WAVEBEAT: END-TO-END BEAT TRACKING WITH TEMPORAL CONVOLUTIONAL NETWORKS

Christian J. Steinmetz
Centre for Digital Music, Queen Mary University of London

ABSTRACT

While deep learning approaches for beat and downbeat tracking have brought advancements, these approaches continue to rely on hand-crafted, subsampled spectral features as input, restricting the information available to the model. In this work, we propose WaveBeat, an end-to-end approach for joint beat and downbeat tracking operating directly on waveforms. This method forgoes engineered spectral features, and instead produces beat and downbeat predictions directly from the waveform, the first of its kind for this task. Our model utilizes temporal convolutional networks (TCNs) operating on waveforms that achieve a very large receptive field (≥ 30 s) at audio sample rates in a memory efficient manner by employing rapidly growing dilation factors with fewer layers. With a straightforward data augmentation strategy our method outperforms previous state-of-the-art methods on some datasets, while producing comparable results on others.

1. INTRODUCTION

Beat tracking involves estimating a sequence of time instants that reflect how a human listener may tap along with a musical piece. Downbeat tracking extends this by requiring the estimation not only of the beat locations, but specifically the locations of beats correspond to the first beat within each bar. Such a system has applications across music signal processing including automatic transcription [1] chord recognition [2], and music similarity [3].

Early signal processing approaches generally utilized a two-stage pipeline composed of an onset detection function followed by a post-processing phase to determine which onsets correspond to beats, often incorporating musical knowledge [3–5]. With the rise of deep learning, state-of-the-art systems have now in contrast adopted predominately data driven approaches. While these approaches have demonstrated superior performance, they still employ traditional techniques, namely the use of hand-crafted magnitude spectrograms as input features, along with specialized post-processing [6]. The use of such features facilitates the construction of models with a large receptive field, but discards a significant amount of information in the process. The additional information contained within the time domain signal may in fact be relevant for the beat tracking task, but likely requires larger models, with more compute and training data [7].

In this work, we investigate jointly predicting beat and downbeat events directly from raw audio. This enables us to forgo engineered features, and additionally take advantage of information within the phase of the input, which has been shown to be useful in traditional approaches [8]. We employ a specialized TCN that achieves significant receptive whilst operating on audio waveforms with the use of rapidly growing dilation factors [9] and strided convolutions. Combined with appropriate data augmentation strategies, our proposed model, WaveBeat, outperforms previous deep learning approaches on a number of datasets in both beat and downbeat tracking. While these results are promising, we find our model struggles to generalize to unseen datasets. This indicates that while superior performance can be achieved, generalization still lags behind spectrogram-based approaches.

2. RELATED WORK

Deep learning approaches have subsumed traditional signal processing methods in a number of MIR tasks with beat tracking being no exception. Recurrent networks were first shown to be successful in the beat tracking task a decade ago [10], and have now been extended through a number of iterations [11, 12]. More recently, Davies and Böck [13] identified that while recurrent models are successful in this task, they are often significantly slower to train compared to convolutional networks. Therefore, they proposed the application of temporal convolutional networks (TCNs) [14] to the beat tracking task, demonstrating comparable results to recurrent models, yet with multiple orders of magnitude improvement in efficiency.

Davies and Böck have since improved upon TCN approaches with a mutli-task formulation, which aims to perform other related tasks such as tempo induction, with the hopes of infusing further rhythmic knowledge in the model [6, 15]. In another direction, Di Giorgi el al. [16] proposed a tempo-invariant convolutional architecture that demonstrated improved generalization performance. Nevertheless, these approaches still rely upon hand-crafted spectral magnitude features as input along with specialized post-processing networks. This potentially restricts the model, a limitation we aim to address in this work.

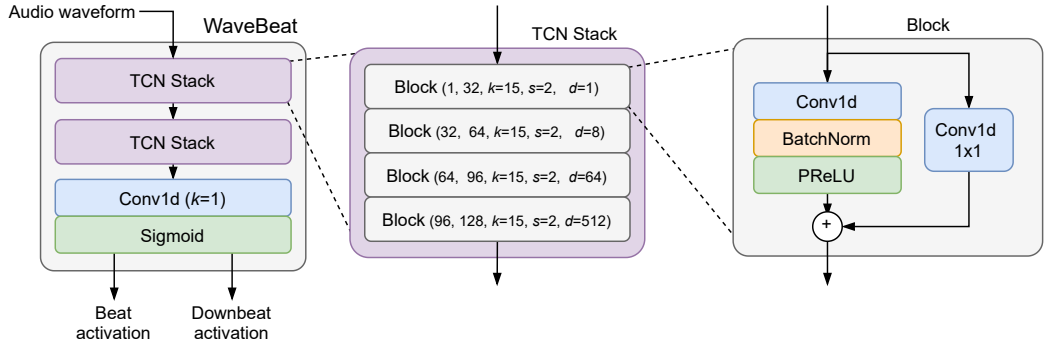

Figure 1. The WaveBeat architecture composed of strided 1-dimensional convolutions with increasing dilation factors.

3. PROPOSED MODEL

Deep learning approaches based on spectral representations offer the ability to efficiently reduce the dimensionality of input features through temporal subsampling and the removal of phase information. This ultimately simplifies the modeling task by utilizing engineered features that are believed to be informative for the task at hand. Nevertheless, these potential benefits can also be seen as a disadvantage. By design they restrict the information available to the model, making strong assumptions about the optimal input representation. This has motivated the design of end-to-end models that instead learn directly from raw audio. Applications of end-to-end architectures in audio include music auto tagging [7, 17], speech enhancement [18, 19], and music source separation separation [20, 21]. End-to-end approaches have yet to be applied to beat and downbeat tracking, and therefore, in this work we aim to explore the design of such an architecture.

The beat and downbeat tracking task is unique compared to the aforementioned audio signal processing tasks. Unlike approaches such as source separation and enhancement, the output is not a waveform, but instead a beat or downbeat activation function. In addition, a significant context of the surrounding music on the order of tens of seconds is generally required to make an accurate estimation of the presence of the downbeat [22]. In the case of convolutional architectures, the receptive field dictates the size of this context. Achieving a window of context up to 30 seconds with a waveform input often requires a receptive field with upwards of 1 M timesteps, imposing a significant comput and memory footprint. This challenge has likely been a dominating factor in the use of spectral features instead of waveforms for beat tracking models.

3.1 Architecture

To address these challenges our network incorporates two major elements: convolutions with rapidly growing dilation patterns paired with carefully designed downsampling operations. We begin with a basic TCN composed of residual 1-dimensional convolutional layers that include batch normalization followed by an activation, as shown on the right of Figure 1. We first consider that is not useful to produce a beat activation function at audio sample rates, as the beat annotations are likely only accurate within tens of milliseconds [4]. This can be handled by downsampling the signal through the depth of the network by employing strided convolutions. With 8 layers, each with stride 2, we downsample the signal by a factor of $2^8 = 256$, which, given an input sample rate of 22.05 kHz produces an output signal with a sample rate of 86 Hz. We chose this since it achieves a beat activation with a sample rate close to those of previous works, which tend to be around 100 Hz [12].

With this architecture, using a generous kernel size of 15 samples, we achieve a receptive of only around 160 ms, which is significantly smaller than desired. To address this, we employ exponentially increasing dilation factors, which are characteristic of the TCN. In nearly all cases, a dilation pattern is employed such that the dilation factor at each layer is given by $d_l = 2^{l-1}$. This convention is likely a result of the approach introduced in [23], which was popularized for audio in the seminal WaveNet architecture [24]. In the case of our 8 layer model, this achieves a much larger receptive field of around 14 seconds. Nevertheless, this still falls short of previous spectrogram-based methods that achieve a receptive field of over one minute [15].

To achieve an even larger receptive field whilst keeping the depth of the network fixed, we consider utilizing a dilation pattern that grows more rapidly. While recent approaches in speech synthesis have investigated dilation factors that grow at a rate given by $d_l = 3^{l-1}$, such models only achieve a receptive field on the order of seconds at audio sampling rates [25, 26]. Inspired by recent work in audio effect modeling that demonstrated the use of dilation factors that grow even more rapidly [9], we consider increasing the growth such that the dilation factor at each layer is given by $d_l = 8^{l-1}$. This enables us to construct a network operating on waveforms that achieves a receptive field of of over 1 M timesteps, $\approx 47\ s$, using only 8 layers.

A block diagram of the complete architecture is shown in Figure 1. In this case, four blocks constitute a stack, after which the dilation pattern is repeated, a common approach [19]. Two stacks are used for a total of 8 blocks, where the number of convolutional channels begins at 32, and increases by 32 at each block. This places more model capacity deeper in the network, where the sample rate is much lower, hence providing a more memory efficient configuration. This results in a model with a total of 2.9 M trainable parameters. As shown on the right of Figure 1, residual connections are implemented as grouped

1x1 convolutions between the input and output of each block, along with BatchNorm, to stabilize training. Parameterized ReLU (PReLU) [27] activation functions are employed, which allow for the slope of the negative side of the ReLU to be learned during training.

3.2 Data augmentation

While end-to-end approaches are more expressive then their counterparts that rely upon spectral features, they often require significantly more training data [7]. Due to the limited music data available with beat and downbeat annotations, we found data augmentation critical in curbing overfitting. We employ a set of fairly common data augmentations while training, each of which has an associated probability, p of being applied. This includes the applications of highpass and lowpass filters with random cutoff frequencies ($p = 0.25$), random pitch shifting between -8 and 8 semitones ($p = 0.5$), additive white noise ($p = 0.05$), applying a tanh nonlinearity ($p = 0.2$), shifting the beat locations forward or back by a random amount between $\pm$ 70 ms ($p = 0.3$), dropping a contiguous block of audio frames and beats of no more than 10% of the input ($p = 0.05$), as well as a random phase inversion ($p = 0.5$).

3.3 Loss function

It is common to utilize the binary cross-entropy where the output of the model should be 1 at the location of a beat, and 0 elsewhere, given by

$$\mathcal{L}_{\text{BCE}}(\hat{y}, y) = -\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n \log(y_n) + (1-\hat{y}_n)\log(1-y_n)),$$

where N is the number of timesteps. This tends to lead to a class imbalance across the temporal dimension, since there are many more locations with no beat. In practice, we found this often encourages the model to never detect beat activations, since such a solution will minimize the loss due to the small number of downbeat activations.

To address this, we adapt the mean false error [28], a metric introduced to handle such class imbalances. Based upon the binary cross-entropy, we compute the loss as a sum of two terms that relate to the false-positive error and false-negative error

$$\mathcal{L}_{\text{FPE}}(\hat{y}, y) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\text{BCE}}(\hat{y}_i, y_i)$$

$$\mathcal{L}_{\text{FNE}}(\hat{y}, y) = \frac{1}{P}\sum_{i=1}^{P}\mathcal{L}_{\text{BCE}}(\hat{y}_i, y_i)$$

$$\mathcal{L}_{\text{MFE}} = \mathcal{L}_{\text{FPE}} + \mathcal{L}_{\text{FNE}},$$

where N is the number of negative examples (no beat), and P is the number of positive examples (beat). This loss function attempts to balance performance by computing the error as a sum of the average error at locations in the prediction where a beat should be present, as well as the average error where there should be no beat. This encourages the model to not converge to a solution of only predicting the majority class, i.e. the absence of a beat.

4. EXPERIMENTS

4.1 Datasets

In order to investigate the performance of the proposed model across a number of styles and audio sources, we consider six popular beat tracking datasets: *Beatles* [29], *Hainsworth* [30], *Ballroom* [31] [32], *RWC Popular* [33], *SMC* [34], and *GTZAN* [35, 36]. For those datasets that include audio at a sampling rate of 44.1 kHz, we resample the audio to 22.05 kHz. Similar to previous works, we train using four datasets (*Beatles*, *Hainsworth*, *Ballroom*, and *RWC Popular*,), and evaulate using two datasets that were not seen during trianing (*SMC* and *GTZAN*). It should be noted that Böck and Davies [6] train with an additional three datasets (*SMC* [34], *HJDB* [37], and *Simac* [38]), amounting to an additional 9 hours of training data, yet we employ more extensive data augmentation.

4.2 Training

We utilize Adam with an initial learning rate of $1e^{-3}$, decreasing the learning rate by a factor of 10 after the beat and downbeat F-measure has not improved on the validation set for 10 epochs. To stabilize training we apply gradient clipping when the norm of the gradients exceeds 4. All models are trained with a batch size of 16 with inputs of 2097152 samples ($\approx$1.6 min at 22.05 kHz) for a total of 100 epochs. In order to balance the influence of the datasets while training, we define a single epoch to constitute 1000 random excerpts with replacement from each dataset. Additionally, we use automatic mixed precision to decrease training time and memory consumption. Code for these experiments is made available, along with the predictions made on the test sets [1] .

4.3 Evaluation

We split the four training datasets into train/val/test sets (80/10/10). We utilize the standard distance threshold of $\pm$70 ms and report the F-measure, CMLt, and AMLt metrics for both beat and downbeat tracking on the test set in Table 1. (In the future, for the final paper, we plan to perform an 8-fold cross validation similar to that proposed in [6]). In the bottom of Table 1 we report results on the left-out datasets (*GTZAN* and *SMC*), in order to test the generalization capability of our model.

On the *Ballroom* dataset we find that WaveBeat achieves comparable results to [6] on the beat tracking task, but achieves an improvement of 4% in downbeat tracking. Similarly, WaveBeat achieves an improvement of over 7% and 23% on the *Hainsworth* dataset for beat and downbeat tracking, respectively. While WaveBeat produces strong results on beat tracking on the *Beatles* dataset, its performance is somewhat worse than [6] on downbeat tracking. Results with WaveBeat begin to fall behind the previous approach when generalizing to data from a set not seen during training. WaveBeat achieves comparable, yet lower results on the *GTZAN* dataset on both beat

[1] https://github.com/csteinmetz1/wavebeat

			Beat			Downbeat		
Dataset	Size	Model	F-measure	CMLt	AMLt	F-measure	CMLt	AMLt
Ballroom	5 h 57 m	Böck and Davies [6]	0.962	0.947	0.961	0.916	0.913	0.960
		WaveBeat (Peak)	0.961	0.929	0.929	0.904	0.762	0.803
		WaveBeat (DBN)	0.925	0.829	0.937	0.953	0.916	0.941
Hainsworth	3 h 19 m	Böck and Davies [6]	0.902	0.848	0.930	0.722	0.696	0.872
		WaveBeat (Peak)	0.965	0.937	0.937	0.912	0.748	0.843
		WaveBeat (DBN)	0.973	0.976	0.976	0.954	0.886	0.970
Beatles	8 h 09 m	Böck and Davies [6]	-	-	-	0.837	0.742	0.862
		WaveBeat (Peak)	0.887	0.733	0.790	0.689	0.327	0.585
		WaveBeat (DBN)	0.929	0.894	0.894	0.732	0.509	0.724
GTZAN	8 h 20 m	Böck and Davies [6]	0.885	0.813	0.931	0.672	0.640	0.832
		WaveBeat (Peak)	0.825	0.682	0.767	0.563	0.279	0.515
		WaveBeat (DBN)	0.828	0.719	0.860	0.598	0.503	0.764
SMC	2 h 25 m	Böck and Davies [6]	0.544	0.443	0.635	-	-	-
		WaveBeat (Peak)	0.403	0.163	0.255	-	-	-
		WaveBeat (DBN)	0.418	0.280	0.419	-	-	-

Table 1. Beat and downbeat tracking results. The *GTZAN* and *SMC* datasets were not seen during training of the WaveBeat model. *NB: At the moment these results reflect evaluation with only a single fold for the WaveBeat model.*

and downbeat tracking. On the *SMC* dataset, which contains a number of challenging pieces, WaveBeat performers clearly worse than previous approaches on beat tracking. Results on the downbeat tracking task are omitted for this dataset due to the absence of downbeat annotations.

4.4 Post-processing

Existing beat tracking systems generally utilize a post-processing stage which inspects the beat activation functions in order to select beat locations, commonly a dynamic Bayesian network (DBN) [12], or conditional random fields (CRF) [39]. Ideally, an end-to-end model would be able to forgo such post-processing. We analyze the beat and downbeat activation functions from WaveBeat using first simple peak picking, selecting peaks with an amplitude greater than 0.5. We then utilize the pre-trained DBN in the `madmom` library [40] in order to examine the performance improvement. As shown in Table 1, we find that while the pre-trained DBN brings about a small improvement in the F-measure, our end-to-end model achieves comparable performance using simple peak peaking. This contrasts with previous approaches, which often report up to 15% improvement with such post-processing [12].

5. DISCUSSION

While these results are encouraging, indicating that waveform based approaches may be a pathway towards improving beat and downbeat tracking systems, they also pose a challenge in learning from limited annotated data. Due to the significant effort required in producing beat and downbeat annotations, it is unlikely that building larger datasets is a desirable route to improving performance. The adoption of recently introduced multi-task formulations or tempo-invariant architecture may further improve robustness to music from unseen distributions, but they likely will not be capable of addressing generalization to music styles outside the training distribution.

Improving the performance of end-to-end models likely requires orders of magnitude more data. We propose the use of self-supervised learning [41], which can leverage large music copra without annotations. This requires appropriate pretext tasks such that we can pre-train a model on unlabeled data and utilize transfer learning to fine-tune this model with available beat tracking datasets. By exposing the model to a diverse collection of music styles during pre-training, the fine-tuned model is likely to generalize better to examples outside the limited annotated datasets.

6. CONCLUSION

We demonstrated for the first time the ability of an end-to-end model to learn directly from waveforms on the joint beat and downbeat tracking task. We find that our model is able to outperform existing state-of-the-art methods for beat and downbeat tracking on a number of common datasets. We additionally investigate the requirement for specialized post-processing networks in the task of locating beat activation functions and find that our model performs well without such post-processing using very simple peak picking. While these results are promising, indicating that waveform based approaches can outperform existing spectrogram-based methods, additional work is needed to improve the generalization ability of these approaches. Future work involves the addition of a more rigorous data augmentation strategy, along with the design of pretext tasks to employ self-supervised pre-training in hopes of leveraging large music copra without annotations.

7. ACKNOWLEDGEMENT

This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (EP/S022694/1).

8. REFERENCES

[1] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[2] B. Di Giorgi, M. Zanoni, A. Sarti, and S. Tubaro, "Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony," in *nDS'13; Proceedings of the 8th International Workshop on Multidimensional Systems*. VDE, 2013, pp. 1–6.

[3] D. P. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[4] S. Dixon, "Evaluation of the audio beat tracking system BeatRoot," *Journal of New Music Research*, vol. 36, no. 1, pp. 39–50, 2007.

[5] M. E. Davies and M. D. Plumbley, "Context-dependent beat tracking of musical audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007.

[6] S. Böck and M. E. P. Davies, "Deconstruct, analyse, reconstruct: how to improve tempo, beat, and downbeat estimation," in *ISMIR*, 2020.

[7] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *ISMIR*, 2018.

[8] D. Eck, "Beat tracking using an autocorrelation phase matrix," in *ICASSP*, vol. 4, 2007.

[9] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time analog audio effect modeling," *arXiv:2102.06200*, 2021.

[10] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *DAFx*, 2011.

[11] S. Böck, F. Krebs, and G. Widmer, "A multi-model approach to beat tracking considering heterogeneous music styles." in *ISMIR*, 2011.

[12] ——, "Joint beat and downbeat tracking with recurrent neural networks." in *ISMIR*, 2016.

[13] M. E. P. Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *EUSIPCO*, 2019.

[14] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv:1803.01271*, 2018.

[15] S. Böck, M. E. P. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other." in *ISMIR*, 2019.

[16] B. Di Giorgi, M. Mauch, and M. Levy, "Downbeat tracking with tempo-invariant convolutional neural networks," in *ISMIR*, 2020.

[17] J. Lee, J. Park, K. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," in *SMC Conference*, 2017.

[18] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," *arXiv:1703.09452*, 2017.

[19] D. Rethage, J. Pons, and X. Serra, "A WaveNet for speech denoising," in *ICASSP*, 2018.

[20] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," in *ISMIR*, 2018.

[21] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv:1911.13254*, 2019.

[22] M. Fuentes, B. Mcfee, H. C. Crayencour, S. Essid, and J. P. Bello, "A music structure informed downbeat tracking system using skip-chain conditional random fields and deep learning," in *ICASSP*, 2019.

[23] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[24] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[25] Q. Tian, Y. Chen, Z. Zhang, H. Lu, L. Chen, L. Xie, and S. Liu, "TFGAN: Time and frequency domain based generative adversarial network for high-fidelity speech synthesis," *arXiv:2011.12206*, 2020.

[26] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, "Multi-band MelGAN: Faster waveform generation for high-quality text-to-speech," *arXiv:2005.05106*, 2020.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[28] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *IJCNN*, 2016.

[29] 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-TR-09-06, 2009.

[30] S. W. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 1–11, 2004.

[31] 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.

[32] F. Krebs, S. Böck, and G. Widmer, "Rhythmic pattern modeling for beat and downbeat tracking in musical audio." in *ISMIR*, 2013.

[33] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases." in *ISMIR*, 2002.

[34] A. Holzapfel, M. E. Davies, J. R. Zapata, J. 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.

[35] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[36] U. Marchand, Q. Fresnel, and G. Peeters, "GTZAN-Rhythm: Extending the GTZAN test-set with beat, downbeat and swing annotations," in *ISMIR*, 2015.

[37] J. Hockman, M. E. Davies, and I. Fujinaga, "One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass." in *ISMIR*, 2012.

[38] F. Gouyon *et al.*, *A computational approach to rhythm description-Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. Universitat Pompeu Fabra, 2005.

[39] F. Korzeniowski, S. Böck, and G. Widmer, "Probabilistic extraction of beat positions from a beat activation function." in *ISMIR*, 2014.

[40] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new python audio and music signal processing library," in *ACM Multimedia*, 2016.

[41] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *IEEE/CVF*, 2020.