FEATURE ADAPTED CONVOLUTIONAL NEURAL NETWORKS FOR DOWNBEAT TRACKING

Simon Durand, Juan P. Bello†, Bertrand David*, Gaël Richard**

* LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France
† Music and Audio Research Laboratory (MARL), New York University – USA

ABSTRACT

We define a novel system for the automatic estimation of downbeat positions from audio music signals. New rhythm and melodic features are introduced and feature adapted convolutional neural networks are used to take advantage of their specificity. Indeed, invariance to melody transposition, chroma data augmentation and length-specific rhythmic patterns prove to be useful to learn downbeat likelihood. After the data is segmented in tatums, complementary features related to melody, rhythm and harmony are extracted and the likelihood of a tatum being at a downbeat position is computed with the aforementioned neural networks. The downbeat sequence is then extracted with a flexible temporal hidden Markov model. We then show the efficiency and robustness of our approach with a comparative evaluation conducted on 9 datasets.

Index Terms— Downbeat Tracking, Music Information Retrieval, Music Signal Processing, Convolutional Neural Networks

1. INTRODUCTION

Music is often organized into structural units at different time scales. One such unit is the measure, or bar, which contains patterns of predefined length in beats, accentuated to define the meter or rhythmic structure of the piece. The downbeats mark the boundaries of these measures, and their automatic detection is useful for various applications in music information retrieval, computer music and computational musicology. Downbeat tracking has received a lot of attention recently with new systems exploring novel temporal models [1] and application to specific music styles [2] [3].

Our recent work [4] explored the use of multiple, complementary signal features encoding various properties connected with downbeats. In that approach, local feature sequences were independently modeled using deep belief networks, both learning higher level features and estimating the likelihood of downbeats. Results show state of the art performance for a variety of Western music styles [4][1]. However, this study neglected to explore how models can be adapted to the specificities of each feature sequence. In other words, the same network configurations were used regardless of whether they were attempting to represent different harmonic, rhythmic or timbral cues. We believe that this imposes limitations on the musical attributes that can be modeled, as well as the optimality of the existing models.

In this paper we aim to expand on our previous work by proposing a few alternative model configurations, each adapted to how different features represent downbeats and metrical structure. More specifically, we make significant improvements to our previous models of harmonic and rhythmic information, and introduce a novel approach to downbeat tracking using melodic cues, an attribute that has been shown to be important for the characterization of metrical structure [5], but remains largely unexplored for computational approaches. Our solutions make use of deep, convolutional neural networks (CNN) both as single and multi-label classifiers, which constitutes, to the best of our knowledge, the first application of CNNs to this task. Our experiments show a significant performance improvement upon past approaches, including our own, on a variety of datasets of annotated music.

The rest of this paper is organized as follows: Section 2 briefly describes our previous approach, emphasizing commonalities and differences with the current work. Section3 describes the details and motivation behind each of the proposed models. Section 4 presents our methodology and the results of our evaluation, and discusses the meaning and significance of those results. Finally, Section 5 includes our conclusions and ideas for future work.

2. PREVIOUS APPROACH

In [4], we use a pulse estimation approach [6] to segment the signal into short temporal units that can be interpreted as tatums. Downbeat tracking is then reduced to a sequence labeling problem where each tatum is either a downbeat or not. We compute 6 low-level features related to harmony, timbre, rhythm, bass content and similarity in timbre and harmony and map them to the pre-computed tatum grid. For each feature series we extract overlapping sub-sequences centered on the position of the candidate downbeat, and use them as input to a fully-connected deep belief network. Network configurations are the same for each feature. Each network estimates the likelihood of a tatum to be at a downbeat position and their outputs are averaged to obtain an overall estimation. The final downbeat sequence is decoded using a hidden Markov model with a uniform initial distribution, states modeling measures of different length and transitions taking into account that changes in time signature are possible albeit unlikely.

In this paper we will use the same tatum segmentation, fusion of the classifiers and temporal modeling as in [4]. The following section discusses the new feature and model configurations that are the central focus of this work.

3. FEATURE ADAPTED CONVOLUTIONAL NEURAL NETWORKS

3.1. Convolutional Neural Networks

CNN are deep neural networks characterized by their convolutional layers [7]. At each layer i, the intermediary input tensor X_i of di-

This article is partly funded by the Futur et Ruptures program of the Institut Mines-Télécom within the DeepMIR project

[1] http://www.music-ir.org/mirex/wiki/2014:Audio_Downbeat_Estimation_Results

mension $[N_i, M_i, P_i]$ is mapped into an output X_{i+1} with a non linear function $f_i(X_i|\theta_i, p_i)$, with $\theta_i = [W_i, b_i]$ the learned layer parameters composed of biases b_i and filters W_i, and p_i the designed parameters related to the network architecture:

$$X_{i+1} = f_i(X_i|\theta_i, p_i) = h_i(c_i(X_i, \theta_i, p_{1i}), p_{2i});\ \forall i \in [0..L-1] \quad (1)$$

where $p_{1i} = [x_{1i}, y_{1i}, P_i, n_i]$ is a designed set of parameters, with x_{1i} and y_{1i} the temporal and vertical dimensions of the filter, P_i the depth of X_i, and n_i the number of filters. c_i is a convolution operator:

$$c_i = b_i[z'] + \sum_{x=1}^{x_{1i}} \sum_{y=1}^{y_{1i}} \sum_{z=1}^{P_i} W_i[x, y, z, z'] X_i[x' + x - 1, y' + y - 1, z] \quad (2)$$

where $x' \in [1..N_{i+1}]$, $y' \in [1..M_{i+1}]$ and $z' \in [1..n_i]$. $L = 4$ is the number of layers of the network, and h_i is in our case a set of one or several cascaded non linear functions among rectified linear units r [8], sigmoids σ, max pooling m, softmax normalization s and dropout regularization d [9]. $p_{2i} = [x_{2i}, y_{2i}]$ is the designed set of parameters of h_i corresponding in our case to the temporal and vertical dimension of the max pooling. X_0 will be our musical input of dimension $[N_0, M_0, 1]$ related to harmony, melody or rhythm and described below. X_L will be the final output and will act as a downbeat likelihood. The network will be trained by minimizing the negative log-likelihood of the correct class or the Euclidean distance between the output and the ground truth by stochastic gradient descent. A more detailed description of CNNs can be found in [10].

We will use the MatConvNet toolbox to design and train the networks [11]. We will describe each network, illustrated in figure 1, and their input computation in more details below.

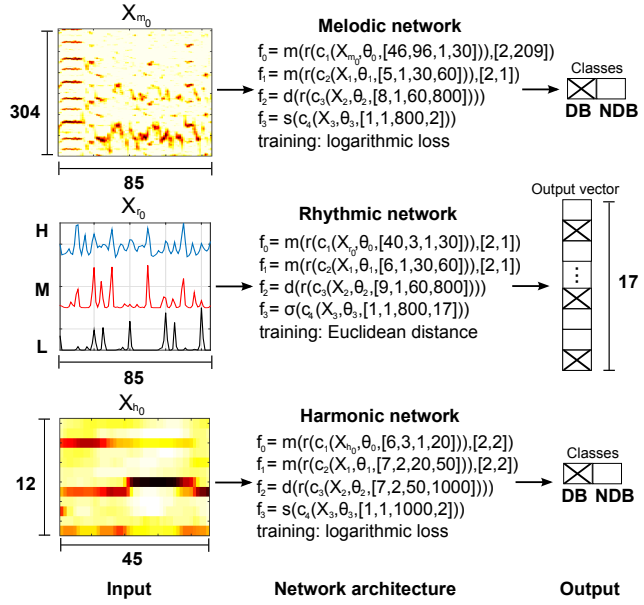

Fig. 1. Convolutional networks architecture, inputs and outputs. The notation is the same as in 3.1. DB and NDB stand for downbeat and no downbeat respectively.

3.2. Melodic neural network (MCNN)

Melodic lines often play around meter conventions and therefore a melody-related downbeat likelihood may not be very reliable by itself. However, it will provide complementary information that can be useful.

While experiments have been carried out to determine note accents in term of their relative position and duration [5], this is rather limited to a certain type of music and needs a good note extraction process. This is also very expensive and hard to do in practice for varied polyphonic audio music signals. We will follow the assumption that melody contour plays a role in perceiving rhythm hierarchies, but we will use a lower-level input representation than in [12] for example and then lead the network to learn higher level abstractions and use this cue to estimate the downbeat likelihood.

Input computation: We down-sample the audio signal at 11025 Hz and use a Hann analysis window of 185.8 ms, a hop size of 11.6 ms to compute the spectrogram via STFT. We then apply a constant-Q transform (CQT) with 96 bins per octave, starting from 196 Hz to the Nyquist frequency, and average the energy of each CQT bin $q[k]$ with the following octaves:

$$s[k] = \frac{\sum_{j=0}^{J_k} q[k + 96j]}{J_k + 1} \quad (3)$$

with J_k such as $q[k + 96J_k]$ is below the Nyquist frequency. We then only keep 304 bins from 392 Hz to 3520 Hz that corresponds to three octaves and two semitones. We tested averaging harmonics or integer multiple of a given frequency instead of octaves or power of 2 of this frequency, and the downbeat likelihood results were slightly better with the octave average. Besides, dependency to chroma input networks was similar in both case. With octave accumulation, melodic line replica, or ghost melodies, are equally spaced so it may be easier for the network to isolate a melodic line with an octave long window, especially at low frequency. While this feature might seem close to chroma, it is quite different as can be seen in figure 1. We are indeed starting at a relatively higher frequency, using many bins per octave and a 3 octave long representation that avoids circular shifting of the melody.

Then, we use a logarithmic representation of our function s:

$$ls = \log(\left|s_{[392Hz\ 3520Hz]}\right| + 1) \quad (4)$$

and we put every value that are below the third quartile Q_3 of a given temporal frame equal to zero to get our melodic feature mf:

$$mf = \min(ls - Q_3(ls), 0) \quad (5)$$

Keeping only the highest values allows us to remove most of the noise and the onsets so we can see some contrast and not be too close to rhythmic features. We interpolate the obtained representation in time to have 5 temporal units per tatum. Considering that we are looking for melodic patterns than can be relatively long, we will feed the network with inputs of 17 tatum length, centered on the tatum to classify.

Feature learning: We then have input features of frequency dimension of 304 and of temporal dimension of 17 times 5: $X_{m0} = [85, 304, 1]$. Our network architecture is presented in figure 1. For example, the first layer:

$$f_0 = m(r(c_1(X_{m0}, \theta_0, [46, 96, 1, 30])), [2, 209]) \quad (6)$$

means that we will use filters of size $[46, 96, 1, 30]$ for convolution, and will then use rectified linear units and max pooling with a reduction factor of $[2, 209]$ as non linearity. The first layer filters are relatively large so we are able to characterize melodic patterns. The

following max pooling will only keep the maximal convolution activation in the whole frequency range. This way, the network is constrained to keep the most linked melodic pattern to a downbeat position, regardless of the absolute pitch. The fourth layer can be seen as a fully connected layer that will map the preceding hidden units into 2 final outputs. Those outputs will represent the likelihood of the center of the input to be at a downbeat position and its complementary. The logarithmic loss to the ground truth is computed as the last layer to be able to train the network.

3.3. Rhythmic neural network (RCNN)

Rhythm patterns are often repeated every bars with possibly small variations over time. They also tend to be relatively stable compared to other musical components and can therefore be used to characterize the downbeat likelihood.

Input computation:

We compute a three bands spectral flux onset detection function (ODF) for that purpose. We compute the spectrogram via STFT using a Hann window of 23.2 ms and a hop size of 11.6 ms for a signal sampled at 44100 Hz. We use μ-law compression, with $\mu = 10^6$. We then sum the discrete temporal difference of the compressed signal on three bands for each temporal interval, subtract the local mean and keep only the positive part of the resulting signal. The frequency intervals of the low, medium and high frequency bands are [0 150], [150 500] and [500 11025] Hz respectively as we believe low frequency bands carry a lot of weight in our problem. It could represent low frequency, medium frequency and higher frequency percussive instruments. The signal is clipped so that all values on the 9^{th} decile are equal and the variation of this feature is reasonable. This new onset feature is a bit more robust to noise than the one in [4]. As before, we interpolate the obtained signal in time to have 5 temporal units per tatum. Since we want the network to be able to extract bar long patterns, we need to feed it with inputs longer than that. Besides, after listening tests, it became apparent that a 1 bar context is very limited to detect the downbeats with rhythm cues. We will then also feed the network with inputs of 17 tatum length, i.e $X_{r0} = [85, 3, 1]$.

Feature learning: We will try here to lead the network to learn length specific rhythmic patterns, instead of change around the downbeat position, that is not very indicative of a downbeat position as shown in the upper figure 2. For example, we would like the network to give different outputs if patterns of different length are observed. One way to give incentives in this direction is to do multi-label learning [13]. In that case, if there is a downbeat position at the first and ninth tatum of our 17 tatum-long input, the output of our network should be $o = [1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0]$. Since there might be multiple downbeats per input, we can't normalize the result with a softmax layer. Instead, we will first use a sigmoid activation unit as a penultimate layer to map the results into probabilities. We will then train the network with an Euclidean distance between the output and the ground truth with a similar shape as o so that each tatum are considered independent. Our network architecture is presented in figure 1. Our first convolutional layer also has relatively large filters. A qualitative analysis in the lower figure 2 shows that the network is therefore able to learn rhythm patterns. Besides, since we are using the Euclidean distance to ground truth vectors to train the network, we are not explicitly using classes such as downbeat and no downbeat. The output is then of dimension 17 and represent the downbeat likelihood of each tatum position in X_{r0}. Since we have 17 tatum-long inputs but a hop size of 1 tatum, overlap will occur. We will reduce the dimension of our downbeat likelihood to 1 by averaging the results corresponding to the same tatum, occurring at the right part of the input[2].

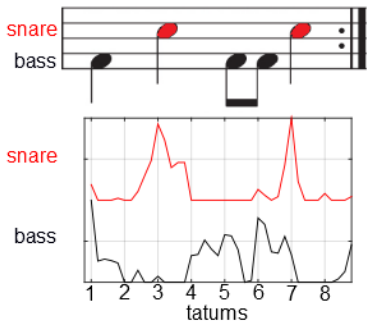

Fig. 2. Upper figure: One bar basic snare and bass drum pattern. Significant change in musical events does not appear specifically at the beginning of the bar. Lower part: Two bands of a first layer filter from the rhytmic network. The bands are normalized for clarity. Upper part: [150 500] Hz band. Lower figure: [0 150] Hz band. We can distinguish for the snare and kick drums a pattern similar to the one above.

3.4. Harmonic neural network (HCNN)

Harmonic content is very strongly connected to downbeats. Contrary to melody and rhythm, we are here mainly looking for change in this feature rather than specific patterns. Indeed, the exact label of a chord is less important for our task than the fact that it is likely to change around a downbeat position. This cue proves to be the most reliable one as far as western music is concerned.

Input computation: An efficient and robust way to model harmonic content in tonal music is to use chroma. We will do it as in [4] to obtain a standard 12 bins chromagram, also with 5 temporal units per tatum. Compared to the melodic feature, we keep 8 times less bins per octave (12 to 96). Indeed, we don't need the same precision to model the dominant harmony and the melodic lines. However, as for melody, we would like to be independent to the absolute pitch. Since chroma are circular, we will augment the training data with the 12 circular shifting combination of the chroma vectors. We will feed the network with 9 tatum-long inputs centered on the tatum to classify. They are relatively shorter than the other inputs since we are mostly looking for change, i.e $X_{h0} = [45, 12, 1]$.

Feature learning: Our network architecture is presented in figure 1. Since we don't need to learn long and specific chroma patterns, our first convolutional layer will feature filters of moderate size. The four layers of the network contain the same non linear functions as in the melodic network while the size of the filters and max pooling differs.

4. EVALUATION AND RESULTS

4.1. Methodology

We use the F-measure, computed with the evaluation toolbox in [14], to evaluate the performance of our system, as in [15–18]. This measure is the harmonic mean of precision and recall rates. We will use a tolerance window of +- 70 ms. We won't take into account the first 5 seconds and the last 3 seconds of audio as the annotations are sometimes missing and often not very reliable there.

[2]The network was indeed more efficient in finding the downbeat likelihood at the right part of the input.

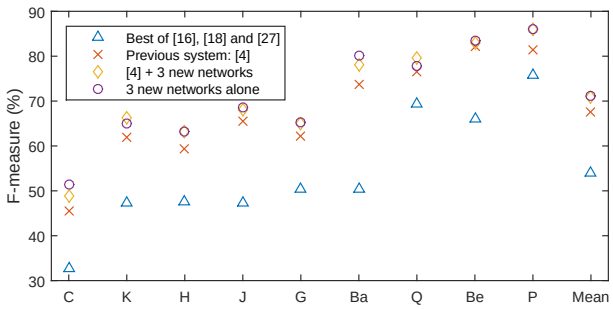

Fig. 3. F-measure results of 4 downbeat tracking systems on nine datasets and as a mean over datasets. C: RWC Classical [20], K: Klapuri 40 excerpts subset [21], H: Hainsworth [22], J: RWC Jazz [20], G: RWC Genre [23], Ba: Ballroom dances [24], Q: Quaero project [25], Be: Beatles collection [26], P: RWC Pop [20] and Mean: mean of the former results.

The evaluation will be carried out on 9 datasets, summarized in figure 3. We will use a leave-one-dataset-out approach, whereby in each of 9 iterations we use 8 datasets for training and validation, and the holdout dataset for testing. This evaluation method is more fair to non machine learning methods and is considered more robust [19]. 90% of the training datasets is used for training the network and the 10% is used to set the parameters value.

4.2. Results and discussion

Overall performance:

The performance of two configurations of our system compared to previous methods for each dataset and overall is shown in figure 3. For both configurations we use the framework presented in section 2. In the first case, denoted by the circles in figure 3, we are using only the 3 new networks. In the second case, denoted by the diamonds in figure 3, we are using the 6 networks in [4] and the 3 new networks. As for all the results presented here, the output of all networks is averaged to obtain the downbeat likelihood.

In each dataset, the F-measure is much higher for both configurations of our method compared to the ones of [16], [18] and [27], with an overall improvement of 17.1 percentage points (pp) when we only use the 3 new networks, from 54.1% to 71.2%. Compared to [4], results are between 3.4 and 3.7 pp higher depending on the configuration. We performed a Friedman's test and a Tukey's honestly significant criterion (HSD) test with a 95% confidence interval and the improvement of our new method is statistically significant in overall and for each individual dataset, except for the Klapuri subset and the RWC Jazz dataset. There is only 40 and 50 songs in those datasets and a statistically significant difference is therefore difficult to achieve.

We will then assess the effect of each new network compared to [4] through different configurations, numbered in the figure 4 and throughout the discussion to facilitate reference.

Rhythmic network performance:

To focus on the effect of our rhythmic network (RCNN), we computed the difference in F-measure between a system with the 6 networks in [4][3] plus the new rhythmic network and [4] (configuration 1). We then computed the difference in F-measure between [4] minus the old rhythmic network plus the new rhythmic network and [4] (configuration 2). We observe in both cases an increase in performance of about 1 pp that illustrates the added value of the new rhythmic network. Finally, to see if the multi-label learning was useful, we computed the difference in F-measure between [4] plus the new rhythmic network and [4] plus a variation of the new rhythmic network without the multi-label learning and trained with a logarithmic loss (configuration 3). Results are also positive with an increase by about 0.9 pp overall.

[3]referred in the following by [4] for concision

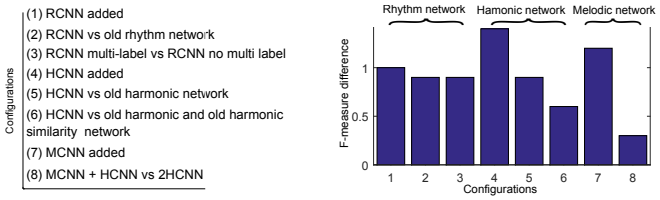

Fig. 4. F-measure difference for different configurations. See [4] for a description of the old networks.

Harmonic network performance:

We then focus on the effect of the harmonic network (HCNN). As before, the added value compared to [4] is +1.4 pp (configuration 4). We then computed the difference in F-measure between [4] minus the old harmonic network plus the new harmonic network and [4] (configuration 5), and also computed the difference in F-measure between [4] minus the old harmonic network and the old harmonic similarity network plus the new harmonic network and [4] (configuration 6). The F-measure still increases in both cases by 0.9 pp and 0.6 pp respectively. Indeed, a lot of information is shared with those 3 networks. They are based on the chroma feature and the old harmonic similarity network encodes chord invariance, that is taken into account by the data augmentation presented in subsection 3.4.

Melody network performance:

Finally, the added value of the melodic network compared to [4] is of 1.2 pp (configuration 7). Considering its design, we then assess if the melodic network may be seen as a degraded version of the harmonic network. While adding more weight to the harmonic network boosts the performance in all almost all cases, we computed the difference in F-measure between [4] plus the 3 new networks and [4] plus the new rhythmic network and two copies of the new harmonic network (configuration 8). We observe an increase in performance of 0.3 pp showing that using the melodic network still adds value compared to the new harmonic network.

Networks complementarity:

Each new network is then useful for our task. A surprising result is that using only the 3 new networks will lead to equivalent results as using the 9 new and old networks as can be seen in figure 3, illustrating the performance and complementarity of these new networks. Besides, since we are averaging the network outputs, low performance networks can get too much weight and high performance network such as the old harmony and harmony similarity networks can be too similar to the new harmonic network to add a lot of value.

5. CONCLUSION

We introduced three convolutional networks that take advantage of the specificity of a new melodic feature, an improved rhythmic feature and a harmonic feature for the task of downbeat tracking. Evaluation over various datasets showed that significant improvements were achieved by adding each new network to our past system and even by using the three new networks alone, therefore reducing the model complexity. It can be interesting in future work to look for an appropriate combination of the networks output and to integrate this powerful feature learning system into an adapted temporal model.

6. REFERENCES

[1] F. Krebs, A. Holzapfel, A. T. Cemgil, and G. Widmer, "Inferring metrical structure in music using particle filters," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 5, pp. 817–827, 2015.

[2] A. Holzapfel, F. Krebs, and A. Srinivasamurthy, "Tracking the "odd": Meter inference in a culturally diverse music corpus," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2014, pp. 425–430.

[3] A. Srinivasamurthy and X. Serra, "A supervised approach to hierarchical metrical cycle tracking from audio music recordings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 5217–5221.

[4] S. Durand, J. P Bello, B. David, and G. Richard, "Downbeat tracking with multiple features and deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 409–413.

[5] J. Thomassen, "Melodic accent: Experiments and a tentative model," *Journal of the Acoustical Society of America*, vol. 71, pp. 1596, 1982.

[6] P. Grosche and M. Müller, "Tempogram Toolbox: MATLAB tempo and pulse analysis of music recordings," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR), late breaking contribution*, 2011.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, et al., "On rectified linear units for speech processing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 3517–3521.

[9] G. Hinton, N. Srivastava, A. Krizhevsky, I. Suskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *The Computing Research Repository (CoRR)*, vol. abs/1207.0580, 2012.

[10] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 253–256.

[11] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," *CoRR*, vol. abs/1412.4564, 2014.

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

[13] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, pp. 1–13, 2007.

[14] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," *Queen Mary University, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.

[15] F. Krebs, F. Korzeniowski, M. Grachten, and G. Widmer, "Unsupervised learning and refinement of rhythmic patterns for beat and downbeat tracking," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2014.

[16] H. Papadopoulos and G. Peeters, "Joint estimation of chords and downbeats from an audio signal," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 138–152, 2011.

[17] M. Khadkevich, T. Fillon, G. Richard, and M. Omologo, "A probabilistic approach to simultaneous extraction of beats and downbeats," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 445–448.

[18] 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*, vol. 19, no. 6, 2011.

[19] A. Livshin and X. Rodet, "The importance of cross database evaluation in sound classification," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2003, pp. 241–242.

[20] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2002, vol. 2, pp. 287–288.

[21] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 342–355, 2006.

[22] S. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 2385–2395, 2004.

[23] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2003, vol. 3, pp. 229–230.

[24] "www.ballroomdancers.com," .

[25] "http://www.quaero.org/," .

[26] "http://isophonics.net/datasets," .

[27] M. E. P. Davies and M. D. Plumbley, "A spectral difference approach to extracting downbeats in musical audio," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2006.