# AUTOMATIC DRUM TRANSCRIPTION FOR POLYPHONIC RECORDINGS USING SOFT ATTENTION MECHANISMS AND CONVOLUTIONAL NEURAL NETWORKS

**Carl Southall, Ryan Stables and Jason Hockman**
DMT Lab, Birmingham City University
Birmingham, United Kingdom
{carl.southall, ryan.stables, jason.hockman} @bcu.ac.uk

## ABSTRACT

Automatic drum transcription is the process of generating symbolic notation for percussion instruments within audio recordings. To date, recurrent neural network (RNN) systems have achieved the highest evaluation accuracies for both drum solo and polyphonic recordings, however the accuracies within a polyphonic context still remain relatively low. To improve accuracy for polyphonic recordings, we present two approaches to the ADT problem: First, to capture the dynamism of features in multiple time-step hidden layers, we propose the use of soft attention mechanisms (SA) and an alternative RNN configuration containing additional peripheral connections (PC). Second, to capture these same trends at the input level, we propose the use of a convolutional neural network (CNN), which uses a larger set of time-step features. In addition, we propose the use of a bidirectional recurrent neural network (BRNN) in the peak-picking stage. The proposed systems are evaluated along with two state-of-the-art ADT systems in five evaluation scenarios, including a newly-proposed evaluation methodology designed to assess the generalisability of ADT systems. The results indicate that all of the newly proposed systems achieve higher accuracies than the state-of-the-art RNN systems for polyphonic recordings and that the additional BRNN peak-picking stage offers slight improvement in certain contexts.

## 1. INTRODUCTION

Music notation, which portrays the instrumentation and playing techniques used within a musical recording, is produced through the process of automatic music transcription (AMT). Fast and accurate production of music notation would benefit multiple areas including the creative, analytical and educational industries. The majority of previous AMT systems has been developed to address pitched instrumentation, while relatively few systems have focussed on the transcription of percussive instruments. Automatic drum transcription (ADT) systems soley focus on producing notation for drum instruments, which strongly portray the rhythm, groove and feel of the piece. High ADT accuracies have been achieved on audio recordings containing only basic drum classes such as kick drum, snare drum and hi-hats [15, 19]. However, accuracies are significantly lower in a *polyphonic context*—in which the recordings contain either additional percussion (e.g., toms, cymbals) or pitched instrumentation (e.g., guitar, piano) [20].

### 1.1 Background

Several early ADT systems have been proposed that perform well on solo drum recordings [3, 5, 10, 13, 18, 23], however a relatively small number of systems have demonstrated the capacity for high performance in a polyphonic context. Wu and Lerch [21] proposed a non-negative matrix factorisation technique with a specialised basis function to capture harmonic activity outside of those for the drum classes under observation. Paulus et al. [12] used a hidden Markov model to detect the presence of individual drum onsets within frames of a spectrogram. Southall et al. [15] and Vogl et al. [19] also formalise ADT as a frame-wise drum onset detection problem, using recurrent neural networks (RNN) for classification. Southall et al. [15] presented a bidirectional RNN (BRNN) system and Vogl et al. [19] presented a RNN system with time-shifted classification labels. RNN systems have achieved the best drum solo performance to date, however their accuracies in the polyphonic context has been marginalised. Vogl et al. [20] later proposed the incorporation of gated recurrent unit (GRU) cells, which incorporate more time-step information into the RNN model, resulting in the highest ADT accuracies to date in a polyphonic context.

### 1.2 Motivation

The increase in accuracy achieved by the GRU RNN in [20] over the standard RNN in [19] demonstrates the effect of storing additional information on classification performance. In a solo drum context, instrumentation overlap is limited to the drums under observation, whereas in a polyphonic context, drums are present along with other instruments. This may obscure the presence of features belonging to the drums under observation, and is mitigated by the incorporation of additional time-step information in the

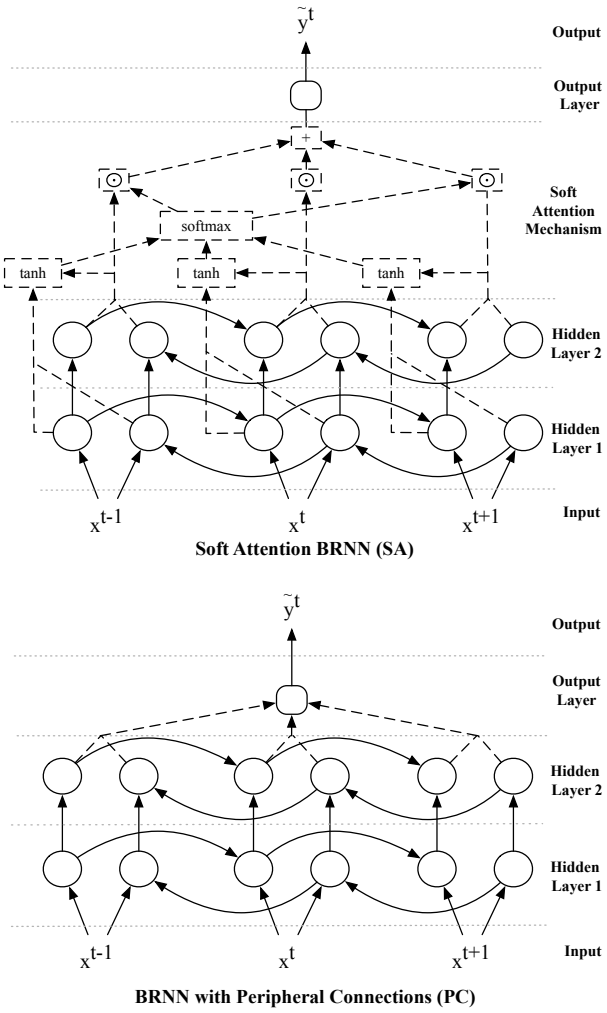


**Figure 1**: Overview of the proposed SA and PC systems. Solid lines depict connections of a standard BRNN configuration and dashed lines depict additional SA and PC connections when attention number $a$ = 1. $x^t$ and $\tilde{y}$ are input features and output activation function at time step $t$. $\odot$ in the SA system represents element-wise multiplication.

GRU RNN. Inclusion of additional information in previous RNN ADT systems however, is still restricted by a bottleneck at the output layer, which is determined by the hidden state sizes. Additionally, larger input feature sizes can not be used at each time step, due to the computational cost of fully-connected layers. We present two approaches in an attempt to overcome the above-stated limitations to ADT in a polyphonic context: First, to capture the dynamism of features in multiple time-step hidden layers, we propose the use of soft attention mechanisms (SA) and an alternative RNN configuration containing additional peripheral connections (PC). Second, to capture these same trends at the input level, we propose the use of a convolutional neural network (CNN), which uses a larger set of time-step features. To further improve the accuracy of the systems, we also propose the use of an additional BRNN for selecting drum onsets from the output activation functions, as peak-picking within a polyphonic context has proven to be more difficult than that of drum solos [15, 19, 20].

The remainder of this paper is structured as follows: Section 2 presents our three newly proposed systems and our new peak-picking technique. The evaluation is outlined in Section 3 and the results are presented in Section 4. Conclusions and future work are provided in Section 5.

## 2. METHOD

For the three new proposed systems, we use the same frame-wise classification ADT technique outlined in [15]. Input features are fed into a separate pre-trained neural network for each instrument under observation. Peak-picking is then performed on the resulting activation functions to determine onset locations.

### 2.1 Soft Attention BRNN (SA)

Attention mechanisms allow the network to focus on different parts of the data stored within a RNN for different tasks. This is achieved by enabling the information fed to the output layer to be created from multiple time-step final hidden layers. This was initially achieved through binary connections in hard attention mechanisms and then by weighted connections in soft attention mechanisms (SA). They have improved RNN results in multiple fields including: machine translation [1] and image caption generation [11, 22]. An overview of the implemented SA ADT system based on [6] is given at the top of Figure 1. We use a BRNN with each hidden layer containing 100 long short-term memory cells with peephole connections (LSTMP) as the basis of the system. This is due to its ability to pass information through its memory cell $c$, which is updated using the input $i$, forget $f$ and output $o$ gates. The equations for a LSTMP cell layer are:

$$i_l^t = \sigma(Wi_l\left[x^t, h_l^{t-1}, c_l^{t-1}\right] + bi_l) \quad (1)$$

$$f_l^t = \sigma(Wf_l\left[x^t, h_l^{t-1}, c_l^{t-1}\right] + bf_l) \quad (2)$$

$$\tilde{c_l^t} = tanh(Wc_l\left[x^t, h_l^{t-1}, c_l^{t-1}\right]) \quad (3)$$

$$c_l^t = f^t \odot c_l^{t-1} + i^t \odot \tilde{c_l^t} + bc_l) \quad (4)$$

$$o_l^t = \sigma(Wo_l\left[x^t, h_l^{t-1}, c_l^t\right] + bo_l) \quad (5)$$

$$h_l^t = o_l^t \odot tanh(c_l^t), \quad (6)$$

where $h_l^t$ is the hidden layer of layer $l$ at time step $t$, the weights $W$, and the biases $b$. $x$ is the input feature where $x^t = h_{l-1}^t$ if $l > 1$. After each hidden layer dropouts [16] are implemented with a probability of $p$. Based on preliminary tests, we use 2 hidden layers as using more did not improve performance.

The SA feeds the LSTM BRNN output into the output layer as a weighted combination of $2a + 1$ time-step final hidden layers, centred on the current time-step $t$, where $a$ is the attention number. First, an intermediate variable $m$ is determined for each attention step $i$ ($i = t-a : t+a$) using the concatenated outputs of the forwards and backwards directional LSTMs $Q$ ($Q = \left[y_L^{\rightarrow}, y_L^{\leftarrow}\right]$) and a context $U$:

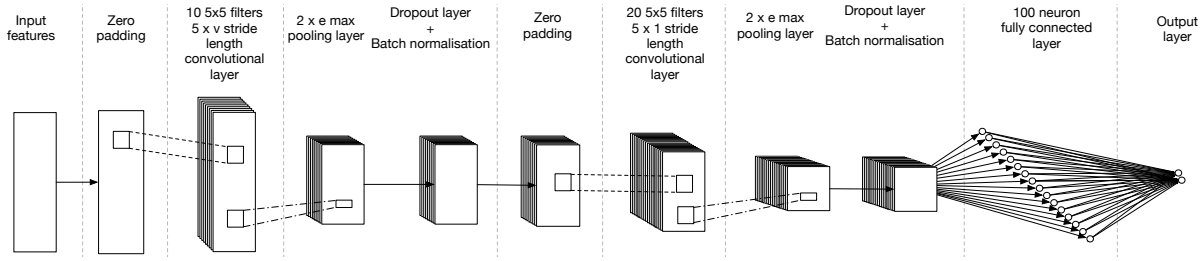


**Figure 2**: Overview of the proposed CNN. Information flows through the network from left to right; solid lines represent connections, with dashed lines representing convolution and dash-dotted lines representing max pooling.

$$m^i = tanh(W_q Q^i + W_U U). \tag{7}$$

The aim of $U$ is to feed the SA mechanism information regarding the wider scope of the current data. We first attempted to use the the cell state of the final hidden layer $c_L$ as in [8], however using the outputs of the first hidden layer ($U = \left[h_1^{t\rightarrow}, h_1^{t\leftarrow}\right]$) resulted in better performance during preliminary testing. The attention weights $s$ are determined using a softmax function across $i$:

$$s^i \propto \exp {W_m}^T m^i, \tag{8}$$

so that $\sum_i s^i = 1$. The output layer input $z$ is then calculated using $Q$ and $s$ and fed into an output layer similar to the BRNN architecture in [15]:

$$z = \sum_i s^i \odot Q^i \tag{9}$$

$$\tilde{y}^t = softmax(W_z z + b_z). \tag{10}$$

$s$ can be thought of as percentage determining how much information from each of the time-step final hidden layers $Q$ is used in the input to the output layer $z$.

### 2.2 BRNN with Peripheral Connections (PC)

Although the SA system allows the information fed into the output layer to be determined directly from multiple time-step hidden layers, the amount of information is still limited by the hidden layer size. We propose an increase in the amount of information passed to the output layer by including direct connections from multiple time-step hidden layers to the output layer, which we term *peripheral connections* (PC). An overview of the PC system is presented at the bottom of Figure 1. The PC system is the same as the SA system in eqns. 1–6. However, these connections are implemented in the output layer using:

$$\tilde{y}^t = softmax(W_v Q^{t-a:t+a} + b_v), \tag{11}$$

where $v$ highlights the weights and biases belonging to the PC output layer and $Q^{t-a:t-a}$ is the concatenation of multiple LSTM time-step outputs:

$$Q = \left[h_L^{\rightarrow t-a}, .... , h_L^{\rightarrow t+a}, h_L^{\leftarrow t-a}, .... , h_L^{\leftarrow t+a}\right]. \tag{12}$$

If $a = 0$, then both the SA and PC systems are the same as a standard BRNN network with LSTMP cells.

### 2.3 Convolutional Neural Network (CNN)

As RNNs contain fully-connected layers, large input feature sizes can not be used as they become extremely computationally expensive. Convolutional neural networks (CNN) overcome this problem by combining feature learning, dimensionality reduction and classification stages in a single trainable network. This ability has enabled CNNs to achieve higher accuracies than RNNs in the closely related fields of onset detection [14] and downbeat detection [4]. We propose to use a convolutional neural network to enable multiple time-step features to be used as input for each frame classification. An overview of the implemented CNN ADT system is outlined in Figure 2 where $j$ frames on either side of the current frame $t$ are included in the input features and different values of $v$ and $e$ are used as $j$ is increased. It consists of two sets of convolutional, max pooling, dropout [16], and batch normalisation [7] layers before a 100-neuron fully-connected layer and a two neuron softmax output layer.

### 2.4 Implementation

The newly proposed models are implemented using the Tensorflow Python library. Four SA and PC systems (`SA1`, `SA2`, `SA3` and `SA5`) and (`PC1`, `PC2`, `PC3` and `PC5`) are implemented where $a$ = [1, 2, 3, 5] and four CNN systems (`CNN2`, `CNN5`, `CNN10`, and `CNN20`) are implemented where $j$ = [2, 5, 10, 20]. These values are chosen as they cover various ranges of important information regarding the typical envelope length of drums.

#### *2.4.1 Input Features*

In order for an audio file to be processed by the neural networks, it must be procedurally segmented into frame-wise spectral features. First, the input audio (16-bit .wav file sampled at 44100 kHz) is segmented into $T$ frames using a Hanning window of $n$ samples ($n = 2048$) with a $\frac{n}{4}$ hopsize. A frequency representation of each of the frames is then created using the magnitudes of a discrete Fourier transform resulting in a $\frac{n}{2}$ x $T$ spectrogram. The spectrogram is input into the SA systems in a frame-wise manner and as a combination of frames ($j$ frames either side of the current frame $t$) for the CNN systems.

#### *2.4.2 Peak Picking*

Once the activation functions $\tilde{Y}$ are output from the systems, peak-picking is used to identify the onset candidates.

In this paper, we implement two peak-picking strategies for each of the systems. The first approach, termed mean threshold (MT), is an updated version of the technique used in [15], in which a threshold is determined for each frame ($\tau^t$) using:

$$\tau^t = mean(\tilde{y}^{t-\theta} : \tilde{y}^{t+\theta}) * \lambda \tag{13}$$

$$\tau^t = \begin{cases} tmax, & \tau > tmax \\ tmin, & \tau < tmin, \end{cases} \tag{14}$$

where $\theta$ sets the number of frames in each direction to calculate the mean, $\lambda$ is a constant and $tmax$ and $tmin$ are the possible maximum and minimum values. The current frame of $\tilde{y}$ is accepted as an onset if it is the maximum of a surrounding number of frames and above the threshold $\tau$:

$$O^t = \begin{cases} 1, & \tilde{y}^t == max(\tilde{y}^{t-\Omega} : \tilde{y}^{t+\Omega}) \quad \& \quad \tilde{y}^t > \tau^t \\ 0, & otherwise, \end{cases} \tag{15}$$

where $O(t)$ represents an onset at time step $t$ and $\Omega$ is the number of frames on either side of the current frame $t$ used to calculate the maximum.

For the second approach we train an additional neural network using the activation functions from the training data in an attempt to learn to identify the drum onsets more difficult to detect. To do this we use a BRNN, with a single 10 LSTMP-cell hidden layer and a softmax output layer. The output of the new BRNN is then processed by the MT technique (eqns. 10–12), we refer to this second technique as BRNN-MT.

#### *2.4.3 Training*

The three models and the BRNN-MT peak-picking networks are trained using the Adam optimiser [9] with a learning rate of 0.003. The training data is created by generating a feature matrix from input features $x$ and an associated class vector from the target activation functions $Y$. Mini-batch gradient descent (batch size = 1000) created from 10 segments (segment length = 100) is used. The activation function output from the models $\tilde{Y}$ are used as the input to the BRNN-MT networks which are trained using the same targets used to train the systems. A new BRNN-MT network is trained independently for each system in an attempt to increase adaptability, similar to [2]. Training is stopped when the following criteria have been met: (1) a minimum of 10 epochs have commenced; and (2) the validation set accuracy has not increased between epochs. To ensure training commences correctly, the weights are initialised using a random uniform distribution scaled to keep constant variance [17] and the biases are initialised to zero. Cross entropy is used as the loss function.

## 3. EVALUATION

To evaluate the newly proposed methods along with the current state-of-the-art systems, we implement four evaluations similar to those carried out in [15, 19, 20], along with an additional evaluation to test the generalisability of the systems. The systems are trained to identify kick drum, snare drum and hi-hat onsets. The first evaluation, termed *drum solo*, aims to demonstrate system performance on drum solo recordings that contain only the three drum instruments under observation. The second evaluation, termed *drum mixture*, aims to demonstrate system performance in a drum-only polyphonic context, where the recordings contain additional drum instrumentation to those under observation (e.g., toms and cymbals). The third evaluation, termed *multi-instrument mixture*, aims to demonstrate system performance in a fully-polyphonic context where multiple instruments are present in addition to the drum instruments under observation (e.g., piano and guitar) and the fourth evaluation, termed *cross-context*, aims to test the systems adaptability to before unseen timbres. The newly proposed evaluation, termed *multi-context*, aims to test the ability of a single system to be trained and used in multiple contexts.

### 3.1 Evaluation Methodology

F-measure is used as the evaluation metric with precision and recall determined using the onset candidates from the peak-picking stage. Detected onsets are accepted as true positives if they fall within 50ms of the ground truth annotations. The individual instrument F-measures are calculated as the mean F-measure across test tracks and the mean instrument F-measure is calculated as the mean F-measure across the individual instruments. The peak-picking parameters ($\theta$, $\lambda$, $tmax$, $tmin$ and $\Omega$) are found using a grid-search on the validation set and the dropout probability $p$ is set to 0.25.

#### *3.1.1 Drum Solo Evaluation*

To test the capability of the systems in the *drum solo* evaluation, we use the updated version of the IDMT-SMT-Drums dataset [3]. This dataset contains 104 tracks divided into three subsets (20 *real drum* tracks, 14 *techno drum* tracks, and 70 *wave drum* tracks) with an average track length of 15 seconds. The dataset is divided by track in equal distributions across the three subsets into 70% training 15% validation and 15% test sets. The training set is used to train the neural network systems, the validation set to prevent overfitting during training and to optimise the peak-picking parameters, and the test subset is used as unseen data for testing. The four SA systems, the four PC systems and the four CNN systems are evaluated along with two current state-of-the-art ADT systems: (1) tanhB, a BRNN system containing tanh cells [15] and (2) lstmpB, a BRNN system containing LSTMP cells. The LSTMP architecture was chosen as it outperformed GRU cells in preliminary testing on the same datasets. Drum onsets are selected from the output activation functions using the two peak-picking techniques.

#### *3.1.2 Drum Mixture and Multi-instrument Evaluations*

To determine system performance in a polyphonic context we use the *minusone* subset of the ENST Drums dataset

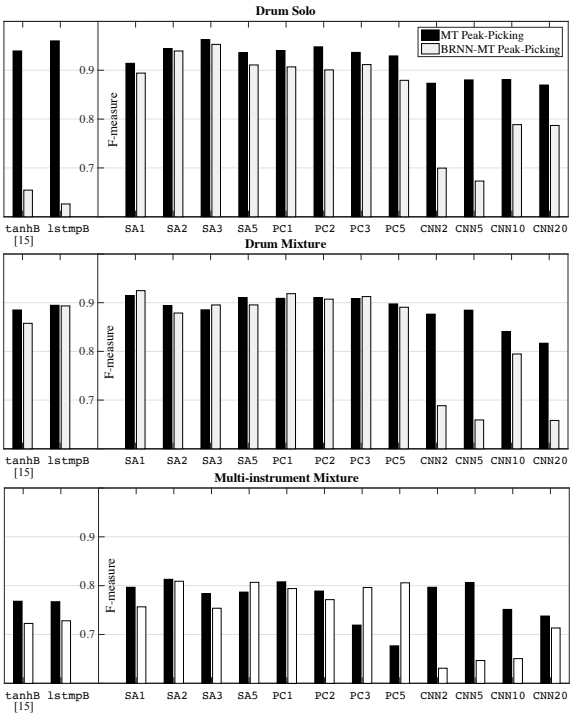


**Figure 3**: Mean instrument F-measures for *drum solo* (top), *drum mixture* (middle) and *multi-instrument mixture* (bottom) evaluations. Previous state-of-the-art RNN systems are on left and the SA, PC and CNN systems on right.

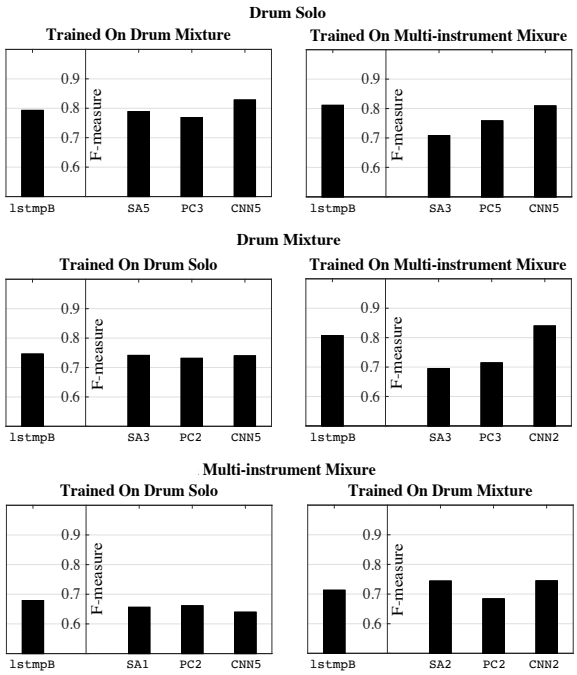


**Figure 4**: Mean instrument F-measure results with `MT` peak-picking for the *cross-context* evaluation: *drum solo* combinations (top); *drum mixture* combinations (middle); and *multi-instrument mixture* combinations (bottom).

[5]. The dataset contains 64 tracks divided into three different drummers (21 tracks by drummer 1, 22 tracks by drummer 2, and 21 tracks by drummer 3) with an average track length of 55 seconds. The dataset is composed of drum-only recordings which contain multiple drum instruments as well as accompaniment files. The drum only recordings are used for the *drum mixture* evaluation and the drum-only recordings are mixed with the accompaniment files using a ratio of 2/3 to 1/3 respectively for the *multi-instrument mixture* evaluation. The same training, validation and evaluation procedures are used as in the *drum solo* evaluation (Section 3.1.1).

#### *3.1.3 Cross-context Evaluation*

To test the adaptability of the trained systems to before unseen contexts we use the three systems trained in the previous evaluations (i.e., *drum solo*, *drum mixture*, and *multi-instrument mixture*) to process the datasets from the other two evaluations. This results in six *cross-context* evaluation combinations (e.g., train with *drum solo* test with *multi-instrument mixture*).

#### *3.1.4 Multi-context Evaluation*

To test how well a single system can be trained to adapt and perform in multiple contexts, we combine the training and validation data from the *drum solo*, *drum mixture* and *multi-instrument mixture* evaluations. The test data from the three evaluations is then processed using the single newly trained systems. Of the five evaluations this is the most realistic scenario.

## 4. RESULTS AND DISCUSSION

### 4.1 Drum Solo, Drum Mixture and Multi-instrument Mixture Results

Figure 3 highlights the mean instrument F-measure results of the SA, PC, CNN, and two previous state-of-the-art systems with both of the peak-picking strategies for the *drum solo*, *drum mixture* and *multi-instrument mixture* evaluations. The SA systems achieve the highest mean instrument F-measure in all three evaluations; 0.9880 (`SA3`), 0.9287 (`SA1`) and 0.9274 (`SA2`) respectively. The PC systems achieve higher F-measures in the *drum mixture* and *multi-instrument mixture* evaluations and the CNN systems achieve higher F-measures than the state-of-the-art systems in the *multi-instrument mixture* evaluation. This demonstrates that within the harder polyphonic contexts, allowing the output layer to access multiple hidden states and including the input features of multiple frames does enable higher performance to be achieved. The `BRNN-MT` peak-picking strategy improves the results of some of the SA and PC systems in both the *drum mixture* and *multi-instrument mixture* evaluations, demonstrating that the `BRNN-MT` strategy is able to improve performance in some contexts by learning to identify peaks within the noisier activation functions. For both the SA and PC systems the systems where $a \leq 3$ achieved the highest F-measures, we believe this is because of the extra information in the `SA5` and `PC5` systems is beyond the scope of the

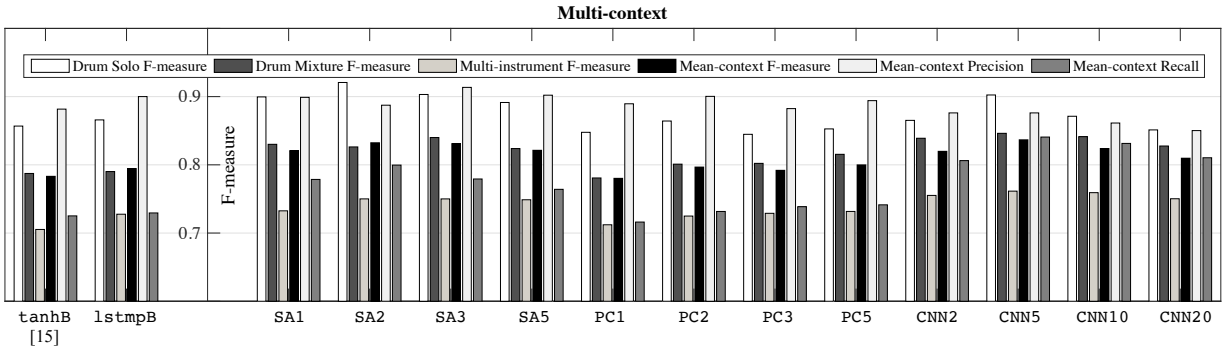


**Figure 5**: Results of the *multi-context* evaluation. For each system using the MT peak-picking technique the *drum solo*, *drum mixture*, *multi-instrument mixture*, and mean-context F-measures are shown in addition to the mean-context precision and mean-context recall.

onset and so has a negative effect on the performance. A similar trend is seen with the CNN systems which again can be explained by the larger input feature sizes reducing the impact of the relevant features. We believe that due to the *drum solo* evaluation being a relatively simple task, the less-complex RNN systems are able to achieve similar accuracies to the newly proposed systems and the CNN performs poorly on this same task due to noisy output activation functions which are the result of not passing information between time steps. This would also explain why the BRNN-MT strategy did not improve the results for the CNN systems and for any of the systems in the *drum solo* evaluation.

### 4.2 Cross-context Results

For each cross evaluation combination the top performing configuration of the existing state-of-the-art RNN, SA, PC and CNN systems using the MT peak-picking technique is displayed in Figure 4. The highest performing CNN system achieves a higher mean instrument F-measure than the highest performing current state-of-the-art RNN system (lsmtpB) in three out of the six combinations, the highest performing SA system only outperforms the current state-of-the-art RNN system in one of the combinations and the PC doesn't out perform the RNN system in any combinations. This suggests that the CNN system is more adaptable than the SA and PC systems even though the SA and PC systems achieve higher mean instrument F-measures than the CNN systems in the previous three evaluations. None of the highest accuracies were achieved by systems that used the BRNN-MT peak-picking strategy, which suggests that it is not suited for adapting to unseen situations.

### 4.3 Multi-context Results

Figure 5 highlights the *drum solo*, *drum mixture*, *multi-instrument mixture*, and mean-context F-measures using the MT peak-picking technique. Also included are the mean-context precision, and recall for each of the systems in the *multi-context* evaluation. The SA and CNN systems outperform the existing state-of-the-art and PC systems, further demonstrating the high performance of the SA systems and the adaptability of the CNN systems. This is achieved through higher recall, but not necessarily higher precision, suggesting that the improvement made by these systems is due to their ability to produce fewer false spikes within the resulting activation functions. All of the highest context F-measures were lower than the F-measures achieved by the systems trained in the single context focused evaluations (i.e., *drum solo*, *drum mixture*, and *multi-instrument mixture* evaluation) demonstrating that a system trained in multiple contexts can not outperform systems trained solely in one situation. The BRNN-MT peak-picking strategy again does not improve the performance of any of the systems in this evaluation.

## 5. CONCLUSIONS AND FUTURE WORK

We have presented three new neural network based systems for ADT in a polyphonic context: First, SA and PC systems that enable multiple time-step hidden states to be accessed by the output layer; and second, a CNN system that allows larger input feature sizes to be used. The results from the conducted evaluations demonstrate that all of the newly proposed systems achieve higher accuracies than the current state-of-the-art systems in polyphonic contexts, highlighting the effect of increased access to more information. Of all the tested systems, the SA performs best in either the single or multi-context, while the CNN systems perform best in situations in which the context is unseen. A possible future step would be to combine the SA and CNN systems into a single system possibly allowing the system to work in both situations (i.e., single and multiple contexts). An open source version of the newly proposed ADT systems can be found within the ADT libray (ADTLib). [1]

## 6. REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the international Conference on Learning Representations*, 2015.

[1] https://github.com/CarlSouthall/ADTLib

[2] Sebastian Böck, Jan Schlüter, and Gerhard Widmer. Enhanced peak picking for onset detection with recurrent neural networks. In *Proceedings of the 6th International Workshop on Machine Learning and Music (MML)*, pages 15–18, Prague, Czech Republic, 9 2013.

[3] Christian Dittmar and Daniel Gärtner. Real-time transcription and separation of drum recordings based on NMF decomposition. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 187–194, Erlangen, Germany, September 2014.

[4] Simon Durand, Juan Pablo Bello, Bertrand David, and Gaël Richard. Robust downbeat tracking using an ensemble of convolutional networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):76–89, 2017.

[5] 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.

[6] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NIPS*, 2015.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[8] Jeremy Irvin, Elliott Chartock, and Nadav Hollander. Recurrent neural networks with attention for genre classification. 2016.

[9] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Marius Miron, Matthew E. P. Davies, and Fabien Gouyon. An open-source drum transcription system for pure data and max MSP. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 221–225, Vancouver, BC, Canada, 2013.

[11] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[12] Jouni Paulus. *Signal Processing Methods for Drum Transcription and Music Structure Analysis*. PhD thesis, Tampere University of Technology, Tampere, Finland, 2009.

[13] Axel Röbel, Jordi Pons, Marco Liuni, and Mathieu Lagrange. On automatic drum transcription using non-negative matrix deconvolution and itakura saito divergence. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 414–418, Brisbane, Australia, 2015.

[14] Jan Schlüter and Sebastian Böck. Improved musical onset detection with convolutional neural networks. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6979–6983. IEEE, 2014.

[15] Carl Southall, Ryan Stables, and Jason Hockman. Automatic drum transcription using bi-directional recurrent neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 591–597, New York City, United States, August 2016.

[16] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[17] David Sussillo and Laurence F. Abbott. Training very deepnonlinear feed-forward networks with smart initialization. *arXiv preprint arXiv*, 1412, 2014.

[18] Lucas Thompson, Simon Dixon, and Matthias Mauch. Drum transcription via classification of bar-level rhythmic patterns. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 187–192, Taipei, Taiwan, 2014.

[19] Richard Vogl, Matthias Dorfer, and Peter Knees. Recurrent neural networks for drum transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 730–736, New York City, United States, August 2016.

[20] Richard Vogl, Matthias Dorfer, and Peter Knees. Drum transcription from polyphonic music with recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 201–205, New Orleans, Louisiana, United States, March 2017.

[21] Chih-Wei Wu and Alexander Lerch. Drum transcription using partially fixed non-negative matrix factorization with template adaptation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 257–263, Malaga, Spain, October 2015.

[22] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[23] 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.