat tracking, a similar RNN model with Gated Recurrent Units (GRUs, a simplified version of the LSTMs) has been used to detect downbeats on beat-synchronised percussive and harmonic features in [12]. Convolutional Neural Networks (CNNs) have also been used for downbeat tracking, with a Hidden Markov Model (HMM) detecting the downbeat positions from various features learned from CNNs [13]. A combination of CNNs and RNNs, Convolutional-Recurrent Neural Networks (CRNNs), have been compared with RNNs for downbeat tracking in [14]. The results show that CRNNs generally work better than RNNs, and provide more robust results on unseen data. Böck et al. propose a joint beat and downbeat tracking model, which applies RNNs to model beat and downbeat probabilities, and uses a Dynamic Bayesian Network (DBN) for detecting beat and downbeat times from the probabilities [1]. There are also models for jointly estimating beats (and downbeats) as well as drums with CRNNs in a multi-task training [15, 16]. With previous work considered, we address joint beat and downbeat tracking by using CRNNs in this paper. Firstly, the CNN layers are used to detect the local events, such as onsets, drums, harmonic changes, and so on. Then, RNN layers work on the CNN output to estimate beats and downbeats globally by modelling the whole sequence.



(a) A pop music clip. (b) A classical music clip.

Figure 1: Spectral representations (as referred to Section 2.1) of two music clips from the RWC pop and RWC classic datasets.

Despite the advantages of CRNNs, one difficulty of beat and downbeat tracking is to deal with music in different styles. It is easier to address dance music (the Ballroom dataset [17, 18]) or pop music (the RWC pop dataset [19]), with percussive onsets and drums, as shown in Fig 1a. However, it is more difficult when dealing with music with soft and blurring onsets, such as music in the RWC classic dataset [19], as shown in Fig 1b. In order to tracking beats and downbeats of music in various styles, [10] proposes a multi-model approach, which trains an individual model on each dataset, and chooses the final result by comparing the results of the individual models to that of a reference model (trained on all datasets). Because of multiple models, this method is more computationally expensive and not extendable.

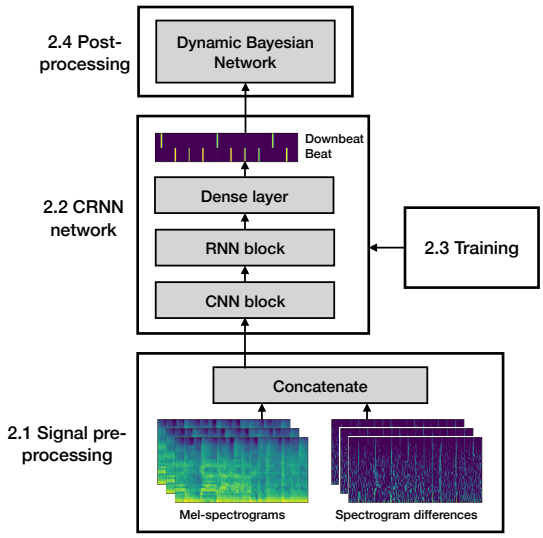


Figure 2: High-level architecture of the CRNN model.

In order to propose a more compact model, we increase the filter sizes in the convolutional layers of the CRNN models to cover a larger context range, and compare four different context ranges (covering 3, 5, 7 and 9 frames, respectively) on ten datasets to analyse the effect of the context range for different music styles. There was work using different context ranges for drum transcription [15], but to the best of our knowledge, there is no such work for beat and downbeat tracking. From the beat tracking results, we find that for music in the RWC classic and SMC datasets [20] with blurring onsets, a context range of 9 frames works best. Using different context ranges has a bigger influence on downbeat tracking than on beat tracking. In general (according to the mean results of eight datasets), context ranges of 5 and 7 frames work better on downbeat tracking than other context ranges.

The proposed models are adapted from the model used in [1], with a similar signal pre-processing and the same post-processing as in [1]. The main difference is that we apply the CRNN and [1] applied the RNN. Besides, we also compare different choices of CNN filters in this paper. In comparison to the method [1] on the testing-only datasets, the proposed models obtain better results on the Songle dataset [21] (an in-house pop music dataset) with a context range of 7 frames, and better results on the SMC dataset with all context ranges. Although the idea of our models is rather simple, we believe that it is worth sharing results of intensive experiments achieving the state-of-the-art performance with our research community to advance beat and downbeat tracking research.

2. CRNN MODEL

We tackle joint beat and downbeat tracking by using CRNNs in this section, with the high-level architecture of the model shown in Figure 2.

2.1 Signal Pre-Processing

We process audio signals to obtain Mel-spectrograms for the input of the CRNN models. Firstly, we read each audio signal into a monotrack waveform sequence with 44100 Hz sampling rate. To obtain the Mel-spectrograms, the signal is segmented into frames of window sizes of 1024, 2048 and 4096 samples with a hop size of 441 samples [1]. For each window size, we compute a magnitude Mel-spectrogram of 36 Mel bins within a frequency range from 30 Hz to 17000 Hz. Then we rescale the magnitude Mel-spectrogram M into a log scale X, as follows:

$$X = \log(1 + M). \tag{1}$$

We also compute the first order difference of the Mel-spectrograms (*DX*):

$$DX = \max(X_t - X_{t-1}, 0). \tag{2}$$

We concatenate the three Mel-spectrograms and their differences into 6 channels, providing an input with a shape of [36, T, 6], where T is the number of frames in a sequence. We show two input examples in Figure 1 with spectrograms in 6 channels stacked together.

2.2 CRNN Network

The CRNN models consist of 4 convolutional layers, 4 bidirectional recurrent layers with GRUs and a full-connected dense layer, with the architecture shown in Table 1. We build the model with tensorflow 1.13.1 [22] and Keras 2.2.4 [23].

The CNN block consists of four convolutional layers, with a maxpooling layer stacked after each of the first two layers, as shown in Table 1a. For the first three layers, we compare four CNN settings with different context ranges. For a context range of X frames, denoted by *cnX*, the width of the convolution filters is X frames, while the heights are set by considering the feature dimensions (36, 18 and 9) of the first three convolutional layers. The detailed parameters are shown in Table 1b.

After the CNN layers, we reshape the output into a tensor of [T, 64] as the input of the RNN layers.

In the preliminary test, we compare RNN architectures of layer number $\in \{3, 4, 5, 6\}$ and GRU number $\in \{32, 64\}$. Finally we use 4 bidirectional layers with 64 GRUs per layer in each direction because this architecture generally works for all above CNN settings.

We stack a dense layer at the end of RNN layers with an output dimension of 3, for the 'no-beat', 'beat' and 'downbeat' labels, respectively.

2.3 Training

Because of the sparse occurrence of the beats and downbeats, we extend the beat and downbeat range for an efficient training [24]. We use an extending range of one frame, meaning that one frame before and one frame after a beat/downbeat are also labelled the same as beat/downbeat.

We train the CRNN models in 8-fold cross validation with random splits by using the datasets in Table 2 without mark $*$. For each split, we use 75% for training, 12.5% for validation and the rest 12.5% for testing. We compare three optimisers (the RMSprop [25], the Adam [26]

Layer	Parameters	Output Shape
input		(36, T, 6)
conv2d_1	(*cnX*)	(36, T, 32)
max_pooling2d_1	(2,1)	(18, T, 32)
conv2d_2	(*cnX*)	(18, T, 32)
max_pooling2d_2	(2,1)	(9, T, 32)
conv2d_3	(*cnX*)	(9, T, 32)
conv2d_4	(64, 9x1)	(1, T, 64)
reshape_1		(T, 64)
bidirectional_1	(64)	(T, 128)
bidirectional_2	(64)	(T, 128)
bidirectional_3	(64)	(T, 128)
bidirectional_4	(64)	(T, 128)
dense_1	(3)	(T, 3)

(a) The architecture of CRNN models. T is the number of frame in a sequence; and *cnX* indicates the parameters of convolutional layers with a context range of *X* frames, as explained in Table 1b.

	cn9	*cn7*	*cn5*	*cn3*
conv2d_1	(32, 7x9)	(32, 7x7)	(32, 5x5)	(32, 3x3)
conv2d_2	(32, 5x9)	(32, 5x7)	(32, 5x5)	(32, 3x3)
conv2d_3	(32, 5x9)	(32, 5x7)	(32, 5x5)	(32, 3x3)
para_num	396,643	373,475	348,387	312,547

(b) Parameters of convolutional layers with different context ranges. 'para_num' indicates the total number of trainable parameters in each model.

Table 1: The architecture of CRNN models.

and stochastic gradient descent) in the preliminary test, and choose the RMSprop with a learning rate of 10^{-3} to minimise the cross entropy error. We stop training if no improvement is found on the validation set in 15 epochs. Then, we fine-tune the models by using the RMSprop with a learning rate of 10^{-4}. The fine-tuning is also stopped with a patient number of 15 epochs.

2.4 Post-Processing

We adapt the post-processing method in [1]. First a threshold of 0.05 is applied on the beat/downbeat activations (from the CRNN models) to delete small activations at the beginning and end of a music piece. Then, a Dynamic Bayesian Network is used to infer the metre, tempo and beat phases jointly based on the observation distributions converted from the beat/downbeat activations. Readers are referred to [1, 27] for more details. In the experiment, we restrict the bar lengths to 2, 3, or 4 beats.

3. EVALUATION

We evaluate beat and downbeat tracking results in F-measures, computed by [28]. A beat/downbeat is considered correct if it falls into a tolerance window of ± 70 ms from the annotation.

3.1 Datasets

We list the datasets used in the experiment in Table 2. The datasets cover a variety of music genres, such as pop, classic, jazz and so on. There are two held-out datasets for testing only, as marked with $*$ in Table 2. One is an in-house dataset (the Songle dataset), with 228 songs registered on [21] and with the annotations manually checked. The other is the SMC dataset, with only beat annotations available, which is considered as a difficult dataset for beat tracking.

Dataset	# files	max_len per piece	total_len
Ballroom [17, 18]	694	30 s	5 h 47 m
GTZAN [29, 30]	1000	30 s	8 h 20 m
Hainsworth [31]	222	60 s	3 h 16 m
RWC classic [19]	50	360 s	4 h 12 m
RWC genre [32]	100	360 s	6 h 31 m
RWC jazz [19]	50	360 s	3 h 34 m
RWC pop [19]	100	360 s	6 h 38 m
RWC royalty [19]	15	180 s	32 m
Songle [21] $*$	228	240 s	13 h 47m
SMC [20] $*$	217	40 s	2 h 25 m

Table 2: A list of datasets. Datasets marked with $*$ are held-out datasets for testing only.

For training, all pieces are segmented into 30 seconds, with a 50% overlap for pieces longer than 30 seconds. For testing either in 8-fold cross-validation or the testing-only datasets, the whole pieces are processed to obtain beats and downbeats.

3.2 Results

We show beat and downbeat tracking results of different datasets in Table 3. The results are obtained by the proposed CRNN models with different context ranges, and compared to previous state-of-the-art methods on several datasets.

3.2.1 A comparison to state-of-the-art

We compare the proposed models to a previous state-of-the-art model [1], which applies RNNs for a joint beat and downbeat tracking model. Among the results on the testing-only datasets (as shown in Table 3a), the proposed models achieve best results on the Songle dataset at a context range of 7 frames, with 0.918 and 0.849 for beat and downbeat F-measures, respectively. This downbeat tracking result is slightly better than that of [1]. All proposed models work better than [1] on the SMC dataset. The best beat F-measure of 0.531 is achieved at a context range of 9 frames, better than the result in [1] by 1.5 percentage points. The testing-only results of the proposed models are competitive to the cross validation result of 0.529 in [10].

From the results obtained with 8-fold cross validation (as shown in Table 3b), we find that the proposed models obtain better results on the Ballroom dataset with all context ranges, with mean F-measures of 0.949 and 0.875 for beat and downbeat tracking, respectively. Both mean results are better than the reported results in [1] by above 1 percentage point. On the RWC pop dataset, the proposed models also outperform [1] on downbeat tracking, with a mean F-measure of 0.882, better than the result in [1] by 2.1 percentage points. However, the beat tracking F-measures of

	Songle		SMC
	F_b	F_d	F_b
cn9	91.3	83.8	**53.1**
cn7	**91.8**	**84.9**	52.7
cn5	91.4	84.8	52.6
cn3	91.3	84.3	52.5
[1]	91.8	84.3	51.6
[10]			52.9 †

(a) Results on testing-only datasets. † denotes results obtained with 8-fold cross validation.

	Ballroom		GTZAN		Hainsworth		RWCpop	
	F_b	F_d	F_b	F_d	F_b	F_d	F_b	F_d
cn9	**95**	87.1	88	67.6	**84.4**	61	93	88.3
cn7	94.9	87.4	**88.5**	**69.2**	84.3	61.1	92.9	88.6
cn5	94.7	**88.2**	88.3	68.3	83.9	**61.8**	93.5	**88.8**
cn3	**95**	87.2	88.3	67.3	83.4	58.4	**93.7**	87
mean	94.9	87.5	88.3	68.1	84	60.6	93.3	88.2
[1]	93.8	86.3	85.6∗	64∗	**86.7**	**68.4**	**94.3**	86.1

(b) Cross validation results on 4 datasets. ∗ denotes testing-only results.

	Mean_len		Mean_num	
	F_b	F_d	F_b	F_d
cn9	**85.9**	72.4	89.2	73.8
cn7	85.5	**72.8**	**89.3**	**74.6**
cn5	85.4	**72.8**	89.1	74.5
cn3	85.7	72.1	89.2	73.4

(c) Mean cross validation results of 8 datasets. 'Mean_len' and 'Mean_num' denote weighted average results, weighted by the total length of songs in the dataset and the number of songs in the dataset, respectively.

Table 3: Results (in percentage) on different datasets. F_b and F_d denote the F-measures of beat and downbeat tracking, respectively. *cnX* represents the CRNN model with the context range of X frames.

the proposed models are worse than the F-measure in [1] (0.943) by around 1 percentage point. On the Hainsworth dataset, the F-measures in [1] (0.867 and 0.684) are better than those of the proposed models, exceeding the mean F-measures by 2.7 and 7.8 percentages on beat and downbeat tracking, respectively. On the GTZAN dataset, the proposed models achieve mean beat and downbeat F-measures of 0.883 and 0.681, respectively; while the corresponding F-measures in [1] (testing-only results) are 0.856 and 0.64 respectively.

3.2.2 Effect of the context range

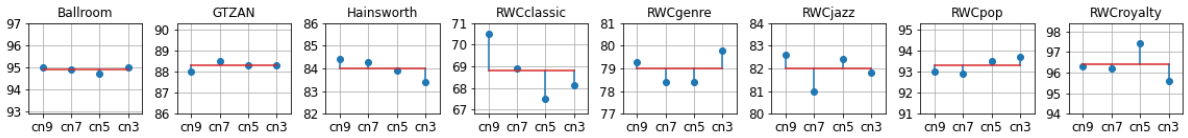
In this subsection, we focus on the change of performance along with the change of the context range. The results of different context ranges are very similar to each other when we look at the mean cross validation results (as shown in Table 3c), and only the mean downbeat tracking results weighted by the song number of the dataset shows relatively better results by using context range of 7 and 5 frames.

For a more detailed analysis, we show cross validation results on individual datasets in Figure 3. We find that the context range has bigger influence on downbeat tracking (Figure 3b) than beat tracking (Figure 3a) except for the RWC classic dataset. The performance variance of each dataset is related to the number of songs in the dataset. For example, the RWC royalty (15 songs), RWC classic (50 songs) and RWC jazz (50 songs) datasets have larger performance variance, while the Ballrrom (694 songs) and GTZAN (1000 songs) datasets have much smaller performance variance. This is because the overall result of a dataset is more likely to be influenced by results of individual songs when the song number is small.

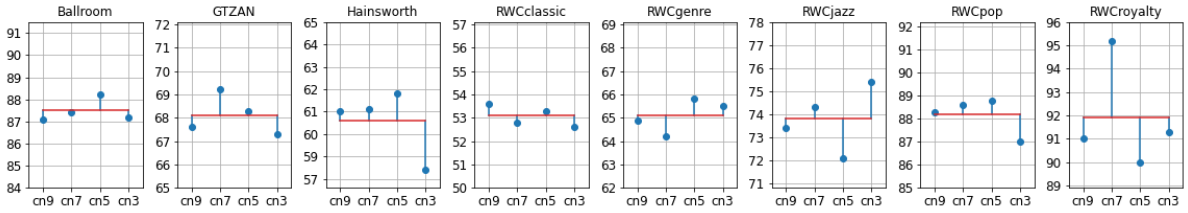
Despite the influence of the song numbers of the datasets, we still can find that for more difficult datasets (the Hainsworth, RWC classic and SMC datasets), it tends to have better beat tracking results with larger context ranges. With regarding to downbeat tracking results in Figure 3b and Table 3a, the Ballroom, Hainsworth, RWC genre, RWC pop datasets have the best results with a context range of 5 frames. The GTZAN, RWC royalty and Songle datasets have the best results with a context range of 7 frames. The RWC classic dataset achieves the best result at a context range of 9 frames, while the RWC jazz dataset achieves the best result at a context range of 3 frames. Despite the best downbeat performance achieved at the context range of 5 frames, we find on the RWC pop and Hainsworth datasets, the downbeat tracking results is significant improved with larger context ranges (*cn5*, *cn7* and *cn9*) in comparison to the results with *cn3*.

From above results, we find that increasing the context range (increasing the parameter number) of the CRNN model doesn't guarantee to increase the beat and downbeat tracking performance. We believe this is because when the data already show clear beat clues, such as the RWC pop music shown in Figure 1a, a small context range is enough to process the information, and a large context range (more parameters) can bring a risk of overfitting. But for more difficult datasets (the RWC classic and SMC datasets), the results indicate increasing the context range can help detect beats and downbeats.

We take an example from the RWC classic dataset to illustrate the effect brought by increasing the context range. In Figure 4, we show the outputs of separated layers of two CRNN models with context ranges of 3 frames (on the left) and 9 frames (on the right) of a classic music clip. For the top sub-figures, we can find that the timing of the onsets or other spectral events is more recognisable on the right side (with a large context range) than on the left side (with a small context range). With a clearer CNN output for the right model, the 2nd RNN layer has already started to recognise the tactus. Then, the 3rd and 4th RNN layers work on detecting the beats and downbeats. In contrast, for the left model, the tactus only starts to become clear in the 4th RNN layer. Hence, based on the output of the left model, double-tempo beats are detected, while in the right model correct beats and better downbeats are detected. The example indicates that using a larger context range in the CNN filters helps to detect the blurring onsets in classic music (as shown Figure 1b), which improves beat and downbeat tracking eventually.

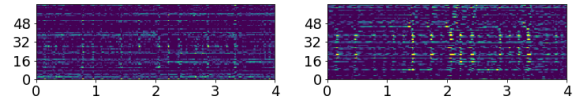


(a) Beat tracking F-measures.

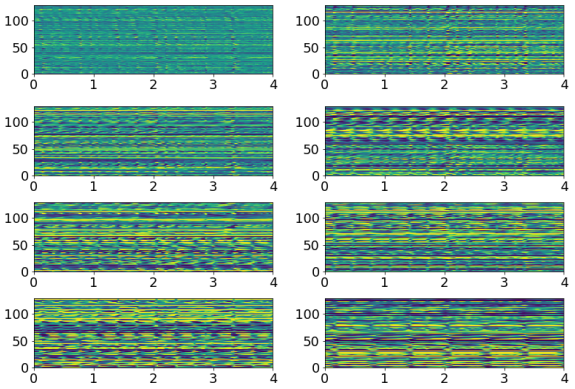


(b) Downbeat tracking F-measures.

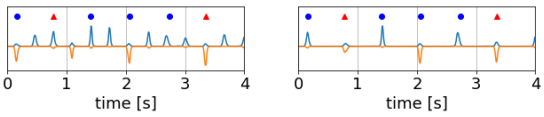
Figure 3: Cross validation results (in percentage) on individual datasets. *cnX* represents the CRNN model with the context range of X frames. Note that the y-axis ranges of sub-figures are different in order to provide a clear illustration.



(a) Output of the reshape layer.



(b) Ouputs of 4 RNN layers.



(c) Output of the dense layer.

Figure 4: Outputs of separated layers of two CRNN models on a music clip from the RWC classic dataset. Left and right denote the outputs of the models with context range of 3 and 9 frames, respectively. In the bottom sub-figures, ● and ▲ denote beat and downbeat annotations, respectively. Blue (up) and orange (down) lines denote estimated beat and downbeat activations, respectively.

4. CONCLUSION

In this paper we apply CRNN models for joint beat and downbeat tracking and compare four models with different convolution filter sizes (covering context ranges of 3, 5, 7 and 9 frames, respectively). We analyse the results of the proposed CRNNs models on ten datasets. In the testing-only datasets, the proposed models outperform a previous state-of-the-art method [1] with a context range of 7 frames on the in-house Songle dataset, and with all context ranges on the SMC dataset. In general, applying different context ranges has a bigger influence on downbeat tracking results than on beat tracking results (in all datasets except for the RWC classic dataset). We observe clear improvements in downbeat tracking performance on the RWC pop and Hainsworth datasets by using larger context ranges (of 5, 7 and 9 frames) instead of a context range of 3 frames. A large context range can improve beat and downbeat performance for difficult datasets (the RWC classic and SMC datasets), but is not optimal for pop and dance music. We could set the default context range to 7 frames for future unseen data for a balance between music in various styles.

We find that in many cases the results obtained with different context ranges are similar to each other. In our analysis, we try to draw conclusion only based on the results where performance differences are clear (larger than 1 percentage) to distinguish from each other, and avoid the influence of the song numbers of the datasets. In order to reflect the context ranges more clearly, we plan to design and test other types of network architectures in the future.

Acknowledgments

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

5. REFERENCES

[1] S. Böck, F. Krebs, and G. Widmer, "Joint Beat and Downbeat Tracking with Recurrent Neural Networks," in *Proc. ISMIR*, 2016.

[2] M. Goto and Y. Muraoka, "Real-Time Beat Tracking

for Drumless Audio Signals: Chord Change Detection for Musical Decisions," *Speech Communication*, vol. 27, pp. 311–335, 1999.

[3] M. Müller, *Fundamentals of Music Processing Audio, Analysis, Algorithms, Applications*. Springer, 2015, ch. 6 Tempo and Beat Tracking.

[4] 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.

[5] M. Mauch and S. Dixon, "Simultaneous Estimation of Chords and Musical Context From Audio," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1280–1289, 2010.

[6] 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 Transactions on Audio, Speech and Language Processing*, vol. 22, no. 2, pp. 556–575, 2014.

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

[8] J. Paulus, M. Müller, and A. Klapuri, "State of the Art Report: Audio-Based Music Structure Analysis," in *Proc. ISMIR*, 2010.

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

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

[11] ——, "Accurate Tempo Estimation Based on Recurrent Neural Networks and Resonating Comb Filters," in *Proc. ISMIR*, 2015, pp. 625–631.

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

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

[14] M. Fuentes, B. Mcfee, H. C. Crayencour, S. Essid, and J. P. Bello, "Analysis of Common Design Choices in Deep Learning Systems for Downbeat Tracking," in *Proc. ISMIR*, 2018.

[15] R. Vogl, M. Dorfer, G. Widmer, and P. Knees, "Drum Transcription via Joint Beat and Drum Modeling Using Convolutional Recurrent Neural Networks," in *Proc. ISMIR*, 2017.

[16] M. Cartwright and J. P. Bello, "Increasing Drum Transcription Vocabulary Using Data Synthesis," in *Proc. DAFx*, 2018.

[17] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An Experimental Comparison of Audio Tempo Induction Algorithms," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[18] 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.

[19] 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.

[20] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. a. L. Oliveira, and F. Gouyon, "Selective Sampling for Beat Tracking Evaluation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.

[21] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A Web Service for Active Music Listening Improved by User Contributions," in *Proc. ISMIR*, 2011, pp. 311–316.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. [Online]. Available: http://tensorflow.org/

[23] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[24] T. Cheng, S. Fukayama, and M. Goto, "Convolving Gaussian Kernels for RNN-based Beat Tracking," in *Proc. EUSIPCO*, 2018, pp. 1919–1923.

[25] T. Tieleman and G. Hinton, "Lecture 6.5-RMSprop: Divide the Gradient by a Running Average of its Recent Magnitude," pp. 26–31, 2012.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] 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.

[28] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A Transparent Implementation of Common MIR Metrics," in *Proc. ISMIR*, 2014.

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

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

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

[32] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Music Genre Database and Musical Instrument Sound Database," in *Proc. ISMIR*, 2003, pp. 229–230.