DOWNBEAT TRACKING USING BEAT-SYNCHRONOUS FEATURES AND RECURRENT NEURAL NETWORKS

Florian Krebs, Sebastian Böck, Matthias Dorfer, and Gerhard Widmer
Department of Computational Perception
Johannes Kepler University Linz, Austria
Florian.Krebs@jku.at

ABSTRACT

In this paper, we propose a system that extracts the downbeat times from a beat-synchronous audio feature stream of a music piece. Two recurrent neural networks are used as a front-end: the first one models rhythmic content on multiple frequency bands, while the second one models the harmonic content of the signal. The output activations are then combined and fed into a dynamic Bayesian network which acts as a rhythmical language model. We show on seven commonly used datasets of Western music that the system is able to achieve state-of-the-art results.

1. INTRODUCTION

The automatic analysis of the metrical structure in an audio piece is a long-standing, ongoing endeavour. A good underlying meter analysis system is fundamental for various tasks like automatic music segmentation, transcription, or applications such as automatic slicing in digital audio workstations.

The meter in music is organised in a hierarchy of pulses with integer related frequencies. In this work, we concentrate on one of the higher levels of the metrical hierarchy, the *measure* level. The first beat of a musical measure is called a *downbeat*, and this is typically where harmonic changes occur or specific rhythmic pattern begin [23].

The first system that automatically detected beats and downbeats was proposed by Goto and Muraoka [15]. It modelled three metrical levels, including the measure level by finding chord changes. Their system, built upon hand-designed features and rules, was reported to successfully track downbeats in $4/4$ music with drums. Since then, much has changed in the meter tracking literature. A general trend is to go from hand-crafted features and rules to automatically learned ones. In this line, rhythmic patterns are learned from data and used as observation model in probabilistic state-space models [23, 24, 28]. Support Vector Machines (SVMs) were first applied to downbeat tracking in a semi-automatic setting [22] and later used in a fully automatic system that operated on several beat-synchronous hand-crafted features [12]. The latter system was later refined by using convolutional neural networks (ConvNets) instead of SVMs and a new set of features [10, 11] and is the current state-of-the-art in downbeat tracking on Western music.

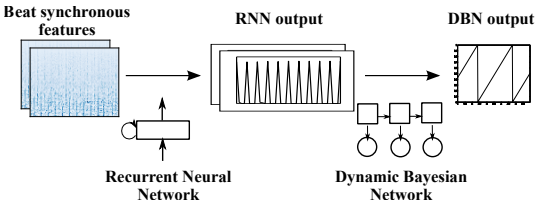


Figure 1: Model overview

Recurrent neural networks (RNNs) are Neural Networks adapted to sequential data and therefore are the natural choice for sequence analysis tasks. In fact, they have shown success in various tasks such as speech recognition [19], handwriting recognition [17] or beat tracking [2]. In this work, we would like to explore the application of RNNs to the downbeat tracking problem. We describe a system that detects downbeats from a beat-synchronous input feature sequence, analyse the performance of two different input features, and discuss shortcomings of the proposed model. We report state-of-the-art performance on seven datasets.

The paper is organised as follows: In Section 2 we describe the proposed RNN-based downbeat tracking system, in Section 3 we explain the experimental set-up of our evaluation and present and discuss the results in Section 4.

2. METHOD

An overview of the system is shown in Fig. 1. Two beat-synchronised feature streams (Section 2.1) are fed into two parallel RNNs (Section 2.2) to obtain a downbeat activation function which indicates the probability whether a beat is a downbeat. Finally, the activation function is decoded into a sequence of downbeat times by a dynamic Bayesian network (DBN) (Section 2.3).

2.1 Feature extraction

In this work we assume that the beat times of an audio signal are known, by using either hand-annotated or automatically generated labels. We believe that the segmentation into beats makes it much more easy for the subsequent stage to detect downbeats because it does not have to deal with tempo or expressive timing on one hand and it greatly reduces the computational complexity by both reducing the sequence length of an excerpt and the search space. Beat-synchronous features have successfully been used before for downbeat tracking [5, 10, 27]. Here, we use two features: A spectral flux with logarithmic frequency spacing to represent percussive content (*percussive feature*) and a chroma feature to represent the harmonic progressions throughout a song (*harmonic feature*).

2.1.1 Percussive feature

As a percussive feature, we compute a multi-band spectral flux: First, we compute the magnitude spectrogram by applying the Short-time Fourier Transform (STFT) with a Hann window, hopsize of 10ms, and a frame length of 2048 samples, as shown in Fig. 2a. Then, we apply a logarithmic filter bank with 6 bands per octave, covering the frequency range from 30 to 17 000 Hz, resulting in 45 bins in total. We compress the magnitude by applying the logarithm and finally compute for each frame the difference between the current and the previous frame. The feature sequence is then beat-synchronised by only keeping the mean value per frequency bin in a window of length Δ_b/n_p, where Δ_b is the beat period and $n_p = 4$ is the number of beat subdivisions, centred around the beginning of a beat subdivision. An example of the percussive feature is shown in Fig. 2b.

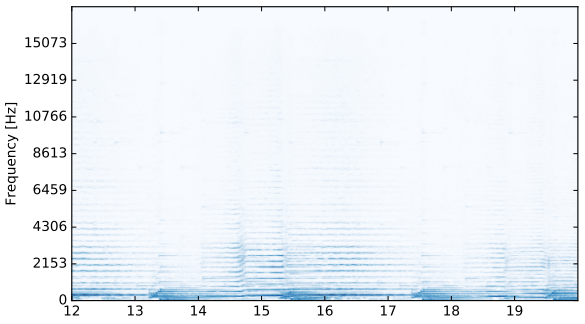
2.1.2 Harmonic feature

As harmonic feature, we use the *CLP* chroma feature [26] with a frame rate of 100 frames per second. We synchronise the features to the beat by computing the mean over a window of length Δ_b/n_h, yielding $n_h = 2$ feature values per beat interval. We found that for the harmonic feature the resolution can be lower than for the percussive feature, as for chord changes the exact timing is less critical. An example of the harmonic feature is shown in Fig. 2d.

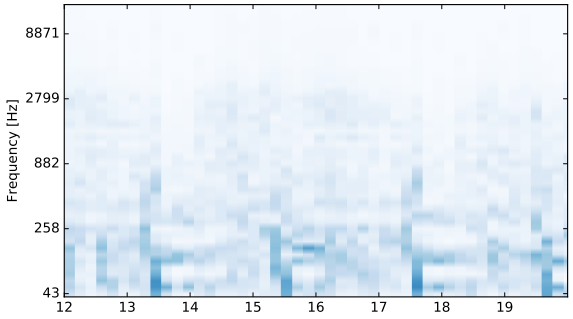
2.2 Recurrent Neural Network

RNNs are the natural choice for sequence modelling tasks but often difficult to train due to the exploding and vanishing gradient problems. In order to overcome these problems when dealing with long sequences, Long-Short-Term memory (LSTM) networks were proposed [20]. Later, [4] proposed a simplified version of the LSTMs named Gated Recurrent Units (GRUs), which were shown to perform comparable to the traditional LSTM in a variety of tasks and have less parameters to train. Therefore, we will use GRUs in this paper.

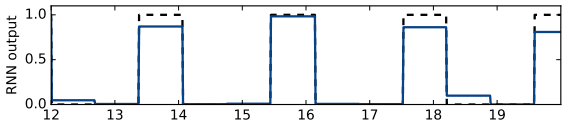
The time unit modelled by the RNNs is the *beat period*, and all feature values that fall into one beat are condensed into one vector. E.g., using the percussive feature with 45



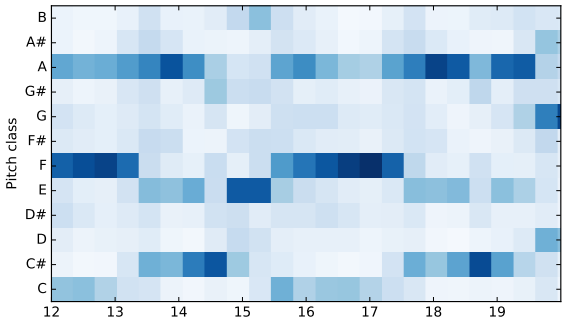
(a) Spectrogram



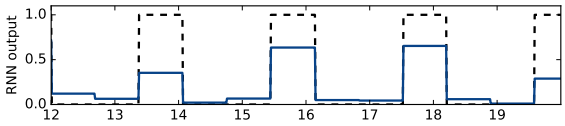
(b) Beat-synchronous percussive feature



(c) Activation and target of the rhythmic network



(d) Beat-synchronous chroma feature



(e) Activation and target of the harmonic network

Figure 2: Visualisation of the two feature streams and their corresponding network output of an 8-second excerpt of the song *Media-105701* (Ballroom dataset). The dashed line in (c) and (e) represents the target (downbeat) sequence, the solid line the networks' activations. The x-axis shows time in seconds. The time resolution is one fourth of the beat period in (b), and half a beat period in (d).

frequency bins and a resolution of $n_p = 4$ beat subdivisions yields an input dimension of $45 \times 4 = 180$ for the rhythmic RNN. In comparison to an RNN that models *subdivisions* of the beat period as underlying time unit, this vectorisation of the temporal context provided an important speed-up of the network training due to the reduced sequence length, while maintaining the same level of performance.

In preliminary tests, we investigated possible architectures for our task and compared their performances on the validation set (see Section 3.3). We made the following discoveries: First, adding bidirectional connections to the models was found to greatly improve the performance. Second, the use of LSTMs/GRUs further improved the performance compared to the standard RNN. Third, using more than two layers did not further improve the performance.

We therefore chose to use a two layer bidirectional network with GRU units and standard tanh non-linearity. Each hidden layer has 25 units. The output layer is a dense layer with one unit and a sigmoid non-linearity. Due to the different number of input units the rhythmic model has approximately 44k, and the harmonic model approximately 19k parameters.

The activations of both the rhythmic and harmonic model are finally averaged to yield the input activation for the subsequent DBN stage.

2.3 Dynamic Bayesian Network

The language model incorporates musical prior knowledge into the system. In our case it implements the following assumptions:

1. Beats are organised into bars, which consist of a constant number of beats.

2. The time signature of a piece determines the number of beats per bar.

3. Time signature changes are rare within a piece.

The DBN stage is similar to the one used in [10], with three differences: First, we model beats as states instead of tatums. Second, as our data mainly contains $3/4$ and $4/4$ time signatures, we only model these two. Third, we force the state sequence to always transverse a whole bar from left to right, i.e., transitions from beat 2 to beat 1 are not allowed. In the following we give a short review of the DBN stage.

A state $s(b, r)$ in the DBN state space is determined by two hidden state variables: the beat counter b and the time signature r. The beat counter counts the beats within a bar $b \in \{1..N_r\}$ where N_r is the number of beats in time signature r. E.g., $r \in \{3, 4\}$ for the case where a 3/4 and a 4/4 time signature are modelled. The state transition probabilities can then be decomposed using

$$P(s_k|s_{k-1}) = P(b_k|b_{k-1}, r_{k-1}) \times P(r_k|r_{k-1}, b_k, b_{k-1}) \tag{1}$$

where

$$P(b_k|b_{k-1}, r_{k-1}) = \begin{cases} 1 & \text{if } b_k = (b_{k-1} \bmod r_{k-1}) + 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Eq. 2 ensures that the beat counter can only move steadily from left to right. Time signature changes are only allowed to happen at the beginning of a bar ($(b_k < b_{k-1})$), as implemented by

$$\begin{aligned} &\text{if} \quad (b_k < b_{k-1}) \\ &\qquad P(r_k|r_{k-1}, b_k, b_{k-1}) = \begin{cases} 1 - p_r & \text{if } (r_k = r_{k-1}) \\ p_r / R & \text{if } (r_k \neq r_{k-1}) \end{cases} \\ &\text{else} \\ &\qquad P(r_k|r_{k-1}, b_k, b_{k-1}) = 0 \end{aligned} \tag{3}$$

where p_r is the probability of a time signature change. We learned p_r on the validation set and found $p_r = 10^{-7}$ to be an overall good value, which makes time signature changes improbable but possible. However, the exact choice of this parameter is not critical, but it should be greater than zero as mentioned in Section 4.5.

As the sigmoid of the output layer of the RNN yields a value between 0 and 1, we can interpret its output as the probability that a specific beat is a downbeat and use it as observation likelihood for the DBN. As the RNN outputs a posterior probability $P(s|\text{features})$, we need to scale it by a factor $\lambda(s)$ which is proportional to $1/P(s)$ in order to obtain

$$P(\text{features}|s) \propto P(s|\text{features})/P(s), \tag{4}$$

which is needed by the observation model of the DBN. Experiments have shown that a value of $\lambda(s(b = 1, r)) = 100$ for downbeat states and $\lambda(s(b > 1, r)) = 1$ for the other states performed best on our validation set, and will be used in this paper.

Finally, we use a uniform initial distribution over the states and decode the most probably state sequence with the Viterbi algorithm.

3. EXPERIMENTS

3.1 Data

In this work, we restrict the data to Western music only and leave the evaluation of Non-Western music for future work. The following datasets are used:

Ballroom [16, 24]: This dataset consists of 685 unique 30 second-long excerpts of Ballroom dance music. The total length is 5h 57m.

Beatles [6]: This dataset consists of 180 songs of the Beatles. The total length is 8h 09m.

Hainsworth [18]: This dataset consists of 222 excerpts, covering various genres. The total length is 3h 19m.

RWC Pop [14]: This dataset consists of 100 American and Japanese Pop songs. The total length is 6h 47m.

Robbie Williams [13]: 65 full songs of Robbie Williams. The total length is 4h 31m

Rock [7]: This dataset consists of 200 songs of the Rolling Stone magazine's list of the "500 Greatest Songs of All Time". The total length is 12h 53m.

System	Ballroom	Beatles	Hainsworth	RWC pop	Robbie Williams	Klapuri	Rock	Mean
With annnotated beats:								
Rhythmic	83.9	87.1	75.7	91.9	93.4	-	87.0	84.4
Harmonic	77.2	89.9	80.1	92.9	92.6	-	86.0	82.2
Combined	91.8	89.6	83.6	94.4	96.6	-	89.4	90.4
With detected beats:								
Combined	80.3	79.8	71.3	82.7	83.4	69.3	79.0	77.3
[11]	77.8	81.4	65.7	86.1	83.7	68.9	81.3	76.1
Beat tracking results:								
Beat tracker [1, 25]	89.0	88.4	88.2	88.6	88.2	85.2	90.5	88.3

Table 1: Mean downbeat tracking F-measures across all datasets. The last column shows the mean over all datasets used. The last row shows beat tracking F-measure scores.

Klapuri [23]: This dataset consists of 320 excerpts, covering various genres. The total length is 4h 54m. The beat annotations of this dataset have been made independently of the downbeat annotations and therefore do not always match. Hence, we cannot use the dataset in experiments that rely on annotated beats.

3.2 Evaluation measure

For the evaluation of downbeat tracking we follow [10, 25] and report the F-measure which is computed by $F = 2RP/(R+P)$, where the recall R is the ratio of correctly detected downbeats within a $\pm 70ms$ window and the total number of annotated downbeats, and the precision P is the ratio of correctly detected downbeats within this window and all the reported downbeats.

3.3 Training procedure

All experiments in this section have been carried out using the leave-one-dataset-out approach, to be as comparable as possible with the setting in [11]. After removing the test dataset, we use 75% of the remaining data for training and 25% for validation. To cope with the varying lengths of the audio excerpts, we split the training data into segments of 15 beats and an overlap of 10 beats. For training, we use cross entropy cost, and AdaGrad [9] with a constant learn rate of 0.04 for the rhythmic model and 0.02 for the harmonic model. The hidden units and the biases are initialised with zero, and the weights of the network are randomly sampled from a normal distribution with zero mean and a standard deviation of 0.1. We stop the learning after 100 epochs or when the validation error does not decrease for 15 epochs. For training the GRUs, we used the Lasagne framework [8].

4. RESULTS AND DISCUSSION

4.1 Influence of features

In this section we investigate the influence of the two different input features described in Section 2.1.

The performance of the two different networks is shown in the upper part of Table 1. Looking at the mean scores over all datasets, the rhythmic and harmonic network achieve a comparable performance. The biggest difference between the two was found in the *Ballroom* and the *Hainsworth* dataset, which we believe is mostly due to differing musical content. While the *Ballroom* set consists of music with clear and prominent rhythm which the percussive feature seems to capture well, the *Hainsworth* set also includes chorales with less clear-cut rhythm but more prominent harmonic content which in turn is better represented by the harmonic feature. Interestingly, combining both networks (by averaging the output activations) yields a score that is almost always higher than the score of the single networks. Apparently, the two networks concentrate on different, relevant aspects of the audio signal and combining them enables the system exploiting both. This is in line with the observations in [11] who similarly combined the output of three networks in their system.

4.2 Estimated vs. annotated beat positions

In order to have a fully automatic downbeat tracking system we use the beat tracker proposed in [1] with an enhanced state space [25] as a front-end to our system. [1] We show the beat tracking F-measures per dataset in the bottom row of Table 1. With regard to beat tracking, the datasets seem to be balanced in terms of difficulty.

The detected beats are then used to synchronise the features of the test set. [2] The downbeat scores obtained with the detected beats are shown in the middle part of Table 1. As can be seen, the values are around $10\% - 15\%$ lower than if annotated beats were used. This makes sense, since an error in the beat tracking stage cannot be corrected in a later stage. This might be a drawback of the proposed system compared to [11], where the tatum (instead of the beat) is the basic time unit and the downbeat tracking stage can still decide whether a beat consists of one, two or more tatums.

Although the beat tracking performance is balanced among the datasets, we find clear differences in the downbeat tracking performance. For example, while the beat tracking performance on the *Hainsworth* and the *Robbie Williams* dataset are similar, the downbeat accuracy differs more than 12%. Apparently, the mix of genres, in-

[1] We use the *DBNBeatTracker* included in *madmom* [3] version 0.13.

[2] We took care that there is no overlap between the train and test sets.

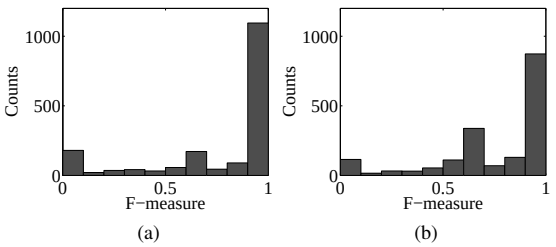


Figure 3: Histogram of the downbeat F-measures of the proposed system (a) and the reference system [11] (b)

cluding time signatures of $2/2$, $3/2$, $3/4$ and $6/8$, in the *Hainsworth* set represents a challenge to downbeat tracking compared to the more simple *Robbie Williams*, which mostly contains $4/4$ time signatures.

4.3 Importance of the DBN stage

System	annotated	detected
RNN	85.0	73.7
RNN+DBN	90.4	77.3

Table 2: Mean downbeat tracking F-measures across all datasets of the proposed, combined system. *annotated* and *detected* means that annotated or detected beats were respectively used to synchronise the features. RNN uses peak-picking to select the downbeats, while RNN+DBN uses the DBN language model.

To assess the importance of the DBN stage (Section 2.3) we implemented a simple baseline, which simply reports downbeats if the resulting (combined) RNN activations exceed a threshold. A threshold of 0.2 was found to yield the best results on the validation set. In Table 2, we show the results of the baseline (RNN) and the results of the combined system (RNN+DBN). As can be seen, the combination of RNN and DBN significantly outperforms the baseline, confirmed by a *Wilcoxon signed-rank* test with $p < 0.01$.

4.4 Comparison to the state-of-the-art

In this section we investigate the performance of our system in relation to the state-of-the-art in downbeat tracking, represented by [11]. Unfortunately, a direct comparison is hindered by various reasons: The datasets used for training the ConvNets [11] are not freely available and the beat tracker at their input stage is different to the one that we use in this work. Therefore, we can only check whether the whole end-to-end system is competitive and leave a modular comparison of the approaches to future work.

In the middle of Table 1 we show the results of the system described in [11], as provided by the authors. The last column shows the mean accuracy over all 1771 excerpts in our dataset. A *paired-sample t-test* did not show any statistically significant differences in the *mean* performance between the two approaches considering all data points. However, a *Wilcoxon signed-rank* test revealed that there is a significant ($p < 0.01$) difference in the *median* F-measure over all data points, which is 89.7% for [11] and 96.2% for the proposed system. Looking at histograms of the obtained scores (see Fig. 3), we found a clear peak at around 66% F-measure, which is typically caused by the beat tracking stage reporting half or double of the correct tempo. The peak is more prominent for the system [11] (Fig. 3b), hence we believe the system might benefit from a more accurate beat tracker.

From this we conclude that overall the proposed system (in combination with the beat tracker [1, 25]) performs comparable to the state-of-the-art when looking at the mean performance and even outperforms the state-of-the-art in terms of the median performance.

4.5 Error analysis

In order to uncover the shortcomings of the proposed model we analysed the errors of a randomly-chosen, small subset of 30 excerpts. We identified two main factors that lead to a low downbeat score. The first one, obviously, are beat tracking errors which get propagated through to the downbeat stage. Most beat tracking errors are octave errors, and among them, the beat tracker mostly tapped twice as fast as the groundtruth tempo. In some cases this is acceptable and therefore would make sense to also allow these metrical levels as, e.g., done in [23]. The second common error is that the downbeat tracker chooses the wrong time signature or has problems following time signature changes or coping with inserted or removed beats. Phase errors are relatively rare. Changing time signatures appear most frequently in the *Beatles* dataset. For this dataset, reducing the transition probability of time signature changes p_r from 10^{-7} to 0 leads to a relative performance drop of 6%, while the results for other datasets remain largely unaffected. Besides, the used datasets mainly contain $3/4$ and $4/4$ time signatures making it impossible for the RNN to learn something meaningful about other time signatures. Here, creating a more balanced training set regarding time signatures would surely help.

5. CONCLUSIONS AND FUTURE WORK

We have proposed a downbeat tracking back-end system that uses recurrent Neural networks (RNNs) to analyse a beat-synchronous feature stream. With estimated beats as input, the system performs comparable to the state-of-the-art, yielding a mean downbeat F-measure of 77.3% on a set of 1771 excerpts of Western music. With manually annotated beats the score goes up to 90.4%.

For future work, a good modular comparison of downbeat tracking approaches needs to be undertaken, possibly with collaboration between several researchers. In particular, standardised dataset train/test splits need to be defined. Second, we would like to train and test the model with non-Western music and 'odd' time signatures, such as done in [21].

The source code will be released as part of the *madmom* library [3], including all trained models and can be found together with additional material under `http://www.cp.jku.at/people/krebs/ismir2016/index.html`.

6. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the GiantSteps project (grant agreement no. 610591), the Austrian Science Fund (FWF) project Z159, the Austrian Ministries BMVIT and BMWFW, and the Province of Upper Austria via the COMET Center SCCH. For this research, we have made extensive use of free software, in particular Python, Lasagne, Theano and GNU/Linux. The Tesla K40 used for this research was donated by the NVIDIA corporation. We'd like to thank Simon Durand for giving us access to his downbeat tracking code.

7. REFERENCES

[1] S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, 2014.

[2] S. Böck and M. Schedl. Enhanced beat tracking with context-aware neural networks. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2011.

[3] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. madmom: a new python audio and music signal processing library. *arXiv:1605.07008*, 2016.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, Dzmitry Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv:1406.1078*, 2014.

[5] M. E. P. Davies and M. D. Plumbley. A spectral difference approach to downbeat extraction in musical audio. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Florence, 2006.

[6] M.E.P. Davies, N. Degara, and M.D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Tech. Rep. C4DM-09-06*, 2009.

[7] T. De Clercq and D. Temperley. A corpus analysis of rock harmony. *Popular Music*, 30(01):47–70, 2011.

[8] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Eric Battenberg, Aäron van den Oord, et al. Lasagne: First release., August 2015.

[9] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[10] S. Durand, J. Bello, D. Bertrand, and G. Richard. Downbeat tracking with multiple features and deep neural networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, 2015.

[11] S. Durand, J.P. Bello, Bertrand D., and G. Richard. Feature adapted convolutional neural networks for downbeat tracking. In *The 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[12] S. Durand, B. David, and G. Richard. Enhancing downbeat detection when facing different music styles. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3132–3136. IEEE, 2014.

[13] B. D. Giorgi, M. Zanoni, A. Sarti, and S. Tubaro. Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony. In *Proceedings of the 8th International Workshop on Multidimensional Systems*, 2013.

[14] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical and jazz music databases. In *Proceedings of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*, Paris, 2002.

[15] M. Goto and Y. Muraoka. Real-time rhythm tracking for drumless audio signals: Chord change detection for musical decisions. *Speech Communication*, 27(3):311–335, 1999.

[16] 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*, 14(5):1832–1844, 2006.

[17] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009.

[18] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 2004:2385–2395, 2004.

[19] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Ng. Deep speech: Scaling up end-to-end speech recognition. *arXiv:1412.5567v2*, 2014.

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] A. Holzapfel, F. Krebs, and A. Srinivasamurthy. Tracking the "odd": Meter inference in a culturally diverse music corpus. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, 2014.

[22] T. Jehan. Downbeat prediction by listening and learning. In *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*, pages 267–270. IEEE, 2005.

[23] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2006.

[24] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, 2013.

[25] F. Krebs, S. Böck, and G. Widmer. An efficient state space model for joint tempo and meter tracking. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, 2015.

[26] Meinard Müller and Sebastian Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, Miami, 2011.

[27] H. Papadopoulos and G. Peeters. Joint estimation of chords and downbeats from an audio signal. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):138–152, 2011.

[28] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: theory and large-scale evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, (99):1–1, 2011.