CONVOLUTIONAL NEURAL NETWORKS FOR REAL-TIME BEAT TRACKING: A DANCING ROBOT APPLICATION

Aggelos Gkiokas
Institute for Language and Speech Processing,
Athena Research and Innovation Center,
Athens, Greece
agkiokas@ilsp.gr

Vassilis Katsouros
Institute for Language and Speech Processing,
Athena Research and Innovation Center,
Athens, Greece
vsk@ilsp.gr

ABSTRACT

In this paper a novel approach that adopts Convolutional Neural Networks (CNN) for the Beat Tracking task is proposed. The proposed architecture involves 2 convolutional layers with the CNN filter dimensions corresponding to time and band frequencies, in order to learn a Beat Activation Function (BAF) from a time-frequency representation. The output of each convolutional layer is computed only over the past values of the previous layer, to enable the computation of the BAF in an online fashion. The output of the CNN is post-processed by a dynamic programming algorithm in combination with a bank of resonators for calculating the salient rhythmic periodicities. The proposed method has been designed to be computational efficient in order to be embedded on a dancing NAO robot application, where the dance moves of the choreography are synchronized with the beat tracking output. The proposed system was submitted to the Signal Processing Cup Challenge 2017 and ranked among the top third algorithms.

1. INTRODUCTION

Rhythm analysis has been one the most important tasks in the Music Information Retrieval (MIR) community. The metrical structure of a music piece, comprising the metrical levels, their relations, the beats and the beat rate (i.e. tempo), fully describe its rhythm content as proposed in the Generative Theory of Tonal Music [1]. The exact beat locations are essential information of a music excerpt which describe various aspects of it, as for example tempo variations, expressive timing, temporal grouping of weak and strong accents within the meter etc.

Automated beat estimation, usually found under the term of Beat Tracking, has been one of the fundamental tasks in the MIR field. Beyond the importance of the information that can be found in the beat locations for a music excerpt as mentioned before, the automatic estimation of beats is important because it can serve as an intermediate step to tackle other MIR tasks, such as chord change detection [2], chord detection [3], the computation of beat-synchronous features for identifying cover songs [4], tempo estimation [5] and downbeat detection [6, 7] to name a few.

A variety of different approaches can be found in the literature to tackle the beat tracking task. In an early work, Scheirer [8] deployed comb resonators on spectral energies to for a tempo and beat estimation system, where the response of the resonators was interpreted as the beat positions. Dixon presented BeatRoot [9], a beat tracking software based on inter-onset interval (IOI) clustering for tempo induction and multiple agents for finding the beats. In [10], the authors extended BeatRoot to a real-time beat tracking software named IBT. In [11] a dynamic programming formulation for the beat-tracking task is presented. In a similar manner in [12] the output of a tempo estimation method was incorporated in a dynamic programming based beat-tracking method. In [13] the authors propose a two-state probabilistic model to handle discontinuities in beats caused by switching metrical levels. Peeters and Papadopoulos [14] propose a probabilistic framework, where beat positions are considered as latent variables in order to extract downbeats and beats. In [15] a unified probabilistic framework in the context of rhythm analysis is proposed, consisting of a time-invariant Bayesian network for modeling the relations of tactus, tatum and meter and the beat locations.

More recent works incorporated Neural Networks (NN) for handling the Beat-Tracking task. A remarkable work was firstly presented in [16], where an onset detection method which was based on Bidirectional Long Short-Term Memory (BLSTM) Neural Networks was adapted to a beat tracking system. The performance of this method outperformed the state-of-the-art. In [17], under the assumption that humans perceive the rhythm in a relative manner with respect to the salient periodicities of a music excerpt, a cepstroid invariant neural network is proposed to estimate the beat positions.

Although Convolutional Neural Networks (CNN) have been successful in many MIR applications such as onset detection [18], structure analysis [19], chord recognition [20] and genre classification [21, 22], to the best of our knowledge there are no works that deploy CNNs for the beat tracking task. CNNs have used recently for downbeat tracking as in [23] and [24]. However, these methods apply the CNN on larger segments of beat synchronous features, i.e. a beat-tracking step (based on another technique) is preceded. In a very recent approach [25] a dance

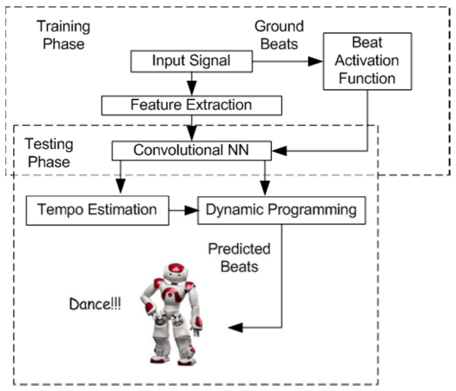

Figure 1. Overview of the proposed method.

genre classifier that is based on CNNs that model temporal features is proposed.

In this paper we propose a novel approach that adopts CNNs to handle the beat-tracking task. The CNN are used to learn a Beat Activation Function (BAF) from a time-frequency input representation. The resulting BAF is subsequently used to infer the beats with a dynamic programming approach. The proposed method was embedded on a dancing NAO[1] robot application. Dancing robots [26] have gained some interest in both robotics and MIR scientific communities and is an essential application for human-robot interaction and entertainment [27]. In [28] the authors focused on the motion of a humanoid dancing robot without any music rhythm analysis. In [29] a simple beat-tracking system was embedded on the RoboNova and Hubo robots. In [30] a singing robot that synchronizes its singing with the estimated beats is presented. In [31] the authors focused on eliminating the ego-motion and beat-synchronous noise caused by a real-time dancing robot.

The rest of the paper is organized as follows. Section 2 will provide a brief overview of the proposed method. The algorithmic details of the beat-tracking method will be presented in Section 3. Technical and implementation details of the dancing robot application are discussed in Section 4. Section 5 is dedicated on the evaluation of the proposed method. Section 6 concludes this paper with discussion and future work directions.

2. METHOD OVERVIEW

An overview of the proposed method is shown in Figure 1. From the input signal the time-varying features that capture the salient musical events are firstly computed. These features along with the Beat Activation Function derived from the ground truth data are used to train a Convolutional Neural Network (CNN). In the testing phase, the output of the CNN is interpreted as a prior probability of a time instant corresponding to a beat. The CNN output is processed further by a filter-bank of comb resonators to capture the salient periodicities and estimate the music tempo, which along with the BAF are processed by a Dynamic Programming Beat Estimation module. The aim of this module is to find a sequence of beats that are rhythmically consistent, and are dominant on the BAF. The beat tracking algorithm is embedded on a NAO robot. The robot reproduces an audio waveform, while in real-time it computes the beat locations. Consequently, the NAO robot synchronizes its dancing movements of the choreography to the predicted beats.

[1] https://www.ald.softbankrobotics.com/en/cool-robots/nao

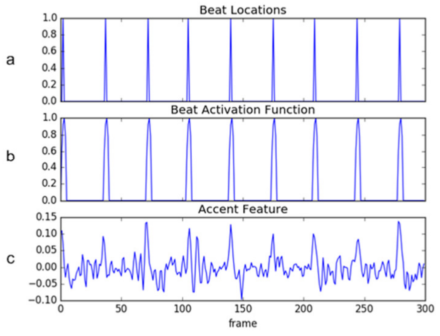

Figure 2. Beat Locations, Beat Activation Function and Accent Features.

3. METHOD DETAILS

3.1 Feature Extraction

A conventional front-end schema is used for feature extraction. The input music signal is firstly downsampled to 16 kHz. Downsampling is required in order to conform with NAO's audio recording capabilities, since NAO can only record at 16 and 48 kHz[2]. At next, the amplitude spectrogram denoted by **X** is calculated, using a sliding window of 1024 samples shifted at 160 samples (100 Hz frame rate). Each frame is then processed by a mel-filterbank of M bands to derive the mel-band amplitudes **S.** Next, the time difference of the logarithm of the amplitudes **S** of consecutive frames is computed. Finally, the output features are half-wave rectified to derive the *accent features* **A**.

3.2 Calculating a Beat Activation Function

A Beat Activation Function (BAF) is a function of time that represents the salience or the probability of time instants being beats. Figure 2 (a) shows the beat locations of a music excerpt, Figure 2 (b) shows the corresponding BAF, while in Figure 2 (c) the input features of the corresponding music excerpt are shown. BAF is a smoother version of pulses at the beat locations and is derived by using a Gaussian curve around the beat locations. The aim of the Gaussian blurring is to decrease the sensitivity to less accurate beat annotations. The standard deviation

[2] Although the current implementation does not use the microphone, the use of the recording sampling rate was chosen for reasons of uniformity.

of each Gaussian is set to be inversely proportional to the beat period.

In the proposed method a Convolutional Neural Network is deployed to learn the BAF denoted by $b[n]$ from the accent features $a[n]$. The intuition for this choice is that a CNN can learn temporal patterns of the input feature space and map these features to the BAF space. An example of how a simple Convolutional Unit can learn a rhythmic pattern and outputs a BAF is shown in Figure 3. Figure 3a shows a pattern r_1 of length L and Figure 3b shows a rhythmic repetitive pattern of r_1 , $r = [r_1, r_1, r_1, \dots r_1]$. We assume that beat positions should occur on the positive peak the r_1. If r is convolved with $\tilde{r}_1[n] = r_1[L-n]$ the resulting vector will exhibit peaks with the same period as r, but in different positions (Figure 3c). However, if r (Figure 3e) is convolved with filter $\tilde{r}_2$ (Figure 3d) which is a circular shift of r_1, then the resulting output (Figure 3f) is in phase, i.e. exhibits peaks at the same locations, with r. This simple example indicates that the CNNs are capable of producing a BAF that is synchronous with its input in a *causal* manner. If a sequence a of length N is convolved with a filter h of length L, the resulting sequence of length N-L+1 can be *synchronous* with a target BAF cropped by L-1 samples from the beginning. In this way, we can compute a BAF in a *causal* manner, i.e. $b[n]$ is derived only from current and past samples of r and r_1:

$$b[n] = \sum_{i=0}^{L-1} \tilde{h}[n-i]a[n-i] \tag{1}$$

where $\tilde{h}[n] = h[L-n]$.

The CNN architecture of the proposed method is shown in Figure 4. It consists of two convolutional layers and three dense layers. No max-pooling or other pooling step is applied anywhere, since this would reduce the frame rate, which is critical in the context of real-time beat tracking. The first convolutional layer consisting of N_1 filters of L_1 length is applied on the accent features **A**. This results to temporal sequences of dimension $N_1 \times M$. Next, the dimension is reduced to dimensions of M and 1 by applying the $N_1 \times 1$ and $M \times 1$ input-output feed forward layers respectively. As a non-linearity function of the CNN, the logistic function is applied to the output of each layer. The above process results to a 1-dimensional temporal sequence of length $N - L_1 + 1$ where N is the length of the input accent features **A**. The output $o_1[n]$ of this subnet is further processed by the 2nd convolutional layer with N_2 filters of L_2 length. The resulting temporal $o_2[n,m]$ sequence of dimension N_2 is then reduced to a single-dimensional sequence $o[n]$ by applying a $N_2 \times 1$ feed forward layer. As before, the logistic function is deployed at each layer. $o[n]$ is considered as the output of the network. For training the net work, the binary cross-entropy is used as the cost function.

After the calculation of the BAF, the beat tracking problem can be seen as a peak selection step, which comprises of two steps. At first a dominant tempo is estimated and then the most salient peaks of the BAF that are rhythmically consistent with that tempo are selected. These two steps are presented in detail in the Sections 3.3-3.5.

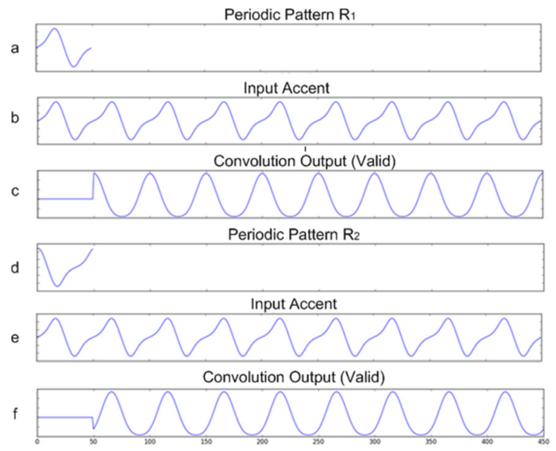

Figure 3. Convolution of periodic sequences with its rhythm elements.

3.3 Tempo Estimation

The next important component of the beat-tracking method is the estimation of the music tempo. A central notion on rhythm analysis is the Periodicity Function (PF) [32] or Periodicity Vector, which is a function or vector that represents the salience of the rhythmic frequencies. In this approach, we compute a PF by processing the BAF by a bank of oscillators, each of which oscillates at a period τ. The output $o_\tau[n]$ of the oscillator with period τ and input $a[n]$ is chosen as:

$$o_\tau[n] = \beta \cdot o_\tau[n-\tau] + (1-\beta) \cdot a[n] \, . \tag{2}$$

The PF for period τ at the frame n is given by the maximum value of o_τ within the past period, i.e.,

$$P[\tau][n] = \max \left\{ o_\tau[k], \quad k = n-\tau \dots n \right\} . \tag{3}$$

The PF is not calculated for every frame but every T_N frames and for periods in the range $\tau \in [\tau_{min}, \tau_{max}]$. In order to estimate a more reliable PF and cope with slight PF variations, the PFs are averaged for the last K values to get an smoothed PF $\tilde{P}$. The tempo period is then calculated as the maximum value of $\tilde{P}$.

3.4 Beat Tracking as Dynamic Programming

The Beat Estimation method is a modification and adaptation of the method presented in [12]. It is a dynamic programming method that finds an optimal path of a beat sequence that maximizes a cost function. Let $\{b_i\}_i$ denote the candidate beat positions, which are the positive peaks of the BAF. The first part of the beat tracking algorithm is to define a beat similarity of two candidate beats $b_i < b_j$ as

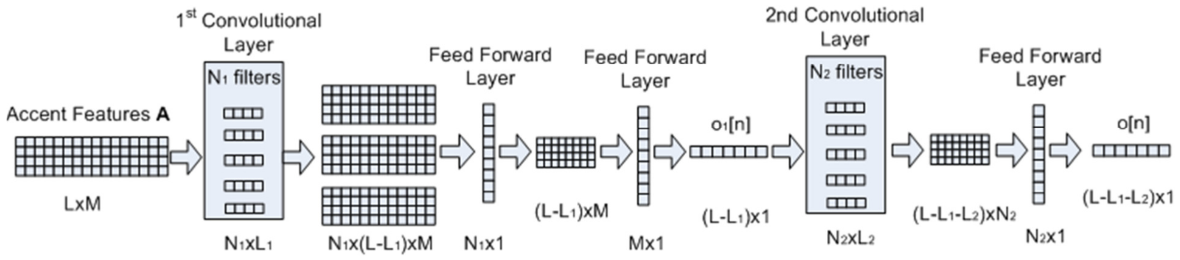

Figure 4. The Convolution Neural Network Architecture.

$$d(b_i,b_j) = c \cdot d_T(b_i,b_j) + (1-c)o[b_j] + d_0, \quad (4)$$

where

$$d_T(b_i,b_j) = \exp\{-\frac{1}{\sigma^2}\ln^2\left((b_j-b_i)/\tau\right)\}, \quad (5)$$

with τ in (5) being the tempo period and $d(b_i,b_j)$ being the weighted sum of two terms plus the bias d_0. The first term in (4) $d_T(b_i,b_j)$ indicates if the beats b_i,b_j are rhythmically consistent, and increases as the distance of b_i,b_j ap proaches the tempo period τ. The second term is related to the beat salience, and is equal to the value of the BAF. Finally, the term d_0 can be seen as a *beat insertion* bias. Given a sequence $\{b_i\}_i$ of possible beats, a beat activation function $o[n]$ and a tempo period τ one can find the optimal path $\{b^*{}_i\}$ that maximizes the objective function

$$O(\{b_l^*, l \in L\}) = \sum_{l\in L} d(b_{l-1}^*, b_l^*) \quad (6)$$

3.5 Real-Time Formulation of Beat Tracking

The beat tracking model described in the previous section is not a straightforward online algorithm and has to be adapted to meet the real-time requirements. For setting the real-time formulation, we define four frame sets. Let us denote with n_0 the current frame. Then we define by $F=\{b_i\}_i$ the beat candidates (peaks of BAF) before n_0, by $B=\{b_i{}^*\}_{n_0}$ the optimal path of beats, by $R=\{r_i{}^*\}_{n_0}$ the output of the real-time beat tracking before n_0 and by $E=\{e_i\}_{n_0}$ the expected beats of the n_0 frame. Note that $\{b_i{}^*\}_{n_0} \neq \{r_i{}^*\}_{n_0}$, since B can be considered as the non-causal output before n_0 , while the set R is the causal output before n_0. A graphical illustration is shown in Figure 5. When a new peak at frame n_1 is detected, then the hypotheses $\{b_i{}^*\}_{n_1}, \{r_i{}^*\}_{n_1}$, and $\{e_i\}_{n_1}$ are updated as follows. Firstly, the optimal path of beats is recalculated $\{b_i{}^*\}_{n_1}$. Then, $E=\{e_i\}_{n_1}$ is updated by adding a new expected beat $\tilde{e}$, which is the last beat $\tilde{b}$ of $B=\{b_i{}^*\}_{n_1}$ plus the tempo period τ. Moreover, the expected beats close to $\tilde{e}$ (based to a threshold) are removed. Next, if the current peak n_1 is close enough (based to the same threshold) to an expected beat (added to $E=\{e_i\}_{n_1}$ on a past frame), the n_1 is considered as a beat, it is added to $R=\{r_i{}^*\}_{n_1}$ and the corresponding expected beat is removed from $E=\{e_i\}_{n_1}$. In other words, a peak is instantly classified as a beat, if it is close enough to an expected

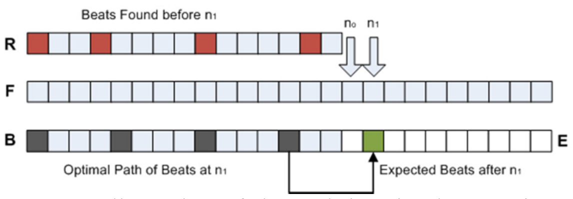

Figure 5. Illustration of the real-time implementation of the beat-tracking method

beat. Based on this formulation, the algorithm is almost online, with a latency of one frame, which is needed to decide whether a frame is peak or not. The algorithm can be summarized as follows:

```
1)  Initialize: $F=\{\}, E=\{\}, B=\{\}, R=\{\}$
2)  Get new frame $n$.
3)  Compute the BAF $o[n]$
4)  If $n$ is not peak GOTO 2.
5)  Compute distance of $n$ to previous peaks.
6)  Get optimal beat sequence $B=\{b_i{}^*\}_n$ before $n$.
7)  Add $\tilde{e}=\tilde{b}+\tau$ ($\tilde{b}$ is last beat of $\{b_i{}^*\}_n$) to $E$.
8)  Remove elements of $E$ very close to $\tilde{e}$.
9)  Get last element $\tilde{r}$ of $R$ (if any).
10) For each $e$ in $E$:
        a.  if ($n$ close to $e$) and ($n-\tilde{r}>\tau/2$)
                i.  add $e$ to $R$.
               ii.  remove $e$ from $E$.
              iii.  This_frame_is_beat = True
11) GOTO 2.
```

The statement x close to y is defined to be true if $|x-y|<4$ frames. The second condition $n-\tilde{r}>\tau/2$ in the statement 10a ensures that once a past frame is classified as beat, the frames that are close to this frame should not be classified as beats.

4. THE DANCING ROBOT APPLICATION

4.1 The NAO Robot

For deploying the real-time beat tracking method to a dancing robot, a NAO Robot v4 was used as the target hardware. It runs on an Intel Atom Z530 CPU with 2 cores at 1600Mhz, with 1GB RAM. The Operating System is the Linux based NAOqi, version 2.1.2. The NAO's kinematics include 25 motors; 2 motors for controlling the head, 4 motors for each arm, 2 motors for each hand, 5 motors for each leg, plus 1 motor for controlling both hip's yaw pitch (Figure 6). The pose (or “state”) of the robot can be uniquely described by the state of the 25 motors, which are used to define NAO's dancing movements that are synchronous with the real-time beat tracking re-

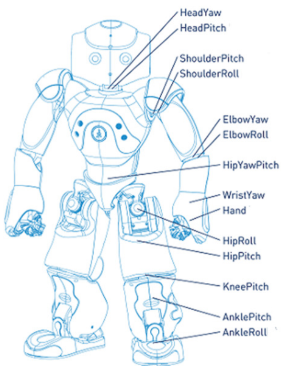

Figure 6. NAO's motor map.

sults. Although NAOqi has built-in functions for controlling the robot's stability, steep movements should be still avoided. Apart from the robot's movements, NAO's eyes are used to show the beat-times that are calculated in real-time. Each eye consists of 8 leds. The color of NAO's eyes is controlled to change on every beat.

4.2 The Choreography Model

NAO's dancing movements have been designed to be easily parameterized. We refer to this parameterization as the "Choreography Model (CM)". The CM consists of a set of poses $p = \{p_1, p_2, \ldots, p_K\}$ and a $K \times K$ transition matrix **T** of these poses. Each pose p_i is represented by the 25 values that correspond to the state of each of the motors mentioned above. The (i, j) element of matrix **T** denotes the probability of moving to pose p_j given that NAO's previous pose is p_i. Thus, each row of **T** sums to one. In this way one can define different choreographies. The elements of **T** can be binary values, thus defining a deterministic sequence of positions, or can take values in the interval (0,1) defining a stochastic sequence, or mixed, i.e. certain groups of deterministic sequences can be mixed stochastically. The CM allows an easy way to define an arbitrary choreography, mixing different choreographies (i.e. reusing p_i) and change the choreography in real-time, as for example in the case where the choreography is linked to a genre classifier. Although the CM poses contain information for the legs, for reasons of simplicity and to maintain the robot's stability, the movement of the legs of the robot are overridden by another simpler CM that considers only leg movements and involves only two poses, which realize the slight hip (or knee) movement of the robot that can be seen in the demonstration videos.

4.3 Robot Dancing in Practice

Since the robot cannot instantly move to the next pose when a beat is found, somehow the next beat-time has to be inferred beforehand. When a beat is found in real-time, then the next beat time is inferred by adding to the current beat time the tempo period (similarly to the expected beat notion of previous Section). Therefore, the predicted next beat that determines the robot's movement, slightly differs from the actual prediction of the real-time beat tracker. However, although such differences might be audible, they are not visible, i.e. they are not evident by human vision, as it can be seen in the demonstration video. Let's consider the case that a beat is found by the real-time algorithm at time instant 0.0 sec with a tempo at 60 BPM (i.e. period of 1 sec). Then the next beat will be inferred to be at 1.0 secs, and therefore the robot will start moving from its current pose to the next at 1.0 sec of a specific choreography. If the beat of the actual real-time algorithm is found a few msecs later of the inferred one (e.g. 50 msecs), this difference will not be visible. Moreover, the robot will adapt its movement in order to complete its next move at 2.10 secs. This can be seen in the demonstration video, since robot's movements look natural and synchronous with the music despite the small variations from the true beat.

However, in order to demonstrate the actual capability of the algorithm and make it clearly visible, apart from dance movement we incorporated an instant change in the color of the eyes when a beat is found. The colors used can also be parameterized as it is shown for the music excerpts in the demonstration video. Moreover, due to motor limitations of the robot, the robot dances on half time, i.e. the movement is planned for every second subsequent beat. Thus, every two beats (two color changes of the eyes) the robot completes one movement of the choreography.

5. EVALUATION

5.1 Algorithm Parameters and Implementation Details

In this section some details of the implementation and parameters of the proposed method will be provided. Regarding the accent features (Eq. 1), the number of bands of the mel-filterbank was set to M=8, and the frame rate was set to 100 Hz (see Section 3.1). For the tempo estimation phase, the corresponding periods of the tempo analysis range (Eq. 4) were set to $\tau_{min} = 350, \tau_{max} = 700$ ms. The size of the CNN is set to $N_1 = 50$, $N_2 = 100$, $L_1 = 50$ and $L_2 = 200$. The whole implementation is written in Python 2.7. The training functions were written using the Lasagne/Theano[1] libraries and run on a Tesla K40 GPU. The Binary Cross Entropy was used as the loss function of the network, which was trained using Nesterov momentum of 0.9 and a variable learning rate. Due to the robot's software limitations, for the embedded algorithm only the Numpy library was used for calculating dot products. The algorithm runs at ~ 100% CPU single core on the NAO robot and ~3 % CPU single core on an Intel i7.

5.2 Beat-Tracking Performance

The proposed method was submitted to the Real-Time Beat Tracking Challenge of the IEEE Signal Processing

[1] http://lasagne.readthedocs.io/en/latest/

Cup 2017 [33]. It was a prerequisite that the submissions should have been implemented around creative application of a real-time beat tracking algorithm that is embedded on a device. The challenge was run in three phases. Initially, the teams were provided two datasets, an *open* dataset consisting of 25 excerpts that were annotated by the organizers, and a *closed* dataset, where the annotations were not released to the participating teams. In the first phase each team had to submit annotations for three annotated excerpts of their choice (other than the *open* and *closed* datasets). One of these three pieces was selected by each team as a *challenge* piece. The challenge pieces for all teams formed the *challenge* dataset, and the dataset consisting of the other two pieces submitted by each team, formed the *team* datasets. In the second phase each team had to submit the results of their algorithm for the *closed* and *challenge* datasets, while they were provided annotation for the *team* dataset to be used for training or fine tuning. The submissions were evaluated based on three criteria, the annotation quality, the beat-tracking performance and the novelty of the application. The beat-tracking performance was measured by a metric based upon the standard AMLt, but instead of the arbitrary inclusion of double, half-time or off-beat tapping, the "allowed" metrical levels were specified on an excerpt-by-excerpt basis. In this way, the evaluation could cope with excerpts in odd meters in a more robust manner. After the 2nd phase, the three best teams were selected to participate in the final phase of the competition at the ICASSP 2017 conference.

The Convolutional Neural Network of the proposed method was trained on three datasets, the GTZAN dataset [34], the Ballroom Dataset [35] and the SMC Mirex Dataset [36]. The parameters of Eq. (4), and (5) were tuned with a grid search on the *open* and the *team* submitted datasets which were used as validation set. The proposed method achieved a beat-tracking performance of 63.3% and 64.2% on the *closed* and *challenge* datasets respectively, based on the modified AMLt metric and was ranked 6th among 21 submissions regarding the beat-tracking performance. Figure 7 presents the comparative beat tracking results for all submissions. Due to the organizers choice, the comparative results are provided anonymously. The video demonstration of the robot dancing for two excerpts can be downloaded from[1].

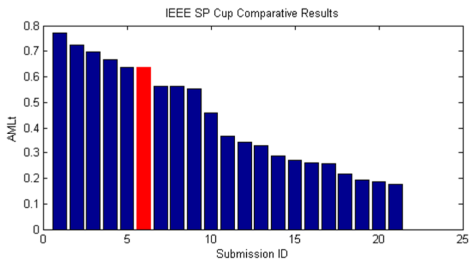

Figure 7. Comparative results on the IEEE SP Cup Challenge. The submissions are ordered by their beat tracking performance.

6. CONCLUSION AND FUTURE WORK

In this paper a novel method that adopts CNNs for the beat-tracking problem is presented. CNN are proved to be capable to function in an online manner, i.e. the output of the CNN depends only on current and past values of its input, thus allowing a real-time implementation. The proposed method was designed to be embedded on a NAO robot, and its parameters were optimized to meet the real-time requirements rather than to optimize beat-tracking performance in general. The proposed method was submitted in the IEEE 2017 Signal Processing Cup Beat Tracking Challenge and was ranked 6th among 21 algorithms.

There are a number of challenges and future work imposed by the proposed method. Regarding the use of CNNs, a further investigation and experimentation of various network parameters, such as the network architecture, the number and size of layers, the use of other than the sigmoid non-linearity functions, more relevant to the beat-tracking problem cost functions, or even the smoothing procedure of the target BAF. Moreover, CNNs can be combined with other Neural Network types such as Recurrent Neural Networks, as for example in a setting that CNN will act as a feature preprocessing step to extract a smooth BAF, which will be further processed by an RNN.

Apart from the CNN, other aspects can be further elaborated to increase the performance of the beat-tracking algorithm. At first, the tempo estimation method may be improved, by considering more complex oscillators or other alternative Periodicity Analysis methods that can be found in the literature. Moreover, further processing of the PF can be incorporated to the method, reducing the so called “octave errors”. Regarding the beat tracking, it can be further improved to more sophisticated approaches, as for example with the use of agents [10] allowing smarter selection from the candidate beats.

Regarding the robot itself, we plan to incorporate a genre classifier. This will allow the robot to change choreographies on the fly, with respect to the music being played. Finally, it will be an important extension of the proposed method to handle audio streams recorded from the robot’s microphones instead of audio files. This would require either the training of the CNN to be made with data recorded from microphone, or by deploying denoising techniques.

7. ACKNOWLEDGEMENTS

This work is part of the iMuSciCA project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731861. Students Myrto Damianou, Stelios Stavroulakis, Giannis Karamanolakis and Christos Palivos, participated in the SP Cup team. Special thanks

[1] http://mir.ilsp.gr/dance_robot.html

to Matthew Davies for providing the evaluation SP Cup results and for his helpful feedback.

8. REFERENCES

[1] F. Lerdahl, and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT press, 1985.

[2] Goto, Masataka, and Yoichi Muraoka. "Real-time Beat Tracking for Drumless Audio Signals: Chord Change Detection for Musical Decisions." *Speech Communication* 27, no. 3 (1999): 311-335.

[3] Zenz V, Rauber A. "Automatic Chord Detection Incorporating Beat and Key Detection," *in Proc. ICSPC* 2007.

[4] Ellis DP, Poliner GE,, "Identifying Cover Songs with Chroma Features and Dynamic Programming Beat Tracking," *in Proc. ICASSP,* 2007.

[5] Böck S, Krebs F, Widmer G., "Accurate Tempo Estimation Based on Recurrent Neural Networks and Resonating Comb Filters," in *Proc. ISMIR,* 2015.

[6] Durand S, Bello JP, David B, Richard G. Feature Adapted Convolutional Neural Networks for Downbeat Tracking," in *Proc. ICASSP,* 2016.

[7] Krebs F, Böck S, Dorfer M, Widmer G., "Downbeat Tracking using Beat-Synchronous Features and Recurrent Neural Networks," *in Proc. ISMIR,* 2016.

[8] E. D. Scheirer, "Tempo and Beat Analysis of Acoustic Musical Signals," *J. Acoust. Soc. Amer*., 103(1), pp. 588–601, 1998.

[9] Dixon S., "Evaluation of the Audio Beat Tracking System Beatroot," *Journal of New Music Research.* 2007 Mar 1;36(1):39-50.

[10] Oliveira JL, Davies ME, Gouyon F, Reis LP., "Beat Tracking for Multiple Applications: A Multi-Agent System Architecture with State Recovery," *IEEE Trans. Audio, Speech, Lang. Process.*, 20(10), pp. 2696-2706, Dec. 2012.

[11] D.P.W Ellis, "Beat Tracking by Dynamic Programming." *Journal of New Music Research*, 36(1), pp. 51-60, 2007.

[12] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis, "Music Tempo Estimation and Beat Tracking by Applying Source Separation and Metrical Relations," in *Proc. ICASSP*, 2012.

[13] M. E. P. Davies and M. D. Plumbley, "Context-Dependent Beat Tracking of Musical Audio," *IEEE Trans. Audio, Speech, Lang. Process.*, 15(3), pp. 1009–1020, Mar. 2007.

[14] G. Peeters and H. Papadopoulos, "Simultaneous Beat and Downbeat Tracking Using a Probabilistic Framework: Theory and Large-Scale Evaluation," *IEEE Trans. Audio, Speech, Lang. Process.*, 19(6), pp. 1754–1769, Aug. 2011.

[15] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the Meter of Acoustic Musical Signals*," IEEE Trans. Audio, Speech, Lang. Process.*, 14(1), pp. 342–355, Jan. 2006.

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

[17] Elowsson A. "Beat Tracking with a Cepstroid Invariant Neural Network," *in Proc. ISMIR*, 2016.

[18] Schluter J, Bock S., "Improved Musical Onset Detection with Convolutional Neural Networks," *in Proc. ICASSP*, 2014.

[19] Ullrich K, Schlüter J, Grill T., "Boundary Detection in Music Structure Analysis using Convolutional Neural Networks," *in Proc. ISMIR*, 2014.

[20] Humphrey EJ, Bello JP. Rethinking Automatic Chord Recognition with Convolutional Neural Networks," *in Proc. ICMLA*, 2012.

[21] Li, T.L., Chan, A.B. and Chun, A., "Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network," in *Proc. Int. Conf. Data Mining and Applications*, 2010.

[22] Dieleman, S., Brakel, P. and Schrauwen, B., "Audio-Based Music Classification with a Pretrained Convolutional Network," in *Proc. ISMIR*, 2011.

[23] Durand S, Bello JP, David B, Richard G., "Robust Downbeat Tracking Using an Ensemble of Convolutional Networks," in *IEEE/ACM Trans. Audio, Speech, Lang. Process.,* 25(1), pp. 76-89, Jan. 2017.

[24] Krebs F, Böck S, Dorfer M, Widmer G., "Downbeat Tracking using Beat-Synchronous Features and Recurrent Neural Networks," in *Proc. ISMIR*, 2016.

[25] Pons J., and Serra X., "Designing Efficient Architectures for Modeling Temporal Features with Convolutional Neural Networks," *in Proc. ICASSP*, 2017.

[26] Aucouturier, J.J., Ikeuchi, K., Hirukawa, H., Nakaoka, S.I., Shiratori, T., Kudoh, S., Kanehiro, F., Ogata, T., Kozima, H., Okuno, H.G. and Michalowski, M.P., 2008. Cheek to Chip: Dancing Robots and AI's Future. *IEEE Intelligent Systems*, *23*(2).

[27] Michalowski, M.P., Sabanovic, S. and Kozima, H., "A Dancing Robot for Rhythmic Social Interaction," in Proc. HRI, 2007.

[28] Shinozaki, K., Iwatani, A. and Nakatsu, R., "Concept and Construction of a Robot Dance System," in *Proc. International Workshop and Conference on Photonics and Nanotechnology,* 2007.

[29] Grunberg, D., Ellenberg, R., Kim, I.H., Oh, J.H., Oh, P.Y. and Kim, Y.E., 2010. Development of an Autonomous Dancing Robot. *International Journal of Hybrid Information Technology*, *3*(2), pp.33-44.

[30] Murata, K., Nakadai, K., Yoshii, K., Takeda, R., Torii, T., Okuno, H.G., Hasegawa, Y. and Tsujino, H., "A Robot Singer with Music Recognition Based on Real-Time Beat Tracking." in Proc. *ISMIR,* 2008.

[31] Oliveira, J.L., Ince, G., Nakamura, K., Nakadai, K., Okuno, H.G., Gouyon, F. and Reis, L.P., "Beat Tracking for Interactive Dancing Robots," in *International Journal of Humanoid Robotics*, *12*(04), p.1550023.

[32] Gkiokas A., Katsouros V., and Carayannis G, "Towards Multi-Purpose Spectral Rhythm Features. An Application to Dance Style, Meter and Tempo,"

IEEE/ACM Trans. Audio, Speech, Lang. Process., 24(11), April 2016, pp. 1885-1896.

[33] C. Jin, M. E. P. Davies and P. Campisi. "Embedded Systems Feel the Beat in New Orleans: Highlights from the IEEE Signal Processing Cup 2017 Student Competition," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 143-170, July 2017

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

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

[36] Holzapfel A., Davies M. E. P., Zapata J. R., Oliveira J., and Gouyon F. "Selective Sampling for Beat Tracking Evaluation", *IEEE Trans. Audio, Speech, Lang. Process.*, 20(9), pp.2539-2548, 2012.