View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by BCU Open Access

AUTOMATIC DRUM TRANSCRIPTION USING BI-DIRECTIONAL RECURRENT NEURAL NETWORKS

Carl Southall, Ryan Stables, Jason Hockman
Digital Media Technology Laboratory (DMT Lab)
Birmingham City University
Birmingham, United Kingdom
{carl.southall, ryan.stables, jason.hockman} @bcu.ac.uk

ABSTRACT

Automatic drum transcription (ADT) systems attempt to generate a symbolic music notation for percussive instruments in audio recordings. Neural networks have already been shown to perform well in fields related to ADT such as source separation and onset detection due to their utilisation of time-series data in classification. We propose the use of neural networks for ADT in order to exploit their ability to capture a complex configuration of features associated with individual or combined drum classes. In this paper we present a bi-directional recurrent neural network for offline detection of percussive onsets from specified drum classes and a recurrent neural network suitable for online operation. In both systems, a separate network is trained to identify onsets for each drum class under observation—that is, kick drum, snare drum, hi-hats, and combinations thereof. We perform four evaluations utilising the IDMT-SMT-Drums and ENST *minus one* datasets, which cover solo percussion and polyphonic audio respectively. The results demonstrate the effectiveness of the presented methods for solo percussion and a capacity for identifying snare drums, which are historically the most difficult drum class to detect.

1. INTRODUCTION

Within the field of music information retrieval, automatic music transcription systems seek to produce a symbolic notation for the instruments in an audio recording. There are a variety of areas in the educational, analytical and creative industries that would benefit from high quality music transcription. To date, the majority of such systems focus on transcription of pitched instruments, with relatively few systems intended for the extraction of drum notation. Automatic drum transcription (ADT) is useful in determining the rhythm and groove inherent in recordings consisting of either drum solos or polyphonic instrument mixtures. While high classification accuracies have been demonstrated for isolated drum hits [9], the task of classification becomes more difficult when multiple different drum instrument hits occur at the same time [10], and is further complicated when other instrumentation is introduced creating a polyphonic mixture.

© Carl Southall, Ryan Stables, Jason Hockman. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Carl Southall, Ryan Stables, Jason Hockman. "Automatic Drum Transcription using Bi-directional Recurrent Neural Networks", 17th International Society for Music Information Retrieval Conference, 2016.

1.1 Background

Using the categorisation presented in [7], the majority of previous ADT systems can be understood as either *segment and classify*, *match and adapt*, or *separate and detect*. Segment and classify methods [2, 6, 17] first divide recordings into regions using either onset detection or a metrical grid derived from beat tracking; second, extract features from the segments; and third perform classification to determine the drum instruments in the segments. Match and adapt methods [21, 22] first associate instruments to predetermined templates then iteratively update the templates to reflect the spectral character of the recording. Separate and detect methods [5, 13, 14, 16] attempt to separate the music signal into the drum sources that make up the mixture prior to identifying the onsets of each source. To date, the most effective separate and detect method for ADT has been non-negative matrix factorisation (NMF), an algorithm that divides a recording into a number of basis functions and corresponding time variant gains. Systems have been proposed for both offline and online applications. Dittmar and Gärtner [3] proposed three types of NMF—fixed, adaptive and semi-adaptive—which can be used in online situations taking each frame as its own NMF instance. For polyphonic audio, Wu and Lerch [20] used harmonic basis functions to separate the drums under observation from the mixtures and improved on standard NMF by introducing new iterative update methods.

In addition to the above methods, ADT systems have been proposed that do not fit in the above categorisation. Paulus incorporated hidden Markov models to identify the probability of drum events based on previous information [15]. Thompson used support vector machines (SVM) with a large dictionary of possible rhythmic configurations to classify automatically detected bars [18].

1.2 Motivation

With the exception of [15, 18] the majority of recent ADT systems rely on single basis functions for each instrument.

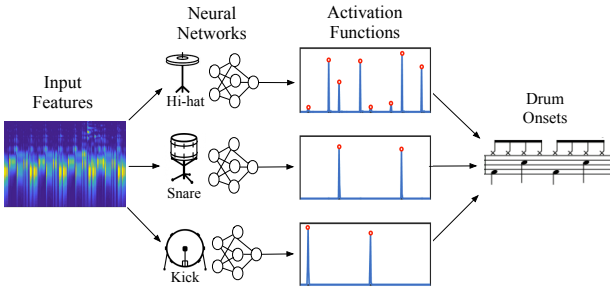

Figure 1: Overview of proposed method. Features are input to individual neural networks for each instrument, resulting in activation functions. Drum onsets are found by peak-picking the activation functions.

This has the potential to overfit to a specific playing technique associated with an individual instrument and fails to recognise more subtle usage. The instrument with the most varied playing techniques in the standard drum kit is the snare drum (e.g., flam, rolls, ghost notes), which not surprisingly is the most difficult to reliably detect. In addition, spectral overlap between basis functions may produce crosstalk between instruments such as snare drums and hi-hats, which can result in noisy time-variant gains, ultimately making peak picking more difficult.

Neural networks are capable of associating complex configurations of features with both individual or combined classes. They have also demonstrated excellent performance in fields related to ADT, such as source separation [8, 11, 12] and onset detection [1, 4]. Other supervised learning techniques such as SVMs have been incorporated in ADT systems [14, 18, 19], however neural networks are capable of capturing the association of class labels with time-series data and producing clean activation functions for the subsequent peak-picking stage. We therefore propose to extend the use of neural networks to ADT in order to exploit their well-known prowess for class separability and their ability to capture the variety of playing techniques associated with each instrument class under observation.

The remainder of this paper is structured as follows: Section 2 outlines our proposed methods for ADT. The evaluation and results are outlined in Section 3 and Section 4 presents a discussion of these results. Conclusions and possible future work are provided in Section 5.

2. METHOD

An overview of our proposed method for ADT is presented in Figure 1. For each percussive instrument under observation, features obtained from the audio recording are input into the pre-trained neural networks iteratively by frame. We then select the peaks from the resulting activation functions to determine the location of onsets for the corresponding instruments.

2.1 Neural Networks

Recurrent neural networks (RNN) incorporate information from previous time steps that allow for temporal information to be understood. Bi-directional neural networks (BDRNN) include information from future time steps by combining two RNNs: the first is a standard backwards directional RNN that incorporates present and previous time information; the second RNN is instead positioned to incorporate information from present and future time positions, achieved by reversing the order of the input time steps. As BDRNNs are unsuitable for online applications, we propose two separate models: an RNN for online usage and a BDRNN for applications that can operate offline. An overview of both neural networks is given in Figure 2.

2.1.1 Recurrent Neural Network

The RNN architecture is represented in Figure 2 by the solid lines. For an RNN with L layers, the equation for each layer l is:

$$a_0^l(t) = f_l(a_0^{l-1}(t)W_0^l + \beta(a_0^l(t-1)U_0^l) + b_0^l), \quad (1)$$

where $\beta = 0$ for $l = L$, and 1 otherwise. With layer l output a, the weight matrices W and U and the bias matrices b. The transfer function is determined by the layer, and is defined as:

$$f_l(x) = \begin{cases} 2/(1+e^{-2x}) - 1, & l \neq L \\ y = e^x/(\sum e^x), & l = L. \end{cases} \quad (2)$$

2.1.2 Bi-directional Recurrent Neural Network

The additional BDRNN connections are represented by dashed lines in Figure 2. For a BDRNN with L layers, the equation for each hidden layer l is:

$$a_n^l(t) = f_l(a_n^{l-1}(t)W_n^l + a_n^l(t-1)U_n^l + a_{(1-n)}^{l-1}Z_{(1-n)}^{l-1} + b_n^l) \quad (3)$$

where the layer is defined as forward directional when $n = 0$ and backwards directional when $n = 1$. Z is an additional weight matrix. The output layer for time t can then be defined as:

$$a_0^L(t) = f_L(a_0^{L-1}(t)W_0^L + a_1^{L-1}(t)W_1^L + b^L) \quad (4)$$

2.2 Input Features

Following the approach in [20], input audio (mono .wav files sampled at 44.1 kHz with 16-bit resolution) is transformed into a 1024 x n spectrogram representation using the short-time Fourier transform (STFT), in which n is the numbers of frames. The STFT is calculated using a Hanning window with a window length of 2048 samples and a hop size of 512 samples.

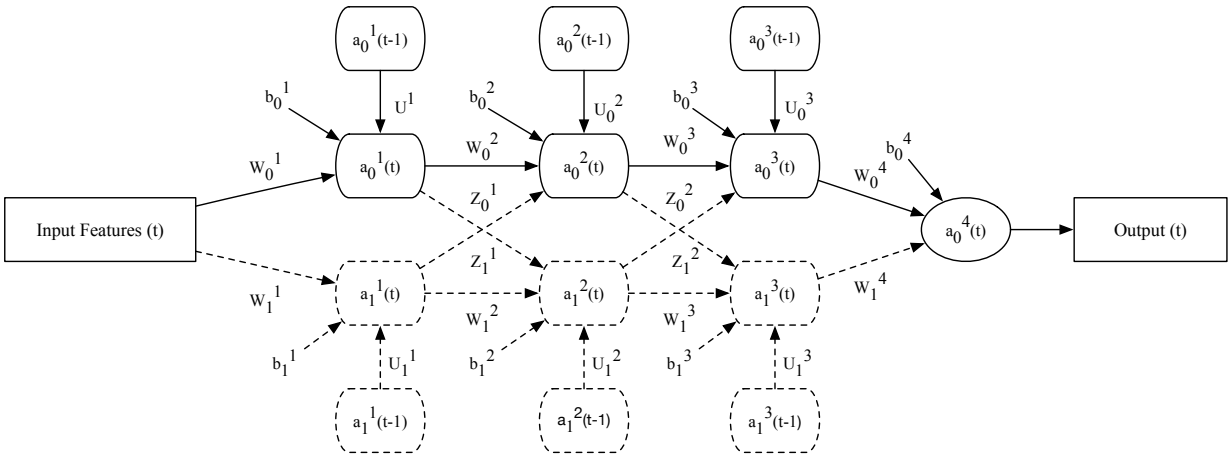

Figure 2: Overview of the proposed bi-directional recurrent neural network (BDRNN) and recurrent neural network (RNN). Solid lines represent the RNN connections and dashed lines are additional BDRNN connections. Tan sigmoid layers are shown as curved rectangles and soft max layers are represented by circles. The weight matrices are denoted as W, U and Z and the biases as b. The output of layer two of the backwards directional recurrent neural network is denoted by a_1^2.

2.3 Architecture

The neural network architectures in each instrument are identical, consisting of three dense hidden layers of 50 neurons. This configuration was chosen as it achieved the highest results in preliminary testing. The neural networks are trained using target activation function representations created from training data annotations. The first row of the activation function frames in which onsets occur are set to one and all other frames are set to zero. The networks are trained with a learning rate of 0.05 using truncated back propagation through time, which updates the weights and biases iteratively using the output errors. A maximum iteration limit is set to 1000, and the weights and biases are initialised to random non-zero values between ± 1 ensuring that training commences correctly. To prevent overfitting, a validation set is created from 10% of the training data. If no improvement is demonstrated on the validation set over 100 iterations, then training is stopped. The performance measure used is cross entropy combined with a softmax output layer, as this proved to be the most effective configuration.

2.4 Onset Detection

Once a drum activation function θ has been generated for each drum class under observation, the onsets must be temporally located from within θ. We adopt the method from [4] for onset detection in the BDRNN. To calculate onset positions, a threshold is first determined using the mean of all frames and a constant λ:

$$T = mean(\theta) * \lambda. \tag{5}$$

If the current frame n is determined to be both a peak and above the threshold T then it is accepted as an onset Γ:

$$\Gamma(n) = \begin{cases} 1, & \theta(n-1) < \theta(n) \geq \theta(n+1) \;\&\; T < \theta(n) \\ 0, & otherwise. \end{cases} \tag{6}$$

For online applications using the RNN, where future information can not be used within the peak picking process, the threshold is determined by taking the mean of the current frame and the previous ρ frames with an onset being accepted if the current frame is greater than the threshold and the previous frame. We selected $\rho = 9$ after initial informal testing. Due to the iterative classification of each frame, onsets may be detected in adjacent frames. We therefore disregard onsets detected within 50 ms of each other to ensure false positives are not obtained for a drum event that has already been detected.

3. EVALUATION

We conduct four evaluations intended to test the presented systems in a variety of different contexts in which an ADT system could be used. The first evaluation, termed *automatic*, aims to demonstrate system performance on drum solo recordings in a general purpose way where no prior information about the test track is known. Following [3], the second evaluation allows information from the test tracks to be used to aid in transcription of drum solo recordings in a *semi-automatic* manner. This scenario could be used for compositional or educational purposes either for identifying an arrangement of a specified drum solo that has been resequenced in another recording, or in a studio situation in which a single drum kit is being used. The third and fourth evaluations aim to evaluate the systems in polyphonic mixtures, where instruments other than the drums under observation are found. Mixtures containing other drums (e.g., floor toms, ride cymbal) are used in the third evaluation and additional harmonic accompaniment (e.g., guitars, keyboards) is found in the fourth. These evalua-

	Precision				Recall				F-measure			
	kick	*snare*	*hi-hat*	*mean*	*kick*	*snare*	*hi-hat*	*mean*	*kick*	*snare*	*hi-hat*	*mean*
BDRNN	0.912	0.834	0.795	**0.847**	0.929	**0.901**	0.729	0.853	0.909	**0.852**	0.738	**0.833**
RNN	0.890	**0.856**	0.741	0.829	0.909	0.851	**0.788**	0.849	0.884	0.833	0.729	0.816
PFNMF	0.934	0.633	**0.939**	0.835	0.931	0.889	0.743	0.854	0.926	0.699	**0.811**	0.812
AM1	0.934	0.633	**0.939**	0.835	0.931	0.889	0.743	0.854	0.926	0.699	**0.811**	0.812
AM2	**0.937**	0.651	0.893	0.827	**0.934**	0.886	0.786	**0.868**	**0.929**	0.713	0.805	0.816

Table 1: Precision, recall and F-measure results for the automatic evaluation using the IDMT-SMT-Drums dataset, including the PFNMF, AM1 and AM2 systems and the proposed BDRNN and RNN systems. The highest accuracy achieved in each of the categories is highlighted bold.

tions are termed *percussive mixtures* and *multi-instrument mixtures* respectively.

3.1 Evaluation Methodology

Standard precision, recall, and F-measure scores are used to measure system performance. Precision and recall are determined from detected drum instrument onsets, with candidate onsets determined as correct if found within 50 ms of annotations. Only kick drum, snare drum and hi-hat onsets are taken into consideration. The mean F-measure is calculated by taking the average of the individual instrument F-measures. We set the λ parameter used in neural network peak picking using grid-search across the dataset.

3.1.1 Automatic Evaluation

To test the generalisation of the proposed system, we undertake the automatic evaluation, using the IDMT-SMT-Drums dataset [3]. This dataset consists of 95 tracks (14 *real drum* tracks, 11 *techno drum* tracks, and 70 *wave drum* tracks) with individual kick drum, snare drum and hi-hat recordings. The average track length is 15 seconds, and in total there are 3471 onsets. Using three-fold cross validation the dataset is split into training and testing data, resulting in approximately 89,000 and 37,000 frames, respectively. Mean precision, recall and F-measure scores are taken across tested folds for each system under evaluation. The proposed neural network systems are evaluated alongside the three methods proposed in [20]: PFNMF, which uses fixed percussive basis functions in conjunction with harmonic basis functions within a NMF framework; AM1, which iteratively updates the percussive basis functions of PFNMF; and AM2, which updates the PFNMF basis functions and activation functions in an alternating fashion. Each of the NMF systems are initialised by taking the mean of each of the basis functions derived from the individual tracks.

3.1.2 Semi-automatic Evaluation

In order to test the systems ability to adapt to a specific situation, we undertake the semi-automatic evaluation. We again utilise the IDMT-SMT-Drums dataset, however in this context we provide the systems exclusively with individual drum hits that are used in the overall track under analysis. For a performance comparison in this evaluation, we also test the worth of training the neural networks using mixed drum hits (e.g., kick drum and hi-hat played together). The proposed methods are evaluated alongside the semi-adaptive online NMF technique CD as presented in [3]. As the evaluation procedures herein are identical to those in [3] the results from this work have been incorporated for comparison.

3.1.3 Percussion and Multi-instrument Evaluations

To test how well the proposed system can identify drums within various types of mixtures, we perform the percussion and multi-instrument evaluations, using the same procedure as in the automatic evaluation. For these evaluations, we use the ENST *minus one* dataset as it contains drum tracks with additional drum instruments (e.g., floor tom, ride cymbal) and techniques (e.g., ghost notes, flams, rolls) as well as accompaniment tracks. The ENST minus one dataset contains 64 recordings performed by three drummers; two drummers performed 21 tracks each and the third drummer performed 22 tracks. The BDRNN and RNN are provided recordings of two of the drummers as training, while testing on the third. The average track length is 55 seconds with a total of 22,410 kick drum, snare drum and hi-hat onsets, resulting in 210,000 and 105,000 frames for training and testing respectively in each fold. We mix the accompaniment and drum recordings in the dataset using the same ratios ($\frac{1}{3}$ and $\frac{2}{3}$, respectively) as in [7, 15, 20]. The evaluation procedures in these two evaluations are identical to those in [7, 15, 20], and as such the results from these studies have been used for comparison herein.

Method	*Mean F-measure*
RNN (individual drums)	0.634
BDRNN (individual drums)	0.700
RNN (mixed drums)	0.955
BDRNN (mixed drums)	**0.961**
CD	0.950

Table 2: Mean F-measure results for the semi-automatic evaluation. BDRNN and RNN systems are trained on individual drum hits (individual drums) or mixtures of drum hits (mixed drums) and are compared with that of the CD method in [3].

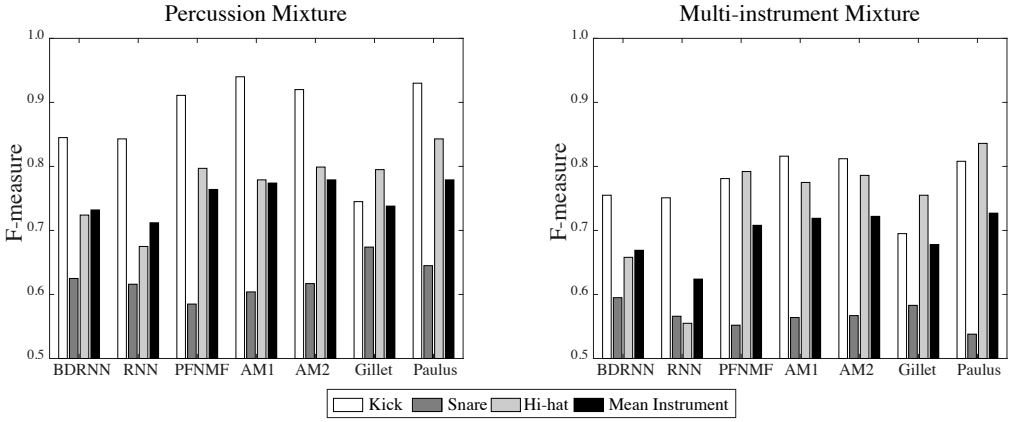

Figure 3: Kick drum, snare drum, hi-hat and mean instrument F-measure results for the BDRNN and RNN. Results for *percussion mixture* (left) and *multi-instrument mixture* (right) evaluation scenarios are compared to those obtained in [7, 15, 20].

3.2 Results

3.2.1 Automatic Results

The proposed BDRNN achieved a higher mean instrument F-measure than existing ADT methods in the first evaluation, which focuses on drum solos. Table 1 demonstrates that the neural network approaches achieved the highest scores in six of the twelve categories, with the largest relative improvement being the snare F-measure and precision. As expected, the RNN achieved lower F-measure scores than the BDRNN for all instruments, however the results matched those of the other evaluated systems. The three methods proposed in [20] achieved similar results as previously obtained on the IDMT-SMT-Drums dataset. Initial tests revealed that the best performance for all three algorithms was achieved with the rank parameter set to 1 and an offset coefficient of 0.2. AM1 showed no improvement on PFNMF in this instance, however AM2 did slightly improve.

3.2.2 Semi-Automatic Results

Table 2 shows the results of the semi-automatic evaluation scenario with both systems compared to the results of the CD system obtained in [3]. Training the neural networks using individual drum hits alone resulted in low accuracies, however when training includes mixed drum instrument signals (e.g., kick drums and hi-hats playing at the same time) both the BDRNN and RNN achieve the highest results of the tested systems.

3.2.3 Percussion and Multi-Instrument Mixture Results

Figure 3 shows the results of the BDRNN and RNN methods as compared to those achieved by the Gillet [7] and Paulus [15] systems, as well as the results of the PFNMF, AM1, and AM2 systems in [20]. The results are shown for both scenarios: percussive mixtures (left figure) and multi-instrument mixtures (right figure). In both evaluations, the neural network approaches achieve high snare F-measures relative to the other systems, and the BDRNN achieves the highest snare F-measure for the multi-instrument mixture evaluation. Figure 4 shows the mean precision and recall scores of the neural network systems in comparison to the other evaluated systems. The highest recall scores are achieved by the BDRNN and RNN for both percussion and multi-instrument mixtures. While the neural networks achieved lower mean F-measure scores, this high recall demonstrates the potential worth of the clean activation functions.

4. DISCUSSION

The results show that the proposed neural network systems achieve higher results for a solo drum dataset in offline and online situations in both automatic and and semi-automatic evaluations. The offline bi-directional recurrent neural network architecture outperformed the online recurrent neural network architecture in all evaluations, demonstrating the worth of additional future information for applications that allow it. The high results for the snare drum class achieved throughout the evaluation indicate the ability of the neural networks to associate multiple different frequency bases to the same class making them well suited to detect a variety

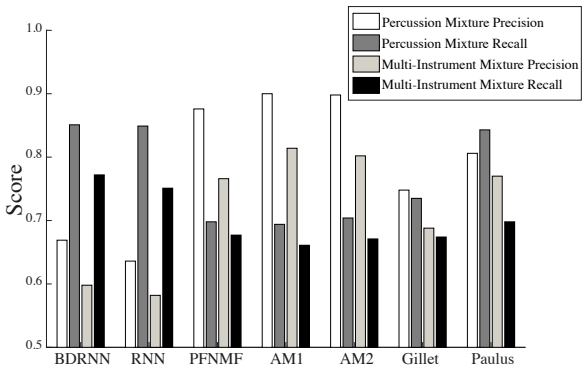

Figure 4: Mean Precision and Recall results from the percussion and multi-instrument mixture evaluations.

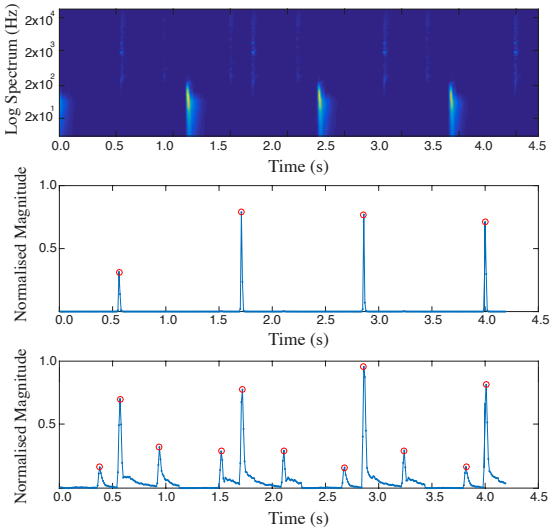

Figure 5: Comparison between neural network activation function and PFNMF time-variant gain: (*top*) Mixed spectrogram containing kick drum, snare drum and hi-hat; (*middle*) BDRNN snare drum activation function and found onsets; (*bottom*) PFNMF snare drum time-variant gain and found onsets.

of playing techniques for a given instrument (e.g., flams, rolls, ghost notes). An example of the instrument-specific focus achievable by neural networks is shown in Figure 5, where spectral overlap exists between the snare drum and hi-hats. While the PFNMF output for this particular example shows the effect of crosstalk, the BDRNN is able to achieve a less noisy activation function.

Although high F-measures for the kick drum and hi-hat were achieved by both the RNN and BDRNN methods, they had lower scores than other techniques for all situations other than the semi-automatic drum transcription test. The precision scores for the kick drum and hi-hat were the main factor in the lower F-measure score. As shown in Table 1 and Figure 4, the BDRNN and the RNN achieved the highest mean recall scores for all tests using the ENST *minus one* dataset, which indicates that the methods benefit from a simplified peak picking process due to clean activation functions. However, F-measures scores for these tests indicate that the BDRNN and RNN were not as successful as other systems—a somewhat expected result as this dataset contains polyphonic mixtures. The addition of a pre-processing stage similar to [20] could remove these sources prior to ADT and potentially improve results for the BDRNN and RNN methods. Another area for possible improvement would be to evaluate the worth of different input features such as MFCCs, which have already been demonstrated to be successful in conjunction with neural networks in the related task of onset detection [1].

5. CONCLUSIONS AND FUTURE WORK

We have presented two neural network based approaches for ADT: the BDRNN for off-line usage and the RNN online applications. Results from the conducted evaluations demonstrate that the proposed methods are capable of outperforming existing ADT systems on drum solo recordings in both automatic and semi-automatic situations. The ability to learn a rich representation of drum classes enables the neural networks to detect multiple playing techniques within the same class. Evaluations were also carried out on polyphonic mixtures in which the neural network achieved high snare F-measures relative to existing approaches. To improve performance of the proposed methods for polyphonic audio, an additional pre-processing source separation stage could be introduced into the system to separate the desired drums from additional instrumentation prior to ADT. Furthermore, additional time step connections to additional previous and future time steps may potentially increase the accuracy of the system. One method for doing this is by using long short-term memory cells within the neural network architecture which have already proven to be effective for onset detection [4]. Further evaluation will be carried out to determine performance when additional drum classes are present, as well as testing other input features.

6. ACKNOWLEDGEMENTS

The authors would like to thank Chih-Wei Wu and Christian Dittmar for their help in providing code for implementations of their methods, as well as the reviewers for their valuable feedback in this paper.

7. REFERENCES

[1] S. Böck, A. Arzt, F. Krebs, and M. Schedl. Online real-time onset detection with recurrent neural networks. In *Proc. of the 15th International Conference on Digital Audio Effects (DAFx-12)*, 2012.

[2] C. Dittmar. Drum detection from polyphonic audio via detailed analysis of the time frequency domain. In *Proc. of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.

[3] C. Dittmar and D. Gärtner. Real-time transcription and separation of drum recordings based on nmf decomposition. In *Proc. of the 17th International Conference on Digital Audio Effects (DAFX)*, 2014.

[4] F. Eyben, S. Böck, B. Schuller, and A. Graves. Universal onset detection with bidirectional long short-term memory neural networks. In *Proc. of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 589–594, 2010.

[5] D. Fitzgerald. *Automatic drum transcription and source separation*. PhD thesis, Dublin Institute of Technology, 2004.

[6] O. Gillet and G. Richard. Automatic transcription of drum loops. In *Proc. of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 269–272, 2004.

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

[8] E. M. Grais, M. U. Sen, and H. Erdogan. Deep neural networks for single channel source separation. In *Proc. of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3734–3738, 2014.

[9] P. Herrera, A. Yeterian, and F. Gouyon. Automatic classification of drum sounds: A comparison of feature selection methods and classification techniques. In *Proc. of the International Conference on Music and Artificial Intelligence*, pages 69–80, 2002.

[10] P. Herrera-Boyer, G. Peeters, and S. Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21, 2003.

[11] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep learning for monaural speech separation. In *Proc. of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1562–1566, 2014.

[12] J. Le Roux, J. R. Hershey, and F. Weninger. Deep NMF for speech separation. In *Proc. of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 66–70, 2015.

[13] H. Lindsay-Smith, S. McDonald, and M. Sandler. Drumkit transcription via convolutive NMF. In *Proc. of the 15th International Conference on Digital Audio Effects (DAFx)*, 2012.

[14] A. Moreau and A. Flexer. Drum transcription in polyphonic music using non-negative matrix factorization. pages 353–354, 2007.

[15] J. Paulus and A. Klapuri. Drum sound detection in polyphonic music with hidden Markov models. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009(1):1–9, 2009.

[16] J. Paulus and T. Virtanen. Drum transcription with non-negative spectrogram factorization. In *Proc. of the 13th European Signal Processing Conference (EUSIPCO)*, pages 1–4, 2005.

[17] K. Tanghe, S. Degroeve, and B. De Baets. An algorithm for detecting and labeling drum events in polyphonic music. pages 11–15, 2005.

[18] L. Thompson, M. Mauch, and S. Dixon. Drum transcription via classification of bar-level rhythmic patterns. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, pages 187–192, 2014.

[19] D. van Steelant and K. Tanghe. Classification of percussive sounds using support vector machines. In *Proc. of the 2004 Machine Learning Conference of Belgium and The Netherlands*, pages 146–152, 2004.

[20] C.-W. Wu and A. Lerch. Drum transcription using partially fixed non-negative matrix factorization with template adaptation. In *Proc. of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pages 257–263, 2015.

[21] K. Yoshii, M. Goto, and H. G. Okuno. Automatic drum sound description for eeal-world music using template adaptation and matching methods. In *Proc. of the 5th International Conference on Music Information Retrieval*, pages 184–191, 2004.

[22] K. Yoshii, M. Goto, and H. 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.