DRUM TRANSCRIPTION VIA JOINT BEAT AND DRUM MODELING USING CONVOLUTIONAL RECURRENT NEURAL NETWORKS

Richard Vogl[1,2] **Matthias Dorfer**[2] **Gerhard Widmer**[2] **Peter Knees**[1]
[1] Institute of Software Technology & Interactive Systems, Vienna University of Technology, Austria
[2] Dept. of Computational Perception, Johannes Kepler University Linz, Austria
{richard.vogl, peter.knees}@tuwien.ac.at

ABSTRACT

Existing systems for automatic transcription of drum tracks from polyphonic music focus on detecting drum instrument onsets but lack consideration of additional meta information like bar boundaries, tempo, and meter. We address this limitation by proposing a system which has the capability to detect drum instrument onsets along with the corresponding beats and downbeats. In this design, the system has the means to utilize information on the rhythmical structure of a song which is closely related to the desired drum transcript. To this end, we introduce and compare different architectures for this task, i.e., recurrent, convolutional, and recurrent-convolutional neural networks. We evaluate our systems on two well-known data sets and an additional new data set containing both drum and beat annotations. We show that convolutional and recurrent-convolutional neural networks perform better than state-of-the-art methods and that learning beats jointly with drums can be beneficial for the task of drum detection.

1. INTRODUCTION

The automatic creation of symbolic transcripts from music in audio files is an important high-level task in music information retrieval. Automatic music transcription systems (AMT) aim at solving this task and have been proposed in the past (cf. [1]), but there is yet no general solution to this problem. The transcription of the drum instruments from an audio file of a song is a sub-task of automatic music transcription, called automatic drum transcription (ADT). Usually, such ADT systems focus solely on the detection of drum instrument note onsets. While this is the necessary first step, for a full transcript of the drum track more information is required. Sheet music for drums—equally to sheet music for other instruments—contains additional information required by a musician to perform a piece. This information comprises (but is not limited to): meter, overall tempo, indicators for bar boundaries, indications for local changes in tempo, dynamics, and playing style of the piece. To obtain some of this information, beat and downbeat detection methods can be utilized. While beats provide tempo information, downbeats add bar boundaries, and the combination of both provides indication for the meter within the bars.

In this work, neural networks for joint beat and drum detection are trained in a multi-task learning fashion. While it is possible to extract drums and beats separately using existing work and combine the results afterwards, we show that it is beneficial to train for both tasks together, allowing a joint model to leverage commonalities of the two problems. Additionally, recurrent (RNN), convolutional (CNN) and convolutional-recurrent neural network (CRNN) models for drum transcription and joint beat and drum detection are evaluated on two well-known, as well as a new data set.

The remainder of this work is structured as follows. In the next section, we discuss related work. In sec. 3, we describe the implemented drum transcription pipeline used to evaluate the network architectures, followed by a section discussing the different network architectures (sec. 4). In sec. 5, we explain the experimental setup to evaluate the joint learning approach. After that, a discussion of the results follows in sec. 6 before we draw conclusions in sec. 7.

2. RELATED WORK

While in the past many different approaches for ADT have been proposed [11,13,15,16,22,24,25,34,38], recent work focuses on end-to-end approaches calculating activation functions for each drum instrument. These methods utilize non-negative matrix factorization (NMF, e.g. adaptive-NMF in Dittmar et al. [7] and partially fixed NMF in Wu et al. [37]) as well as RNNs (RNNs with label time-shift in Vogl et al. [35, 36] and bidirectional RNNs in Southall et al. [31]) to extract the activation functions from spectrograms of the audio signal. Such activation-function-based end-to-end ADT systems circumvent certain issues associated with other architectures. Methods which first segment the song (e.g. using onset detection) and subsequently classify these segments [22, 23, 38] suffer from a loss of information after the segmentation step—i.e. whenever the system fails to detect a segment, this information is lost. Such systems heavily depend on the accuracies of the single components, and can never perform better than the weakest component in the pipeline. Additionally, information of the input signal which is discarded after a processing step might still be of value for later steps.

Since RNNs, especially long short-term memory (LSTM) [17] and gated recurrent unit (GRU) [5] networks, are designed to model long term relationships, one might suspect that systems based on RNNs [31, 35, 36] can leverage the repetitive structure of the drum tracks and make use of this information. Contrary to this intuition this is not the case for RNN-based systems proposed so far. Both the works of Vogl et al. [35, 36] and Southall et al. [31] use snippets with length of only about one second to train the RNNs. This prohibits learning long-term structures of drum rhythms which are typically in the magnitude of two or more seconds. In [35], it has been shown that RNNs with time-shift perform equally well as bidirectional RNNs, and that backward directional RNNs perform better than forward directional RNNs. Combining these findings indicates that the learned models actually mostly consider local features. Therefore, RNNs trained in such a manner seem to learn only an acoustic, but not a structural model for drum transcription.

Many works on joint beat and downbeat tracking have been published in recent years [2, 9, 10, 19–21, 26]. A discussion of all the different techniques would go beyond the scope of this work. One of the most successful methods by Böck et al. [2] is a joint beat and downbeat tracking system using bidirectional LSTM networks. This approach achieves top results in the 2016 MIREX task for beat detection and can be considered the current state of the art. [1]

In this work, a multi-task learning strategy is used to address the discussed issues of current drum transcription systems, cf. [4]. The use of a model jointly trained on drum and beat annotations, combined with longer training snippets, allows the model to learn long-term relations of the drum patterns in combination with beats and downbeats. Furthermore, learning multiple related tasks simultaneously at once can improve results for the single tasks. To this end, different architectures of RNNs, CNNs, and a combination of both, convolutional-recurrent neural networks (CRNNs) [8, 27, 39], are evaluated.

The rationale behind selecting these three methods for comparison is as follows. RNNs have proven to be well-suited for both drum and beat detection, as well as learning long-term dependencies for music language models [30]. CNNs are among the best performing methods for many image processing and other machine learning tasks, and have been used on spectrograms of music signals in the past. For instance, Schlüter and Böck [28] use CNNs to improve onset detection results, while Gajhede et al. [12] use CNNs to successfully classify samples of three drum sound classes on a non-public data set. CRNNs should result in a model, in which the convolutional layers focus on acoustic modeling of the events, while the recurrent layers learn temporal structures of the features.

3. DRUM TRANSCRIPTION PIPELINE

The implemented method is an ADT system using a similar pipeline as presented in [31] and [36]. Fig. 1 visualizes the overall structure of the system. The next subsections discuss the single blocks of the system in more detail.

[1] http://www.music-ir.org/mirex/wiki/2016:MIREX2016_Results

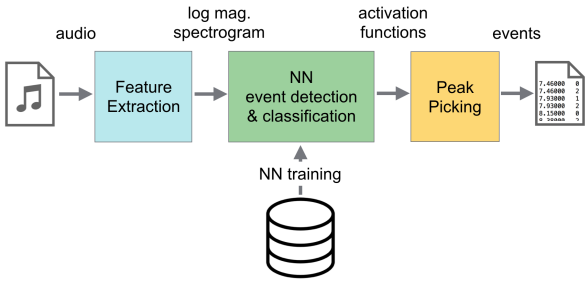

Figure 1. System overview of the implemented drum transcription pipeline used to evaluate the different neural network architectures.

3.1 Feature Extraction

First, a logarithmic magnitude spectrogram is calculated using a 2048-samples window size and a resulting frame rate of 100Hz from a 44.1kHz 16bit mono audio signal input. Then, the frequency bins are transformed to a logarithmic scale using triangular filters (twelve per octave) in a frequency range from 20 to 20,000 Hz. Finally, the positive first-order-differential over time of this spectrogram is calculated and concatenated. This results in feature vectors with a length of 168 values (2x84 frequency bins).

3.2 Activation Function Calculation

The central block in fig. 1 represents the activation function calculation step. This task is performed using a neural network (NN) trained on appropriate training data (see sec. 4). As in most of the related work, we only consider three drum instruments: bass- or kick drum, snare drum, and hi-hat.

While the architectures of the single NNs are different, they share certain commonalities: *i.* all NNs are trained using the same input features; *ii.* the RNN architectures are implemented as bidirectional RNNs (BRNN) [29]; *iii.* the output layers consist of three or five sigmoid units, representing three drum instruments under observation (drum only) or three drum instruments plus beat and downbeat (drum and beats), respectively; and *iv.* the NNs are all trained using the RMSprop optimization algorithm proposed by Tieleman et al. [33], using mini-batches of size eight. For training, we follow a three-fold cross validation strategy on all data sets. Two splits are used for training, 15% of the training data is separated and used for validation after each epoch, while testing/evaluation is done on the third split. The NNs are trained using a fixed learning rate with additional refinement if no improvement on the validation set is achieved for 10 epochs. During refinement the learning rate is reduced and training continues using the parameters of the best performing model so far.

More details on the individual NN architectures are provided in sec. 4.

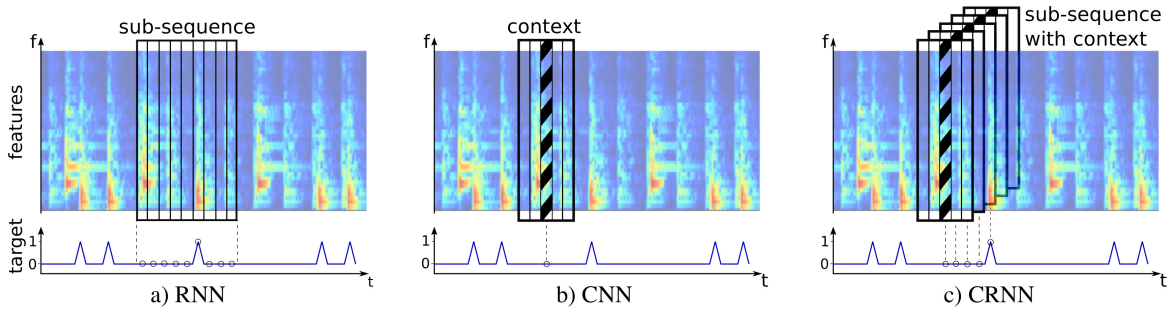

Figure 2. Comparison of mode of operation of RNNs, CNNs, and CRNNs on spectrograms of audio signals. RNNs process the input in a sequential manner. Usually, during training, only sub-sequences of the input signal are used to reduce the memory footprint of the networks. CNNs process the signal frame by frame without being aware of sequences. Because of this, a certain spectral context is added for each input frame. CRNNs, like RNNs, process the input sequentially, but additionally, a spectral context is added to every frame on which convolution is performed by the convolutional layers.

3.3 Preparation of Target Functions

For training the NNs, target functions of the desired output are required besides the input features. These target functions are generated by setting frames of a signal with the same frame rate as the input features to 1 whenever an annotation is present and to 0 otherwise. A separate target function is created for each drum instrument as well as for beats and downbeats.

3.4 Peak Picking

In the last step of our pipeline (rightmost block of fig. 1), the drum instrument onsets (and beats if applicable) are identified using a simple peak picking method introduced for onset detection in [3]: A point n in the activation function $f_a(n)$ is considered a peak if these terms are fulfilled:

1. $f_a(n) = max(f_a(n-m), \cdots, f_a(n))$,
2. $f_a(n) \geq mean(f_a(n-a), \cdots, f_a(n)) + \delta$,
3. $n - n_{lp} > w$,

where δ is a variable threshold. A peak must be the maximum value within a window of size $m+1$, and exceeding the mean value plus a threshold within a window of size $a+1$. Additionally, a peak must have at least a distance of $w+1$ to the last detected peak (n_{lp}). Values for the parameters were tuned on a development data set to be: $m = a = w = 2$.

The threshold for peak picking is determined on the validation set. Since the activation functions produced by the NN contain little noise and are quite spiky, rather low thresholds $(0.1 - 0.2)$ give best results.

4. NEURAL NETWORK MODELS

In this section, we explore the properties of the neural network models considered more closely. Of the NN categories mentioned before, we investigate three different types: bidirectional recurrent networks (BRNN), convolutional networks (CNN), and convolutional bidirectional recurrent networks (CBRNN). For every class of networks, two different architectures are implemented: *i.* a smaller network, with less capacity, trained on shorter subsequences (with focus only on acoustic modeling), and *ii.* a larger network, trained on longer subsequences (with additional focus on pattern modeling).

Even though we previously showed that RNNs with label time-shift achieve similar performance as BRNNs [35, 36], in this work, we will not use time-shift for target labels. This is due to three reasons: *i.* the focus of this work is not real-time transcription but a comparison of NN architectures and training paradigms, therefore using a bidirectional architecture has no downsides; *ii.* it is unclear how label time-shift would affect CNNs; *iii.* in [2], the effectiveness of BRNNs (BLSTMs) for beat and downbeat tracking is shown. Thus, in the context of this work, using BRNNs facilitates combining state-of-the-art drum and beat detection methods while allowing us to compare CNNs and RNNs in a fair manner.

4.1 Bidirectional Recurrent Neural Network

Gated recurrent units (GRUs [5]) are similar to LSTMs in the sense that both are gated RNN-cell types that facilitate learning of long-term relations in the data. While LSTMs feature forget, input, and output gates, GRUs only exhibit two gates: update and output. This makes the GRU less complex in terms of number of parameters. It has been shown that both are equally powerful [6], with the difference that more GRUs are needed in an NN layer to achieve the same model capacity as with LSTMs, resulting in more or less equal number of total parameters. An advantage of using GRUs is that hyperparameter optimization for training is usually easier compared to LSTMs.

In this work, two bidirectional GRU (BGRU) architectures are used. The small model (BGRU-a) features two layers of 50 nodes each, and is trained on sequences of 100 frames; the larger model (BGRU-b) consists of three layers of 30 nodes each, and is trained on sequences of 400 frames. For training an initial learning rate of 0.007 is used.

	Frames	Context	Conv. Layers	Rec. Layers	Dense Layers
BGRU-a	100	—	—	2 x 50 GRU	—
BGRU-b	400	—	—	3 x 30 GRU	—
CNN-a	—	9	1xA + 1xB	—	2 x 256
CNN-b	—	25	1xA + 1xB	—	2 x 256
CBGRU-a	100	9	1xA + 1xB	2 x 50 GRU	—
CBGRU-b	400	13	1xA + 1xB	3 x 60 GRU	—

Table 1. Overview of used neural network model architectures and parameters. Every network additionally contains a dense sigmoid output layer. Conv. block A consists of 2 layers with 32 3x3 filters and 3x3 max-pooling; conv. block B consists of 2 layers with 64 3x3 filters and 3x3 max-pooling; both use batch normalization.

4.2 Convolutional Neural Network

Convolutional neural networks have been successfully applied not only in image processing, but also many other machine learning tasks. The convolutional layers are constructed using two different building blocks: block *A* consists of two layers with 32 3x3 filters and block *B* consists of two layers with 64 3x3 filters; both in combination with batch normalization [18], and each followed by a 3x3 max pooling layer and a drop-out layer ($\lambda = 0.3$) [32].

For both CNN models, block *A* is used as input, followed by block *B*, and two fully connected layers of size 256. The only difference between the small (CNN-a) and the large (CNN-b) model is the context used to classify a frame: 9 and 25 frames are used for CNN-a and CNN-b respectively. While plain CNNs do not feature any memory, the spectral context allows the CNN to access surrounding information during training and classification. However, a context of 25 frames (250ms) is not enough to find repetitive structures in the rhythm patterns. Therefore, the CNN can only rely on acoustic, i.e., spectral features of the signal. Nevertheless, with advanced training methods like batch normalization, as well as the advantage that CNNs can easily learn pitch invariant kernels, CNNs are well-equipped to learn a task adequate acoustic model. For training an initial learning rate of 0.001 is used.

4.3 Convolutional Bidirectional RNN

Convolutional recurrent neural networks (CRNN) represent a combination of CNNs and RNNs. They feature convolutional layers as well as recurrent layers. Different implementations are possible. In this work, the convolutional layers directly process the input features, i.e. spectrogram representations, meant to learn an acoustic model (cf. 2D image processing tasks). The recurrent layers are placed after the convolutional layers and are supposed to serve as a means for the network to learn structural patterns.

For this class of NN, the two versions differ in the following aspects: CBGRU-a features 2 recurrent layers with 30 GRUs each, uses a spectral context of 9 frames for convolution, and is trained on sequences of length 100; while CBGRU-b features 3 recurrent layers with 60 GRUs each, uses a spectral context of 13 frames, and is trained on sequences of length 400. For training an initial learning rate of 0.0005 is used.

Table 1 recaps the information of the previous sections in a more compact form. Figure 2 visualizes the modes of operation of the different NN architectures on the input spectrograms.

5. EVALUATION

For evaluation of the introduced NN architectures, the different models are individually trained on single data sets in a three-fold cross-validation manner. For data sets which comprise beat annotations, three different experiments are performed (explained in more detail in section 5.2); using data sets only providing drum annotations, just the drum detection task is performed.

5.1 Data Sets

In this work, the different methods are evaluated using three different data sets, consisting of two well-known and a newly introduced set.

5.1.1 IDMT-SMT-Drums v.1 (SMT)

Published along with [7], the IDMT-SMT-Drums [2] data set comprises tracks containing three different drum-set types. These are: *i.* real-world, acoustic drum sets (titled *RealDrum*), *ii.* drum synthesizers (*TechnoDrum*), and *iii.* drum sample libraries (*WaveDrum*). It consists of 95 simple drum tracks containing bass drum, snare drum and hi-hat only. The tracks have an average length of 15s and a total length of 24m. Also included are additional 285 shorter, single-instrument training tracks as well as 180 single instrument tracks for 60 of the 95 mixture tracks (from the *WaveDrum02* subset)—intended to be used for source separation experiments. These additional single instrument tracks are used as additional training samples (together with their corresponding split) but not for evaluation.

5.1.2 ENST Drums (ENST)

The ENST-Drums set [14] contains real drum recordings of three different drummers performing on different drum kits. [3] Audio files for separate solo instrument tracks

[2] https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/drums.html

[3] http://perso.telecom-paristech.fr/~grichard/ENST-drums/

	Input Features		Target Functions	
	Spectrogram	Beats	Drums	Beats
DT	✓		✓	
BF	✓	✓	✓	
MT	✓		✓	✓

Table 2. Overview of experimental setup. Rows represent individual tasks and show their input feature and target function combinations.

as well as for two mixtures are included. Additionally, accompaniment tracks are available for a subset of the recordings—the so called minus-one tracks. In this work, the wet mixes (contains standard post-processing like compression and equalizing) of the minus-one tracks were used. They make up 64 tracks of 61s average length and a total length of 1h.

Evaluation was performed on the drum-only tracks (ENST solo) as well as the mixes with their accompaniment tracks (ENST acc.). Since the ENST-Drums data set contains more than the three instruments under observation, only the snare, bass, and hi-hat annotations were used.

5.1.3 RBMA Various Assets 2013 (RBMA13)

This new data set consists of the 30 tracks of the freely available 2013 Red Bull Music Academy Various Assets sampler. [4] The sampler covers a variety of electronically produced music, which encompasses electronic dance music (EDM) but also singer-songwriter tracks and even fusion-jazz styled music. Three tracks on the sampler do not contain any drums and are therefore ignored. Annotations for drums, beats, and downbeats were manually created. Tracks in this set have an average length of 3m 50s. The total length of the data set is 1h 43m.

This data set is different from the other two data sets in three aspects: *i.* it contains quite diverse drum sounds, *ii.* the drum patterns are arranged in the usual song-structure within a full length track, and *iii.* most of the tracks contain singing voice, which showed to be a challenge for systems solely trained on music without singing voice. The annotations for drums and beats have been manually created and are publicly available for download. [5]

5.2 Experimental Setup

To compare the different NN architectures, and evaluate them in the context of ADT using joint learning of beat and drum activations, the following experiments were performed.

5.2.1 Drum Detection (DT)

In this set of experiments, the features as explained in sec. 3.1 and target functions generated from the drum annotations described in sec. 3.3 are used for NN training.

[4] https://rbma.bandcamp.com/album/various-assets-not-for-sale-red-bull-music-academy-new-york-2013

[5] http://ifs.tuwien.ac.at/~vogl/datasets/

	SMT	ENST		RBMA13		
		solo	acc.	DT	BF	MT
GRUts [36]	*92.5*	*83.3*	*75.0*	-	-	-
BGRU-a	93.0	80.9	70.1	59.8	63.6	64.6
BGRU-b	93.3	82.9	72.3	61.8	64.5	64.3
CNN-a	87.6	78.6	70.8	66.2	66.7	63.3
CNN-b	93.4	**85.0**	78.3	66.8	65.2	64.8
CBGRU-a	**95.2**	84.6	76.4	65.2	66.1	66.9
CBGRU-b	93.8	83.9	**78.4**	**67.3**	**68.4**	**67.2**

Table 3. F-measure results for the evaluated models on different data sets. The columns DT, BF, and MT show results for models trained only for drum detection, trained using oracle beats as additional input features, and simultaneously trained on drums and beats, respectively. Bold values represent the best performance for an experiment across models. The baseline can be found in the first row.

These experiments are comparable to the ones in the related work, since we use a similar setup. As baseline, the results in [36] are used. The results of this set of experiments allow to compare the performance of different NN architectures for drum detection.

5.2.2 Drum Detection with Oracle Beat Features (BF)

For this set of experiments, in addition to the input features explained in sec. 3.1, the annotated beats, represented as the target functions for beats and downbeats, are included as input features. As targets for NN training only the drum target functions are utilized. Since beat annotations are required for this experiment, only data sets comprising beat annotations can be used. Using the results of these experiments, it can be investigated if the prior knowledge of beat and downbeat positions is beneficial for drum detection.

5.2.3 Joint Drum and Beat Detection (MT)

This set of experiments represents the multi-task learning investigation. As input for training, again, only the spectrogram features are used. Targets for training of the NNs comprise, in this case, drum and beat activation functions. As discussed in the introduction, in some cases it can be beneficial to train related properties simultaneously. Beats and drums are closely related, because usually drum pattern are repetitive on a bar-level (separated by downbeats) and drum onsets often correlate with beats.

The insight which can be drawn from these experiments is whether simultaneous training of drums, beats, and downbeats is beneficial. It is of interest if the resulting performance is higher than the one achieved for DT; and also if it is below, comparable, or even surpasses the results in the BF experiment series.

Table 2 gives an overview of the properties of the experiments and the used feature/target combination.

5.3 Evaluation Method

To evaluate the performance of the different architectures and training methods, the well-known metrics precision,

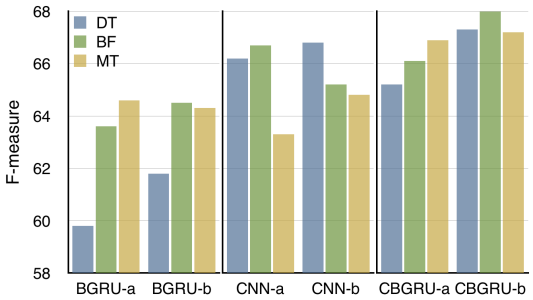

Figure 3. Results for RBMA13 data set, highlighting the influence of oracle beat features (BF) and multi-task learning (MT). While recurrent models (left and right) benefit, convolutional models (center) do not.

recall, and F-measure are used. These are calculated for drum instrument onsets as well as beat positions. True positive, false positive, and false negative onset and beat positions are identified by using a $20ms$ tolerance window. This is in line with the evaluation in [36] which is used as baseline for the experiments of this work. Note that other work, e.g. [7, 25, 31], uses less strict tolerance windows of $30ms$ or $50ms$ for evaluation.

6. RESULTS AND DISCUSSION

Table 3 shows the F-measure results for the individual NN architectures on the data sets used for evaluation. The results for BGRU-a and BGRU-b on the ENST data set are lower than for the baseline, although the models should be comparable. This is due to the fact that in [36] data augmentation is applied. This is especially helpful in the case of the ENST data set, since e.g. the pitches of the base drums vary greatly over the different drum kits. The results for CNN-a are lower than the state of the art, which implies that the context of 9 frames is too small to detect drum events using a CNN. All other results on the ENST and SMT data sets represent an improvement over the state of the art. This shows that CNN with a large enough spectral context (25 frames in this work) can detect drum events better than RNNs. A part of the large increase for the ENST data set can be attributed to the fact that CNNs can model pitch invariance easier than RNNs.

The results for the MT experiments show the following tendencies: For the BGRU-a and BGRU-b models, an improvement can be observed when applying multi-task learning. Compared to using oracle beats (BF) for training, the improvement is higher for BGRU-a and similar in the case of BGRU-b. This result is interesting for two reasons: *i.* although BGRU-a is trained on short sequences, an improvement can be observed, and *ii.* the improvement is comparable to that when using oracle beats (BF) although the beat tracking results are low. This could imply that multi-task learning is also beneficial for the acoustic model of the system. As expected, the CNNs (CNN-a, CNN-b) can not improve when using multi-task learning, but rather the results deteriorate. In case of the convolutional-recurrent models, the result for CBGRU-a is similar to BGRU-a. In case of CBGRU-b no improvement of drum detection performance using multi-task learning can be observed, although it is the case using oracle beats (BF). We attribute this to the fact that CBGRU-b has enough capacity for good acoustic modeling, while the low beat detection results limit the effects of multi-task learning on this level.

BLSTM [2]	*85.6*
BGRU-a	46.4
BGRU-b	46.2
CNN-a	44.9
CNN-b	46.9
CBGRU-a	47.6
CBGRU-b	48.8

Table 4. F-measure results for beat detection for the multi-task learning experiments compared to a state-of-the-art approach (first row) on the RBMA13 set.

Table 4 shows the F-measure results for beat and downbeat tracking. The results are all below the state-of-the-art beat tracker used as baseline [2]. This is due to several factors. In [2], *i.* much larger training sets for beat and downbeat tracking are used, *ii.* the LSTMs are trained on full sequences of the input data, giving the model more context, and *iii.* an additional music language model in the form of a dynamic Bayesian network (DBN) is used.

The results for CNNs and CRNNs show that convolutional feature processing is beneficial for drum detection. The finding considering drum detection results for multi-task learning are also promising. The low results of beat and downbeat tracking are certainly a limiting factor and probably the reason for the lack of improvement for MT over DT in the case of BGRU-b. As a next step, to better leverage multi-task learning effects, beat detection results must be improved using similar techniques as in [2].

7. CONCLUSIONS

In this work, convolutional and convolutional-recurrent NN models for drum transcription were introduced and compared to the state of the art of recurrent models. The evaluation shows that the new models are able to outperform this state of the art. Furthermore, an investigation whether *i.* beat and downbeat input features are beneficial for drum detection, and *ii.* this benefit is also achievable using multi-task learning of drums, beats, and downbeats, was conducted. The results show that this is the case, although the low beat and downbeat detection results achieved with the implemented architectures is a limiting factor. While the goal of this work was not to improve the capabilities of beat and downbeat tracking per se, future work will focus on improving these aspects, as we believe this will have an overall positive impact on the performance of the joint model. The newly created data set consisting of freely available music and annotations for drums, beats and downbeats will be an asset for this line of research to the community.

8. ACKNOWLEDGEMENTS

This work has been partly funded by the Austrian FFG under the BRIDGE 1 project *SmarterJam* (858514), by the EU's seventh Framework Programme FP7/2007-13 for research, technological development and demonstration under grant agreement no. 610591 (*GiantSteps*), as well by the Austrian ministries BMVIT and BMWFW, and the province of Upper Austria via the COMET Center SCCH. We would like to thank Wulf Gaebele and the annotators Marc Übel and Jo Thalmayer from the Red Bull Music Academy, as well as Sebastian Böck for advice and help with beat and downbeat annotations and detection.

9. REFERENCES

[1] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[2] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *Proc. 17th Intl Society for Music Information Retrieval Conf (ISMIR)*, 2016.

[3] Sebastian Böck and Gerhard Widmer. Maximum filter vibrato suppression for onset detection. In *Proc. 16th Intl Conf on Digital Audio Effects (DAFx)*, 2013.

[4] Rich Caruana. Multitask learning. In Thrun and Pratt (eds.) *Learning to learn*, pages 95–133. Springer, 1998.

[5] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. Learning phrase representations using rnn encoderdecoder for statistical machine translation. In *Proc. Conf on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. `http://arxiv.org/abs/1412.3555`, 2014.

[7] Christian Dittmar and Daniel Gärtner. Real-time transcription and separation of drum recordings based on nmf decomposition. In *Proc. 17th Intl Conf on Digital Audio Effects (DAFx)*, 2014.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. IEEE conference on computer vision and pattern recognition (CVPR)*, 2015.

[9] Simon Durand, Juan P Bello, Bertrand David, and Gaël Richard. Downbeat tracking with multiple features and deep neural networks. In *Proc. 40th IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[10] Simon Durand, Juan P Bello, Bertrand David, and Gaël Richard. Feature adapted convolutional neural networks for downbeat tracking. In *Proc. 41st IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[11] Derry FitzGerald, Bob Lawlor, and Eugene Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *Proc. Irish Signals & Systems Conf*, 2003.

[12] Nicolai Gajhede, Oliver Beck, and Hendrik Purwins. Convolutional neural networks with batch normalization for classifying hi-hat, snare, and bass percussion sound samples. In *Proc. Audio Mostly Conf*, 2016.

[13] Olivier Gillet and Gaël Richard. Automatic transcription of drum loops. In *Proc. 29th IEEE Intl Conf on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.

[14] Olivier Gillet and Gaël Richard. Enst-drums: an extensive audio-visual database for drum signals processing. In *Proc. 7th Intl Conf on Music Information Retrieval (ISMIR)*, 2006.

[15] Olivier Gillet and Gaël Richard. Supervised and unsupervised sequence modelling for drum transcription. In *Proc. 8th Intl Conf on Music Information Retrieval (ISMIR)*, 2007.

[16] Olivier Gillet and Gaël Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):529–540, 2008.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. `http://arxiv.org/abs/1502.03167`, 2015.

[19] Florian Krebs, Sebastian Böck, Matthias Dorfer, and Gerhard Widmer. Downbeat tracking using beat-synchronous features and recurrent neural networks. In *Proc. 17th Intl Society for Music Information Retrieval Conf (ISMIR)*, 2016.

[20] Florian Krebs, Sebastian Böck, and Gerhard Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proc. 15th Intl Society for Music Information Retrieval Conf (ISMIR)*, 2013.

[21] Florian Krebs, Filip Korzeniowski, Maarten Grachten, and Gerhard Widmer. Unsupervised learning and refinement of rhythmic patterns for beat and downbeat tracking. In *Proc. 22nd European Signal Processing Conf (EUSIPCO)*, 2014.

[22] Marius Miron, Matthew EP Davies, and Fabien Gouyon. Improving the real-time performance of a causal audio drum transcription system. In *Proc. 10th Sound and Music Computing Conf (SMC)*, 2013.

[23] Marius Miron, Matthew EP Davies, and Fabien Gouyon. An open-source drum transcription system for pure data and max msp. In *Proc. 38th IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[24] Arnaud Moreau and Arthur Flexer. Drum transcription in polyphonic music using non-negative matrix factorisation. In *Proc. 8th Intl Conf on Music Information Retrieval (ISMIR)*, 2007.

[25] Jouni Paulus and Anssi Klapuri. Drum sound detection in polyphonic music with hidden markov models. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009.

[26] Geoffroy Peeters and Helene Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1754–1769, 2011.

[27] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *Proc. 31st Intl Conf on Machine Learning (ICML)*, Beijing, China, 2014.

[28] Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. In *Proc. 39th IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[29] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[30] Siddharth Sigtia, Emmanouil Benetos, Srikanth Cherla, Tillman Weyde, Artur S d'Avila Garcez, and Simon Dixon. An RNN-based music language model for improving automatic music transcription. In *Proc. 15th Intl Society for Music Information Retrieval Conf (ISMIR)*, 2014.

[31] Carl Southall, Ryan Stables, and Jason Hockman. Automatic drum transcription using bidirectional recurrent neural networks. In *Proc. 17th Intl Society for Music Information Retrieval Conf (ISMIR)*, 2016.

[32] Srivastava, Nitish and Hinton, Geoffrey and Krizhevsky, Alex and Sutskever, Ilya and Salakhutdinov, Ruslan. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[33] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5rmsprop: Divide the gradient by a running average of its recent magnitude. In *COURSERA: Neural Networks for Machine Learning*, October 2012.

[34] Christian Uhle, Christian Dittmar, and Thomas Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. In *Proc. 4th Intl Symposium on Independent Component Analysis and Blind Signal Separation*, 2003.

[35] Richard Vogl, Matthias Dorfer, and Peter Knees. Recurrent neural networks for drum transcription. In *Proc. 17th Intl Society for Music Information Retrieval Conf (ISMIR)*, 2016.

[36] Richard Vogl, Matthias Dorfer, and Peter Knees. Drum transcription from polyphonic music with recurrent neural networks. In *Proc. 42nd IEEE Intl Conf on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[37] Chih-Wei Wu and Alexander Lerch. Drum transcription using partially fixed non-negative matrix factorization with template adaptation. In *Proc. 16th Intl Society for Music Information Retrieval Conf (ISMIR)*, 2015.

[38] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G Okuno. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):333–345, 2007.

[39] Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen. Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In *Proc. IEEE Conf on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015.