A SINGLE-STEP APPROACH TO MUSICAL TEMPO ESTIMATION USING A CONVOLUTIONAL NEURAL NETWORK

Hendrik Schreiber
tagtraum industries incorporated
hs@tagtraum.com

Meinard Müller
International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de

ABSTRACT

We present a single-step musical tempo estimation system based solely on a convolutional neural network (CNN). Contrary to existing systems, which typically first identify onsets or beats and then derive a tempo, our system estimates the tempo directly from a conventional mel-spectrogram in a single step. This is achieved by framing tempo estimation as a multi-class classification problem using a network architecture that is inspired by conventional approaches. The system's CNN has been trained with the union of three datasets covering a large variety of genres and tempi using problem-specific data augmentation techniques. Two of the three ground-truths are novel and will be released for research purposes. As input the system requires only 11.9 s of audio and is therefore suitable for local as well as global tempo estimation. When used as a global estimator, it performs as well as or better than other state-of-the-art algorithms. Especially the exact estimation of tempo without tempo octave confusion is significantly improved. As local estimator it can be used to identify and visualize tempo drift in musical performances.

1. INTRODUCTION

Undoubtedly, the *tempo* of a musical piece is one of its main characteristics. Its estimation is often defined as measuring the frequency with which humans "tap" along to the beat. This is notably different from *beat tracking*, which aims at determining individual beat positions. If the tempo of a musical piece stays constant throughout the whole performance, it is called *global tempo*. It can be represented by a single number usually specified in *beats per minute* (BPM). Global tempi often occur in genres like Rock, Pop, and Dance music. The method proposed in this paper was primarily developed for estimating the tempo of short excerpts, but can also be applied to global tempo estimation.

Many different approaches to tempo estimation have been taken in the past. Gouyon et al. [11] provided a comparative evaluation of the systems that participated in the ISMIR 2004 contest, the first large-scale evaluation of tempo induction algorithms. Five years later, Zapata and Gómez gave an updated overview [39]. To our knowledge, the most recent comprehensive evaluations are presented in [2, 25, 31]. For a textbook-style introduction see [22].

Early tempo estimation methods often combined signal processing with heuristics. Scheirer [28] for example used bandpass filters, followed by parallel comb filters, followed by peak picking. Klapuri et al. [17] replaced the conventional bandpass approach with STFTs, producing 36 band spectra. By differentiating and then half-wave rectifying the power in each band, they created band-specific onset strength signals (OSS), which were then combined into four accent signals and fed into comb filters in order to detect periodicities. Instead of processing an OSS with comb filters, several other methods have been proposed. Among them autocorrelation [1, 22], clustering of inter-onset intervals (IOI) [5, 33], and the discrete Fourier transform (DFT) [22, 23].

Recent approaches put emphasis on finding not just *a* periodicity, but on finding one corresponding to the perceived tempo, trying to avoid common errors by a factor of 2 or 3, so-called *octave errors* [11, 31]. The methods used range from genre classification (e.g., obtained by a genre classification component) [14, 32], secondary tempo estimation [30], and the discrete cosine transform of IOI histograms [7], to machine learning approaches like Gaussian mixture models (GMM) [24], support vector machines (SVM) [9, 25], k-nearest neighbor classification (k-NNC) [37, 38], neural networks [6], and random forests [31].

Another area of active research aims at creating a better OSS through the use of neural networks. Elowsson [6] uses harmonic/percussive source separation and two different feedforward neural networks to classify a frame as beat or non-beat. Böck et al. [2] use a bidirectional long short-term memory (BLSTM) recurrent neural network (RNN) to map spectral magnitude frames and their first order differences to beat activation values. These are then processed further with comb filters. For their dancing robot application, Gkiokas et al. [10] use a convolutional neural network (CNN) to derive a beat activation function, which is then used for beat tracking and tempo estimation.

What all these methods have in common is the *multi-step* approach of decomposing the signal into sub-bands, deriving some kind of OSS, detecting periodicities, and then trying to pick the best one. As Humphrey et al. [15] point out, this can be described as a deep architecture con-

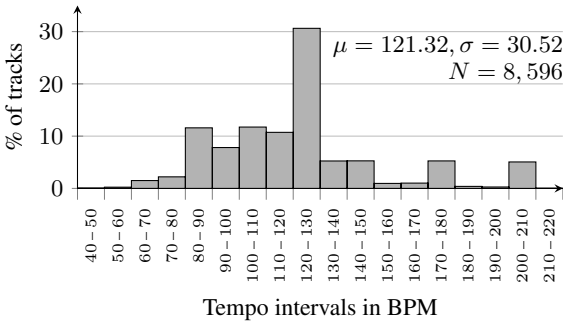


Figure 1: Tempo distribution for the `Train` dataset consisting of `LMD Tempo`, `MTG Tempo`, and `EBall`.

sisting of multiple components ("layers") that has evolved naturally. But to the best of our knowledge, nobody has replaced the traditional multi-component architecture with a single deep neural network (DNN) yet. In this paper we describe a CNN-based approach that estimates the local tempo of a short musical piece or excerpt based on mel-scaled spectrograms in a single step, i.e., without explicitly creating mid-level features like an OSS or a beat activation function that need to be processed further by another, separate system component. Using averaging, we can combine multiple local tempi into a global tempo.

The remainder of this paper is structured as follows: Section 2 introduces our training datasets. Then Section 3 describes the signal representation, network architecture, network training, and how we combine multiple local estimates into a global estimate. In Section 4 we evaluate our global tempo estimation approach quantitatively by benchmarking against known datasets and state-of-the-art algorithms. Then we discuss local tempo estimation qualitatively using samples from different genres and eras. Finally, in Section 5 we present our conclusions.

2. TRAINING DATASETS

Our goal is to create a general purpose system that does not suffer from strong genre-bias. Therefore we avoid cross-validation on small datasets and instead created a large, multi-genre training dataset, consisting of three smaller datasets: One derived from a subset of the Lakh MIDI dataset (`LMD`) [27], a subset of the GiantSteps MTG key dataset (`MTG Key`) [8] [1], and a subset of the `Extended Ballroom` [20] dataset. Two of the derived ground-truths have been newly created for this paper.

2.1 LMD Tempo

`LMD` is a dataset containing MIDI files that have been matched to 30 s audio excerpts. While some of the MIDI files contain tempo information, none of the audio files are annotated, and there is no guarantee that associated MIDI and audio files have the same tempo. Our idea is to create a sub-dataset, called `LMD Tempo`, that can be used for training supervised tempo induction algorithms. To this end, we estimated the tempo of the matched audio previews using the algorithm from [31]. Then the associated MIDI files were parsed for tempo change messages. If the value of more than half the tempo messages for a given preview were within 2% of the estimated tempo, we assumed the estimated tempo of the audio excerpts to be correct and added it to `LMD Tempo`. This resulted in 3,611 audio tracks. We were able to match more than 76% of the tracks to the Million Song Dataset (MSD) genre annotations from [29]. Of the matched tracks 29% were labeled `rock`, 27% `pop`, 5% `r&b`, 5% `dance`, 5% `country`, 4% `latin`, and 3% `electronic`. Less than 2% of the tracks were labeled `jazz`, `soundtrack`, `world` and others. Thus it is fair to characterize `LMD Tempo` as a good cross-section of popular music.

[1] https://github.com/GiantSteps/GiantSteps-mtg-key-dataset

2.2 MTG Tempo

The `MTG Key` dataset was created by Faraldo [8] as a ground-truth for key estimation of electronic dance music (`edm`), a genre that is very much underrepresented in `LMD Tempo`. Each two-minute track in `MTG Key` is annotated with one or more keys and a confidence value $c \in \{0, 1, 2\}$ for the key annotation. We annotated those tracks that have an unambiguous key and a confidence of $c = 2$ with a manually tapped tempo, which makes it one of the very few datasets that is suitable for key *and* tempo estimation. The resulting dataset size is 1,159 tracks. In the following we will refer to this new ground-truth as `MTG Tempo`.

2.3 Extended Ballroom

The original `Ballroom` dataset [11] is still used as test dataset today, which is why we exclude it from training. Better suited is the recently released and much larger `Extended Ballroom` dataset. Because it contains some songs also occurring in `Ballroom`, we use the complement `Extended Ballroom` $\setminus$ `Ballroom`. We refer to the resulting dataset as `EBall`. It contains 3,826 tracks with 30 s length each. `EBall` contributes tracks from genres that are underrepresented or simply absent from both `MTG Tempo` and `LMD Tempo`.

2.4 Combined Training Dataset

Combined, `LMD Tempo`, `MTG Tempo`, and `EBall` have a size of 8,596 tracks with tempi ranging from 44 to 216 BPM (Figure 1). In the following we will call it `Train`. The *sweet octave* (i.e., the tempo interval $[\tau, 2\tau)$ that contains the most tracks [31]) for `Train` is $77 - 154$ BPM, covering 84.4% of the items. The shortest interval that covers 99% of the items is $65 - 204$ BPM. Even though many different tempi are represented, `Train` is not tempo-balanced. More than 30% of its tracks have tempi in the $[120, 130)$ interval. Its mean is $\mu = 121.32$ and the standard deviation $\sigma = 30.52$. And while covering many different genres, `Train` is not genre-balanced, either. Genres like `jazz` and `world` only have relatively few representatives. But despite these shortcomings,

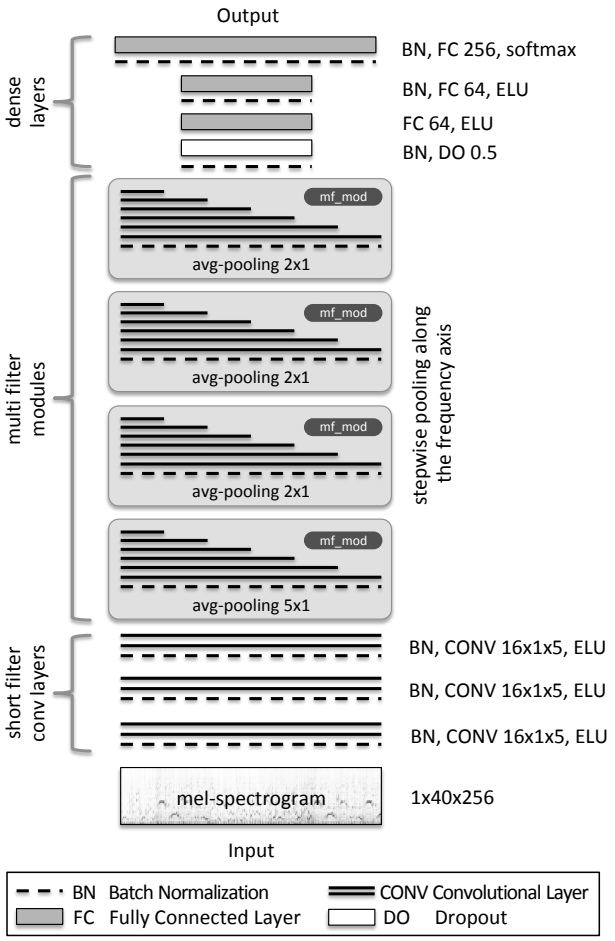


Figure 2: Schematic overview of the network architecture. Three convolutional layers are followed by four *mf_mod* modules, which in turn are followed by four dense layers.

`Train` is a very rich, multi-faceted dataset and completely independent from the test datasets we are going to use for evaluation in Section 4.1.

3. METHOD

Our proposed method for estimating a local tempo consists of a single step. Using a suitable representation we classify the signal with a CNN, which produces a BPM value. We extend the system for global tempo estimation by averaging the $\mathrm{softmax}$ activation function over different parts of a full track.

3.1 Signal Representation

Although we believe that it is possible to build a system like ours with raw audio as input [4, 19], we choose to represent the signal as mel-scaled magnitude spectrogram to reduce the amount of data that needs to be processed by the CNN. The mel-scale as opposed to a linear scale was chosen for its relation to human perception and instrument frequency ranges.

To create the spectrogram, we convert the signal to mono, downsample to $11{,}025$ Hz and use half-overlapping windows of $1{,}024$ samples. This is equivalent to a frame rate of 21.5 Hz, which (according to the Nyquist-Shannon sampling theorem) suffices to represent tempi up to 646 BPM—well above the tempi we usually find in music. Each window is transformed into a 40 band mel-scaled magnitude spectrum covering $20 - 5{,}000$ Hz by applying a Hamming window, the DFT, and a suitable filterbank. Since musical tempo is not an instantaneous quantity, we require a spectrogram of a musically sufficient length. As such we choose 256 frames, equivalent to $\approx 11.9\,\mathrm{s}$.

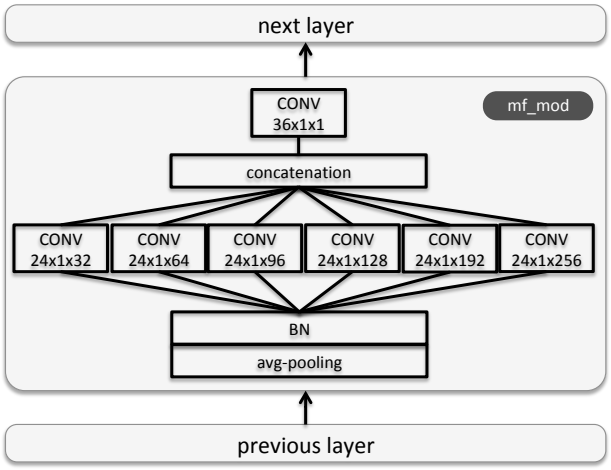


Figure 3: Each multi-filter module *mf_mod* consists of a pooling layer, batch normalization, six different convolutional layers, a concatenation layer and a bottleneck layer. The activation function for all convolutional layers is ELU.

3.2 Network Architecture

Even though tempo estimation appears to be a regression problem, we are approaching it as a classification problem for two reasons. First, a probability distribution over multiple classes allows us to judge how reliable a given estimate is. Additionally, such a distribution is naturally capable of representing tempo ambiguities [21], allowing for the estimation of a second best tempo. Second, in informal experiments we found that a classification-based approach led to more stable results compared to a regression-based approach. So instead of attempting to estimate a BPM value as decimal number, we are choosing one of 256 tempo classes, covering the integer tempo values from 30 to 285 BPM.

The proposed network architecture (Figure 2) is inspired by the traditional approach of first creating an OSS, which is then analyzed for periodicities. In our approach, we first process the input with three convolutional layers with 16 (1×5) filters each. All filters are oriented along the time axis using padding and a stride of 1. Using these fairly short filters, we hope to match onsets in the signal.

These three layers are followed by four almost identical multi-filter modules (*mf_mod*, Figure 3) each consisting of an average pooling layer $(m \times 1)$, parallel convolutional layers with different filter lengths ranging from (1×32) to (1×256), a concatenation layer and a (1×1) bottleneck layer for dimensionality reduction. With each of these

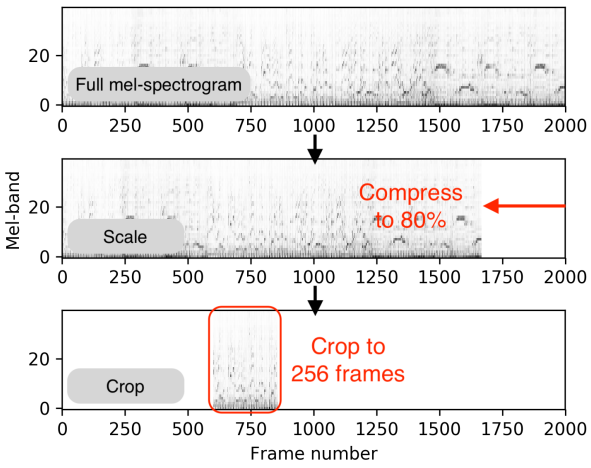


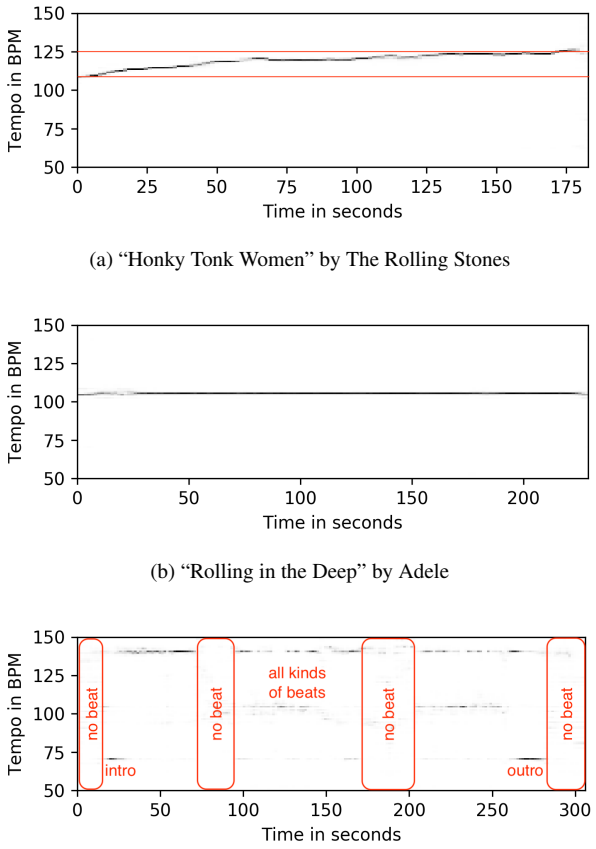
Figure 4: Scale-&-crop data augmentation. During training, the mel-spectrogram is first stretched or compressed along the time axis, which requires an adjustment of the ground-truth label, and then cropped to 256 frames at a randomly chosen offset.

modules we are trying to achieve two goals: 1) Pooling along the frequency axis to summarize mel-bands, and 2) matching the signal with a variety of filters that are capable of detecting long temporal dependencies. Using parallel convolutional layers with different filter lengths has been inspired by [26, 35]. In a traditional system, this could be regarded as some sort of comb filterbank

To classify the features delivered by the convolutional layers, we add two fully connected layers (64 units each) followed by an output layer with 256 units. The output layer uses softmax as activation function, while all other layers use ELU [3]. Each convolutional or fully connected layer is preceded by batch normalization [16]. The first fully connected layer is additionally preceded by a dropout layer with $p = 0.5$ to counter overfitting. As loss function we use categorical cross-entropy. Overall, the network has 2,921,042 trainable parameters.

3.3 Network Training

We use 90% of `Train` for training and 10% for validation. To counter the tempo class imbalance and, at the same time, augment the dataset during training, for each epoch, we use a scale-&-crop-approach borrowed from image recognition systems (see e.g., [34]). Contrary to regular images, the two dimensions of spectrograms have very different meaning, which is why we cannot simply scale-&-crop indiscriminately. Instead, we have to be careful to either not change the labeled meaning of a sample or change its label suitably (Figure 4). In our case this means that we have to preserve the properties of the frequency axis, but may manipulate the time axis. Concretely, we scale the time axis of the samples' mel-spectrograms with a randomly chosen factor $\in \{0.8, 0.84, 0.88, \ldots, 1.16, 1.2\}$ using spline interpolation and adjust the ground-truth tempo labels accordingly. This substantially increases the number



(a) "Honky Tonk Women" by The Rolling Stones

(b) "Rolling in the Deep" by Adele

(c) "Typhoon" by Foreign Beggars/Chasing Shadows

Figure 5: Tempo class probabilities for tracks from different genres and eras. (a) The tempo drift of the performance is clearly visible: the track starts with 108 BPM and ends with 125 BPM. (b) Very stable tempo of a modern pop music production. (c) Dubstep track with several no beat passages, a very active middle section, and halve tempo intro and outro.

of different samples we can present to the network. Since the full mel-spectrogram for a sample is longer than the network input layer (e.g., covering 60 s vs. 11.9 s), we crop each scaled sample at a randomly chosen time axis offset to fit the input layer. This again drastically increases the number of different samples we can offer to the network. After scaling and cropping, the values of the resulting sub-spectrogram are rescaled to $[0, 1]$. In order to ensure comparability, time-axis augmentations are skipped during validation.

We define *Accuracy0* as the fraction of estimates that are correct when rounding decimal ground-truth labels to the nearest integer. To avoid overfitting, we train until *Accuracy0* for the validation set has not improved for 20 epochs using Adam (with a learning rate of 0.001, $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1\text{e}{-8}$) as optimizer, and then keep the model that achieved the highest validation *Accuracy0* (early stopping).

Dataset	schr	böck	new
ACM Mirum	38.3	29.4-	**40.6**
ISMIR04	**37.7**	27.2-	34.1
Ballroom	46.8-	33.8-	**67.9**
Hainsworth	**43.7**	33.8	43.2
GTzan	**38.8**	32.2-	36.9
SMC	14.3	**17.1**	12.4
GiantSteps	53.5-	37.2-	**59.8**
Combined	40.9-	31.2-	**44.8**
DS Average	39.0	30.1	**42.1**

(a) *Accuracy0*

Dataset	schr	böck	new
ACM Mirum	72.3-	74.0-	**79.5**
ISMIR04	**63.4**	55.0	60.6
Ballroom	64.6-	84.0-	**92.0**
Hainsworth	65.8-	**80.6**	77.0
GTzan	**71.0**	69.7	69.4
SMC	31.8	**44.7+**	33.6
GiantSteps	63.1-	58.9-	**73.0**
Combined	66.5-	69.5-	**74.2**
DS Average	61.7	66.7	**69.3**

(b) *Accuracy1*

Dataset	schr	böck	new
ACM Mirum	97.3	**97.7**	97.4
ISMIR04	92.2	**95.0**	92.2
Ballroom	97.0	**98.7**	98.4
Hainsworth	85.6	**89.2+**	84.2
GTzan	93.3	**95.0+**	92.6
SMC	55.3	**67.3+**	50.2
GiantSteps	88.7	86.4-	**89.3**
Combined	92.2	**93.6+**	92.1
DS Average	87.1	**89.9**	86.4

(c) *Accuracy2*

Table 1: Accuracies in percent. The '+' and '−' signs indicate a statistically significant difference between either schr or böck, and new. Bold numbers mark the best-performing algorithm(s) for a dataset. DS Average is the mean of the algorithms' results for each dataset.

3.4 Global Tempo Estimation

Since the input layer is usually shorter than the mel-spectrogram of a whole track, it estimates merely a local tempo. To estimate the global tempo for a track, we calculate multiple output activations using a sliding window with half-overlap, i.e., a hop size of 128 frames $\approx 5.96\,\mathrm{s}$. The activations are averaged class-wise and then—just like in the local approach—the tempo class with the greatest activation is picked as the result.

4. EVALUATION

For evaluation, we trained three models and chose the one with the highest *Accuracy0* measured against the validation set as our final model. As metrics we used *Accuracy0* as well as *Accuracy1* and *Accuracy2*, which are typically used for evaluating tempo estimation systems. *Accuracy1* is defined as the fraction of estimates identical to reference values while allowing a 4% tolerance. *Accuracy2* is the percentage of correct estimates allowing for octave errors 2 and 3 again using a 4% tolerance.

4.1 Global Tempo Benchmarking

It has become customary to benchmark tempo estimation methods with results reported for a small set of datasets: ACM Mirum [24], Ballroom [11], GTzan [36], Hainsworth [12], ISMIR04 [11], GiantSteps Tempo [18], and SMC [13]. The latter was specifically designed to be difficult for beat trackers. Where applicable, we used the corrected annotations from [25]. A detailed description of the datasets is given in [31]. We refer to the union of these seven datasets as Combined. Unweighted averages of results for all seven datasets will be referred to as DS Average. We benchmarked our approach new with the algorithms by Böck et al. (böck) [2] [2] and Schreiber (schr) [31]. Table 1 shows the results.

Overall, new achieves the highest results when tested against Combined with the strict metrics *Accuracy0* ($44.8\,\%$) and *Accuracy1* ($74.2\,\%$). Both accuracy values are slightly lower when summarized as DS Average. When testing with octave-error tolerance, i.e., *Accuracy2*, böck reaches $93.6\,\%$ for Combined, versus $92.2\,\%$ reached by schr, and $92.1\,\%$ reached by new. In essence, new is better than böck at estimating the tempo octave correctly, while böck—and to a lesser degree schr—achieve a slightly higher accuracy when ignoring the metrical level. This may be due to the fact that both böck and schr use a traditional periodicity analysis (DFT and comb filters, respectively) that tends to be prone to octave errors, while new does not use a comparable isolated component.

When inspecting the dataset-specific results, we find that new's *Accuracy1* is particularly high for Ballroom ($92.0\,\%$), GiantSteps ($73.0\,\%$), and ACM Mirum ($79.5\,\%$). In fact, they are significantly higher than böck's ($+8.0\,\mathrm{pp}/+14.1\,\mathrm{pp}/+5.5\,\mathrm{pp}$) or schr's ($+27.4\,\mathrm{pp}/+9.9\,\mathrm{pp}/+7.2\,\mathrm{pp}$) results. Both the Ballroom and GiantSteps values can be explained through our training dataset. They clearly correspond to EBall and MTG Tempo, therefore high values are not surprising. We believe the same is true for ACM Mirum and LMD Tempo. To us these results indicate that a genre-complete training set may lead to better results for the other datasets as well. This hypothesis is supported by the fact that GTzan contains genres like reggae, classical, blues, and jazz, and Hainsworth contains the genres choral, classical, folk, and jazz—none of which are well represented in Train. For both datasets new performs worse than böck or schr. A similar connection may exist for böck and GiantSteps—as far as we know, böck has not been trained on edm.

4.2 Local Tempo Visualization

To illustrate the system's performance for continuous local tempo estimation, we analyzed several tracks from different genres using overlapping windows with a relatively small hop size of 32 frames, i.e., ≈ 1.5 seconds. For clarity, we cropped the images at 50 and 150 BPM. Figure 5a beautifully reveals the tempo drift in The Rolling Stone's 1969 performance of "Honky Tonk Women", starting out at 108 BPM and ending in 125 BPM. In contrast, Adele's recent studio production "Rolling in the Deep" (Figure 5b) stays very stable at 105 BPM. A more complicated picture is presented by the dubstep track "Typhoon" by Foreign

[2] madmom-0.15.1, default options, available at https://github.com/CPJKU/madmom

Beggars/Chasing Shadows (Figure 5c). After several seconds of weather noises, the intro starts with 70 BPM. The main part's tempo is clearly 140 BPM interrupted by two sections with no beat. The outro again feels like 70 BPM followed by a fade out.

5. CONCLUSIONS

We have presented a single-step tempo estimation system consisting of a convolutional neural network (CNN). With a conventional mel-spectrogram as input, the system is capable of estimating the musical tempo using multi-class classification. The network's architecture consolidates traditional multi-step approaches into a single CNN, avoiding explicit mid-level features such as onset strength signals (OSS) or beat activation functions. Consequently and contrary to many other systems, our approach does not rely on handcrafted features or ad-hoc heuristics, but is completely data-driven. The system was trained with samples from the union of several large datasets, two of which were newly created. To aid training, we applied problem-specific data augmentation techniques. For global tempo estimation, we have shown that our single network, data-driven approach performs as well as or better than other more complicated state-of-the-art systems, especially w.r.t. *Accuracy1*. Furthermore, by visualizing examples for local tempo estimations, we have demonstrated qualitatively how the system can aid music analysis, e.g., to identify tempo drift.

We believe that the system can be improved even further by training with a more balanced dataset that contains tracks for all tested genres. Notably missing from the current training set are `jazz`, `classical`, or `reggae` tracks. Another area of potential improvement is the network architecture. Shorter filters, dilated convolutions, residual connections, and a suitable replacement for the fully connected layers might be used to reduce the number of parameters and thus the number of operations needed for training and estimation.

Additional Material

Datasets are available at `http://www.tagtraum.com/tempo_estimation.html`. Code to estimate tempi and create tempograms is available at `https://github.com/hendriks73/tempo-cnn`.

Acknowledgments

The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. Meinard Müller is supported by the German Research Foundation (DFG MU 2686/11-1).

6. REFERENCES

[1] Miguel Alonso, Bertrand David, and Gaël Richard. Tempo and beat estimation of musical signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004.

[2] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 625–631, Málaga, Spain, 2015.

[3] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, Feb. 2015.

[4] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7014–7018, Florence, Italy, 2014. IEEE.

[5] Simon Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.

[6] Anders Elowsson. Beat tracking with a cepstroid invariant neural network. In *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, pages 351–357, New York, NY, USA, 2016.

[7] Anders Elowsson and Anders Friberg. Modeling the perception of tempo. *The Journal of the Acoustical Society of America*, 137(6):3163–3177, 2015.

[8] Ángel Faraldo, Sergi Jordà, and Perfecto Herrera. A multi-profile method for key estimation in EDM. In *Proceedings of the AES International Conference on Semantic Audio*, Erlangen, Germany, June 2017. Audio Engineering Society.

[9] Aggelos Gkiokas, Vassilios Katsouros, and George Carayannis. Reducing tempo octave errors by periodicity vector coding and svm learning. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 301–306, Porto, Portugal, 2012.

[10] Aggelos Gkiokas and Vassilis Katsouros. Convolutional neural networks for real-time beat tracking: A dancing robot application. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 286–293, Suzhou, China, October 2017.

[11] Fabien Gouyon, Anssi P. Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[12] Stephen Webley Hainsworth. *Techniques for the Automated Analysis of Musical Audio*. PhD thesis, University of Cambridge, UK, September 2004.

[13] Andre Holzapfel, Matthew E.P. Davies, José R. Zapata, João Lobato Oliveira, and Fabien Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, 2012.

[14] Florian Hörschläger, Richard Vogl, Sebastian Böck, and Peter Knees. Addressing tempo estimation octave errors in electronic music by incorporating style information extracted from wikipedia. In *Proceedings of the Sound and Music Computing Conference (SMC)*, Maynooth, Ireland, 2015.

[15] Eric J Humphrey, Juan Pablo Bello, and Yann LeCun. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 403–408, Porto, Portugal, 2012.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

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

[18] Peter Knees, Ángel Faraldo, Perfecto Herrera, Richard Vogl, Sebastian Böck, Florian Hörschläger, and Mickael Le Goff. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 364–370, Málaga, Spain, October 2015.

[19] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 220–226, Espoo, Finland, July 2017.

[20] Ugo Marchand and Geoffroy Peeters. The extended ballroom dataset. In *Late Breaking Demo of the International Conference on Music Information Retrieval (ISMIR)*, New York, NY, USA, 2016.

[21] Martin F. McKinney and Dirk Moelants. Deviations from the resonance theory of tempo induction. In *Proceedings of the Conference on Interdisciplinary Musicology*, Graz, Austria, 2004.

[22] Meinard Müller. *Fundamentals of Music Processing – Audio, Analysis, Algorithms, Applications*. Springer Verlag, 2015.

[23] Geoffroy Peeters. Template-based estimation of time-varying tempo. *EURASIP Journal on Advances in Signal Processing*, 2007(1):158–158, 2007.

[24] Geoffroy Peeters and Joachim Flocon-Cholet. Perceptual tempo estimation using GMM-regression. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies (MIRUM)*, pages 45–50, New York, NY, USA, 2012. ACM.

[25] Graham Percival and George Tzanetakis. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(12):1765–1776, 2014.

[26] Jordi Pons and Xavier Serra. Designing efficient architectures for modeling temporal features with convolutional neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2472–2476, New Orleans, USA, March 2017. IEEE.

[27] Colin Raffel. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. PhD thesis, Columbia University, 2016.

[28] Eric D. Scheirer. Tempo and beat analysis of acoustical musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[29] Hendrik Schreiber. Improving genre annotations for the million song dataset. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 241–247, Málaga, Spain, 2015.

[30] Hendrik Schreiber and Meinard Müller. Exploiting global features for tempo octave correction. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 639–643, Florence, Italy, 2014.

[31] Hendrik Schreiber and Meinard Müller. A post-processing procedure for improving music tempo estimates using supervised learning. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pages 235–242, Suzhou, China, October 2017.

[32] Björn Schuller, Florian Eyben, and Gerhard Rigoll. Tango or waltz?: Putting ballroom dance style into tempo detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2008:12, 2008.

[33] Jarno Seppänen. Tatum grid analysis of musical signals. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 131–134, 2001.

[34] Patrice Y. Simard, David Steinkraus, John C. Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of*

the 7th International Conference on Document Analysis and Recognition (ICDAR), volume 3, pages 958–962, August 2003.

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Boston, MA, USA, June 2015.

[36] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[37] Fu-Hai Frank Wu. Musical tempo octave error reducing based on the statistics of tempogram. In *23th Mediterranean Conference on Control and Automation (MED)*, pages 993–998, Torremolinos, Spain, 2015. IEEE.

[38] Fu-Hai Frank Wu and Jyh-Shing Roger Jang. A supervised learning method for tempo estimation of musical audio. In *22nd Mediterranean Conference of Control and Automation (MED)*, pages 599–604, Palermo, Italy, 2014. IEEE.

[39] Jose R. Zapata and Emilia Gómez. Comparative evaluation and combination of audio tempo estimation approaches. In *42nd AES Conference on Semantic Audio*, Ilmenau, Germany, 2011.