Improved Acoustic Modeling for Automatic Piano Music Transcription Using Echo State Networks

Peter Steiner[1(✉)], Azarakhsh Jalalvand[2,3], and Peter Birkholz[1]

[1] Institute for Acoustics and Speech Communication, Technische Universität Dresden, Dresden, Germany
{peter.steiner,peter.birkholz}@tu-dresden.de
[2] IDLab, Ghent University–imec, Ghent, Belgium
azarakhsh.jalalvand@ugent.be
[3] Mechanical and Aerospace Engineering Department, Princeton University, Princeton, USA

Abstract. Automatic music transcription (AMT) is one of the challenging problems in Music Information Retrieval with the goal of generating a score-like representation of a polyphonic audio signal. Typically, the starting point of AMT is an acoustic model that computes note likelihoods from feature vectors. In this work, we evaluate the capabilities of Echo State Networks (ESNs) in acoustic modeling of piano music. Our experiments show that the ESN-based models outperform state-of-the-art Convolutional Neural Networks (CNNs) by an absolute improvement of 0.5 F_1-score without using an extra language model. We also discuss that a two-layer ESN, which mimics a hybrid acoustic and language model, achieves better results than the best reference approach that combines Invertible Neural Networks (INNs) with a biGRU language model by an absolute improvement of 0.91 F_1-score.

Keywords: Automatic piano transcription · Acoustic modeling · Echo state network

1 Introduction

Automatic Music Transcription (AMT) is one of the most challenging problems in Music Information Retrieval. The goal of AMT is to generate a score-like representation of a polyphonic audio signal. Due to many concurrently played notes from various instruments, complex overlapping of harmonics occurs in the acoustic signal. In many cases, the polyphony, e.g. the number of simultaneously active notes, is unknown and can vary over time. In recent years, AMT was successfully treated as a multi-label classification problem, in which every possible note is treated as one class. Recurrent Neural Networks (RNNs) define the state-of-the-art for acoustic modeling in piano transcription. In [2], one of the first approaches for acoustic modeling with recurrent neural networks was

I. Rojas et al. (Eds.): IWANN 2021, LNCS 12862, pp. 143–154, 2021.
https://doi.org/10.1007/978-3-030-85099-9_12

presented. The authors used multi-resolution features as input for an LSTM network that performed onset and pitch detection on the frame-level. Later, in [10], a large-scale study to determine features and different neural network architectures was conducted to find general guidelines towards simple acoustic modeling for piano transcription. The outcome was that feature design is important, and the best features are spectrum-like representations with log-spaced frequency bins and log-scaled magnitudes. Furthermore, it turned out that the Convolutional Neural Network (CNN) performed significantly better than a Deep Neural Network (DNN) and the All Convolutional Neural Network (AllConv).

Sigtia et al. [13] also used a CNN as acoustic model, but with an additional music language model that is supposed to learn the relationship between successive notes, similar as in speech recognition. The language model is the combination of a Recurrent Neural Network (RNN) and a Neural Autoregressive Distribution Estimator (NADE). This combination of an acoustic and language model led to smoother outputs compared to the purely acoustic model and thus improved the transcription results. Kelz et al. [11] recently showed that Invertible Neural Networks (INNs) together with RNNs as language models are performing slightly better than the CNN. So far, this is the best performing combination of an acoustic and language model that uses simple spectral features as input and output note probabilities for each frame.

In different studies [3,21], it was shown that incorporating onset information can boost piano transcription. Therefore, several systems combine onset information as language model with acoustic models. In [5], a complex multitask-approach was introduced. It consisted of models for onset detection, frame-wise pitch tracking, and for combining the information from onset detection and pitch tracking. All models were trained jointly. The ADSR model [9] incorporated attack, decay, sustain and release for each piano note. Currently, models that utilize onset and offset information in combination with larger models and a large-scale dataset [6] define the state-of-the-art in piano transcription.

In this paper, we investigate the potential of Echo State Networks (ESNs) [7] as *simple but effective* neural acoustic models for automatic transcription of piano music. In our prior work, it was shown that the performance of ESNs is similar to CNNs for automatic music transcription on the MusicNet dataset [17, 19] and for note onset detection [14,16]. However, as only a few types of models were evaluated on the MusicNet dataset and all of them were conceptionally complex and computationally intensive, it is difficult to objectively compare ESN with the contenders. On the other hand, the MAPS piano dataset [4], has been considered as a benchmark for many acoustic and hybrid models. This allows for a fair comparison with Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs), which are frequently used nowadays.

We compare our ESN-based acoustic model to a CNN-based acoustic model as a reference system and to several hybrid models that have an RNN in the second layer and show that a purely ESN-based approach outperforms a wide range of models.

The remainder of the paper is structured as follows: In Sect. 2, we introduce acoustic modeling using Echo State Networks. We explain our extracted features and the ESN model. Section 3 gives an overview about the utilized MAPS dataset and the metrics utilized to evaluate our model. After presenting our main results in Sect. 4, we end with conclusions and outlook in Sect. 5.

2 Acoustic Modeling Using Echo State Networks

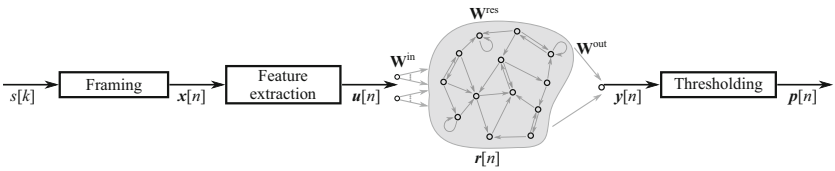


Fig. 1. Outline of the proposed ESN-based acoustic model from [17]: The audio signal $s[k]$ is divided into overlapping frames $\mathbf{x}[n]$ (hop size 10 ms). A filter-bank with logarithmic-spaced center frequencies and logarithmic magnitudes is used to extract spectral features $\mathbf{u}[n]$ for each frame n. The sequence of feature vectors is fed through the ESN-based acoustic model that computes the likelihoods $\mathbf{y}[n]$ for the presence of each note in the frame n. To obtain the binary piano-roll representation $\mathbf{p}[n]$, a global threshold is applied on the likelihoods.

Echo State Networks (ESNs) [7] are a variant of Recurrent Neural Networks (RNNs). In contrast to widely used RNN architectures, which usually consist of sequential layers that need to be trained jointly using iterative algorithms, the input and recurrent connections of ESNs are fixed by random values, and only the output weights are trained in one shot using linear regression. This one-shot training has two advantages compared to iterative algorithms:

- Fast training: All available data is presented *in one time* during training, which is time-efficient.
- Adaptability: The model can be *adapted to new data* in a later stage without presenting old data again.

In [17], we have summarized several properties that make ESNs interesting for multipitch tracking. Relying on our previous work, we adapted the system from [17] and evaluated its capability on acoustic modeling for piano transcription. The main outline of the proposed ESN-based acoustic model for piano transcription is depicted in Fig. 1.

2.1 Framing

The acoustic model works with feature vectors extracted from a discrete input signal $s[k]$, where k is the sample index. It is sampled with a sampling frequency $f_s = 44.1$ kHz. For the subsequent feature extraction, it is divided into overlapping frames $\mathbf{x}[n]$ of length 46.4 ms with a hop size of 10 ms, and with n being the discrete frame index.

2.2 Feature Extraction

For each frame, the short-term Fourier transform was computed. A triangular filter-bank with semitone-spaced center frequencies from 30 Hz to 17 000 Hz was applied on the short-term spectra to reduce the feature dimension. In order to avoid large negative values and to compress large magnitudes in the feature vector, the $\log_{10}$ was applied to the magnitude m plus 1, i.e. $\log_{10}(m+1)$.

To enrich the input feature vector with more temporal information, the first derivative of the computed magnitude spectrum was considered. Therefore, a first-order difference filter kernel with length of 3 frames was used to compute the temporal differences based on one frame before and after. The magnitude spectrum and its derivative were concatenated, so that each feature vector consisted of a spectrum and the first derivative.

We did not apply any additional standardization or normalization steps, and directly supplied these features as input $\mathbf{u}[n]$ to the ESN. This is a result from our previous work about note onset detection [14], where we found that any kind of standardization over-emphasized less important features with a very low variance. This is visualized in Fig. 2.

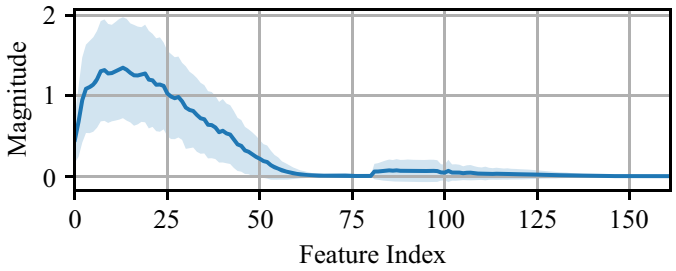


Fig. 2. Mean and variance for each feature. In higher frequencies, any kind of standardization would over-emphasize less important features with a very low variance.

2.3 Basic Echo State Network

The basic ESN outline that was used in this paper is depicted in in the center of Fig. 1 and is based on our general description of an ESN for multipitch tracking in [17], which was adapted in this paper. We briefly summarize the initialization of the ESN here. Basically, it consists of three weight matrices: The input weights $\mathbf{W}^{\text{in}}$ pass the input features to the reservoir, an unordered group of N^{res} nonlinear neurons. Since the reservoir weights $\mathbf{W}^{\text{res}}$ basically connect the reservoir neurons to each other in a recurrent fashion, past information can "echo" for some time inside the reservoir. The output weights $\mathbf{W}^{\text{out}}$ connect the neurons inside the reservoir to the output nodes.

Both, $\mathbf{W}^{\text{in}}$ and $\mathbf{W}^{\text{res}}$, were initialized from random distributions. The reservoir weights $\mathbf{W}^{\text{res}}$ fulfill the *Echo State Property* (ESP), which says that, for a

finite input sequence, the reservoir states need to decay in a finite time [7]. This was done by normalizing $\mathbf{W}^{\text{res}}$ to its maximum absolute eigenvalue.

In summary, the key difference between ESNs and typical RNN architectures is that $\mathbf{W}^{\text{in}}$ and $\mathbf{W}^{\text{res}}$ are initialized randomly and kept fixed during the training. Only the output weights $\mathbf{W}^{\text{out}}$ are trained using linear regression, whereas all weights need to be jointly optimized in typical RNN architectures.

To briefly recapitulate the main equations of an ESN, let $\mathbf{r}[n]$ represent the reservoir state. The two Eqs. (1) and (2) are then used to describe ESNs.

$$\mathbf{r}[n] = (1-\lambda)\mathbf{r}[n-1] + \lambda f_{\text{res}}(\mathbf{W}^{\text{in}}\mathbf{u}[n] + \mathbf{W}^{\text{res}}\mathbf{r}[n-1] + \mathbf{w}^{\text{bi}}) \tag{1}$$

$$\mathbf{y}[n] = \mathbf{W}^{\text{out}}\mathbf{r}[n] \tag{2}$$

Equation (1) is a leaky integration of the reservoir states $\mathbf{r}[n]$ with $\lambda \in (0, 1]$ being the leakage, and $f_{\text{res}}(\cdot)$ is the non-linear reservoir activation, in this paper the tanh-function. Every neuron in the reservoir receives a bias input using the bias weight vector $\mathbf{w}^{\text{bi}}$, which is initialized and fixed from a uniform distribution between ± 1. Equation (2) describes the N^{out}-dimensional output $\mathbf{y}[n]$ as a linear combination of a given reservoir state $\mathbf{r}[n]$.

During the training phase, all reservoir states $\mathbf{r}[n]$ were expanded by a constant of 1 as the intercept term for linear regression, and then collected in the reservoir state collection matrix $\mathbf{R}$. The desired binary outputs $\mathbf{d}[n]$, which are 0 for non-active and 1 for active pitches, are collected into the desired output collection matrix $\mathbf{D}$. Afterwards, $\mathbf{W}^{\text{out}}$ is obtained using ridge regression $\mathbf{W}^{\text{out}} = \left(\mathbf{R}\mathbf{R}^{\text{T}} + \epsilon\mathbf{I}\right)^{-1}(\mathbf{D}\mathbf{R}^{\text{T}})$, where the regularization parameter $\epsilon = 0.01$ penalizes large values in $\mathbf{W}^{\text{out}}$, and $\mathbf{I}$ is the identity matrix. The size of the output weight matrix $N^{\text{out}} \times (N^{\text{res}} + 1)$ determines the total number of free parameters to be trained in ESNs. The output $\mathbf{y}[n]$ indicated whether each note is active or not.

An ESN has several control parameters, which need to be tuned task-dependently: α_{u}, ρ, and α_{bi} control the absolute importance of the input feature vector, old reservoir state and the constant bias inputs, respectively. They are global scaling factors of the weight matrices $\mathbf{W}^{\text{in}}$, $\mathbf{W}^{\text{res}}$ and $\mathbf{w}^{\text{bi}}$. The leakage λ is a control parameter for the leaky integration and matches the input and output dynamics. The workflow to optimize the hyper-parameters is described detailed in [14,16,17].

2.4 Bidirectional Reservoirs

In the case of bidirectional reservoirs, the ESN is able to incorporate future information to compute the outputs. Therefore, the feature vectors are first fed through the ESN and the reservoir states are collected as described before. Next, the input is reversed in time and again fed through exactly the same model. This results in new reservoir states that are reversed in time. Afterwards, the reservoir states of both directions are concatenated to compute the output.

Bidirectional ESNs are usually more powerful than unidirectional models because (1) they have twice the number of free parameters in W^{out} and (2) they use future information to compute the output.

2.5 Stacked Reservoirs

In the case of stacked reservoirs, several ESN models are chained sequentially in layers as in [20]. Typically, subsequent ESNs receive the output of the previous layer as input. The target outputs of all layers are usually the same, and the layers are trained sequentially. By stacking reservoirs, the temporal modeling capacity of a single layer model is extended. This can be done for unidirectional as well as for bidirectional reservoirs.

In [8,17,20], it was shown that this improved the results for phoneme and image recognition, and for multipitch tracking, because subsequent layers are able to smooth the output of previous layers. In this paper, we also used a second layer to smooth the outputs of the first layer.

2.6 Thresholding

In [17], we have discussed that the output of an ESN after linear regression would ideally be zero in case of an absent and one in case of a present note. In practice, this is not always valid and the output is neither bounded between zero and one, nor truly binary. We used a simple thresholding method to convert the raw output values of the ESN into a binary piano roll representation. All output values above the threshold were set to one, the remaining to zero. The threshold was empirically set to 0.36 and tuned to maximize the F_1-Measure on the validation set.

3 Experimental Setup

The model was implemented in Python 3 using the `madmom` [2] framework for feature extraction and PyRCN[1] [15] for developing the ESN models. The code together with pre-trained models is available online in our GitHub repository.

3.1 MAPS Dataset

We use the MAPS dataset [4] to compare our proposed acoustic model with reference models. It contains audio files and annotations of isolated notes, chords, and piano pieces. All audio files are sampled with 44.1 kHz and are stored as stereo WAV files. We considered only the complete piano pieces. The MAPS dataset contains audio files rendered by software synthesizers and recordings from a Yamaha Disklavier player piano. In [13], the dataset was split into subsets for a 4-fold cross validation in different configurations. The most real-world case

[1] https://github.com/TUD-STKS/PyRCN.

that was used for later studies is called "Configuration 2" and uses just the synthetic audio files for training/optimization. The Disklavier recordings were held back as an unseen test set. In [5], the authors modified "Configuration 2" and removed music pieces from the training set that are present in both, training and test sets. For the sake of fair comparison and as they have re-trained several reference models based on this reduced training set, we opted for the same MAPS configuration as [5,11].

3.2 Evaluation Metrics

The evaluation was based on standard frame-level metrics proposed in [1], namely Precision P, Recall R and F_1. We used the library *mir_eval* [12] to compute P and R. The F_1-Measure in Eq. (3) is the harmonic mean of both, P and R. For details on the metrics, we refer to [1].

$$F_1 = 2\frac{P \cdot R}{P + R} \tag{3}$$

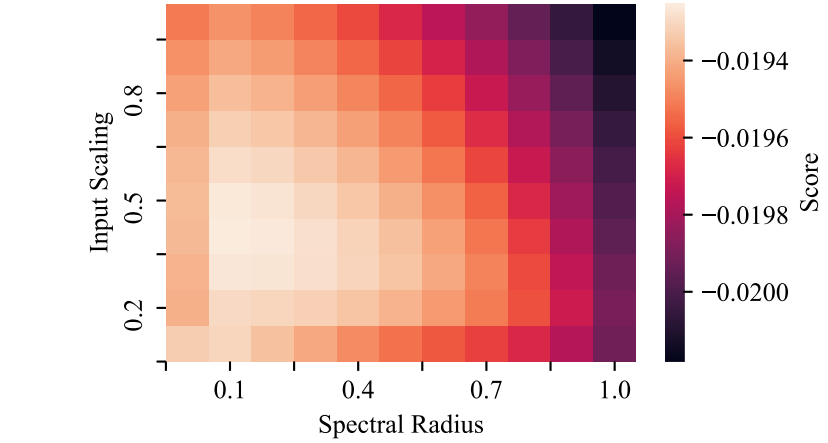
4 Results on the MAPS Dataset

4.1 Hyperparameter Optimization

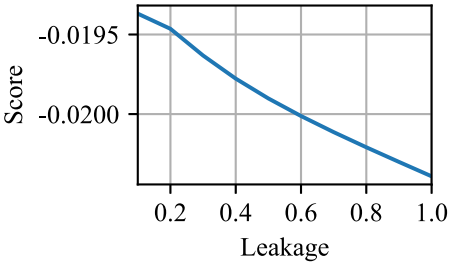
We sequentially optimized the hyper-parameters minimizing the mean squared error as in [17] in the steps (a) ρ and α_{u}, (b) λ, and (c) α_{bi}, respectively using 5-fold cross validation. Figure 3 shows the cross validation scores (negative mean squared error) for each step. We can see that the ESN model benefits from recurrent connections, because $\rho = 0.1$ and $\lambda = 0.1$. The bias has only a small impact on the final performance. We also checked the regularization parameter ϵ that, however, did not influence the results.

4.2 General Observations

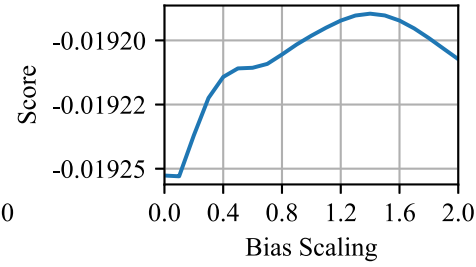
Figure 4 compares the performance of uni- and bidirectional ESNs next to a reference CNN model. It can be observed that the F_1-Measure strongly depends on the reservoir size N_{res}. Incorporating future information by using bidirectional architectures strongly improves the results over unidirectional architectures. In fact, the proposed bidirectional model with 12000 neurons outperforms the reference CNN [10] (marked by the dashed line), the best performing acoustic model with comparable features. Of course, we need to note that the CNN utilizes a limited amount of future information (two frames) compared to the ESN.



(a) Optimization of input scaling and spectral radius



(b) Optimization of leakage



(c) Optimization of bias scaling

Fig. 3. Hyperparameter optimization

Training and inference of this ESN model with ca. 12M free parameters on a modern laptop CPU is still feasible, which is an important aspect for real-world applications. Further enlarging the reservoir still slightly improves the performance of the model, but at the cost of many more free parameters.

Figure 5 shows the transcription result for the first 30 s of the piano piece "MAPS_MUS-bk_xmas5_ENSTDkCl" computed with the bidirectional ESN model with 24 000 neurons (ESN 24 000bi). It can be seen that many false positive notes (green) were recognized in higher pitch areas. In many cases, a pitch was recognized along with a higher harmonic of itself, leading to an octave error. The missing notes often occurred for low pitches. However, most of the notes were recognized properly.

4.3 Comparison to the State of the Art

In Table 1, we summarize the performance of the proposed ESN-based acoustic models along with some reference models. It is worth to mention that all models but the CNN + LSTM [5] and ADSRNet [9] are supplied only with the extracted

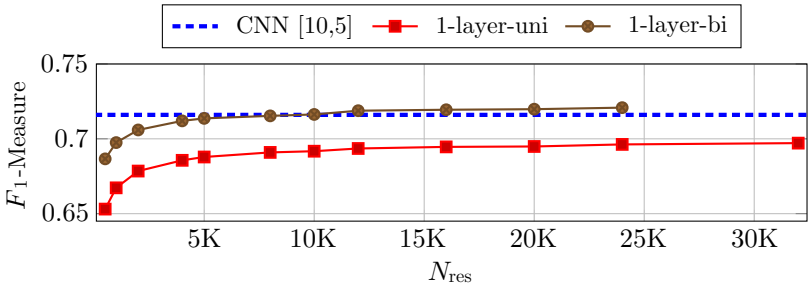


Fig. 4. Pitch detection results on the MAPS test dataset in the modified configuration 2 [5]: A strong dependency of the F_1-Measure from the reservoir size N_{res} and a uni- or bidirectional architectures can be observed.

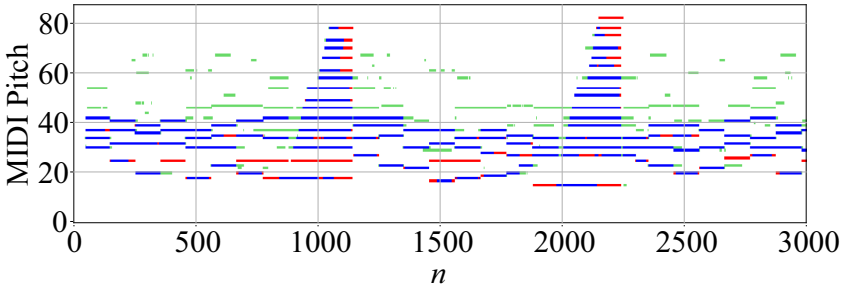


Fig. 5. Transcription result for the first 30 s of the piano piece "MAPS_MUS-bk_xmas5_ENSTDkCl" computed with the model "ESN 24 000b". The blue color (true positive) indicates that most of the notes are transcribed correctly. The green color (false positive) shows that the acoustic model recognized many additional notes, especially in the higher frequency ranges. The red color (false negative) indicates that, especially for lower pitches, several notes were not recognized. (Color figure online)

acoustic features and their derivatives. The latter onces also take advantage of much more information about, e.g., on-/offset or sustain pedal information in addition to the conventional acoustic features.

This comparison suggests that the bidirectional ESN as acoustic model outperforms the CNN-based model [10]. Furthermore, we suppose that the ESN is better than the INN for acoustic modeling, as it even suppresses the INNs with small language models (e.g. GRU (S) and LSTM (S)) as well. Only in case of a large bi-directional language model (biGRU), the models from [11,13] perform slightly better. In order to compare simple combinations of acoustic and language models more fairly, we have also trained a second layer that should denoise and smoothen the frame-based output of the acoustic model. We can see that this strongly improves the results. The middle rows of Table 1 show that our two-layer system perform better than the typical combinations of acoustic and language model.

Table 1. Results on the MAPS test dataset under Configuration 2 without duplicated music pieces in the training set [5]. The ESN has outperformed the CNN-based acoustic model. Both, CNN and ESN performed better than the combination of the INN with small RNN language models. With an additional small ESN-based language model, we were able to outperform all classic combinations of acoustic and language models. The models [5,9] with a lot of additional information still perform better.

Method	P	R	F_1	Only spectral features
CNN only [5,10]	**81.18**	65.07	71.60	x
ESN 32 000u	71.63	67.89	69.71	x
ESN 24 000b	72.89	**71.33**	**72.10**	x
INN + GRU (S) [11]	79.74	63.73	70.84	x
INN + LSTM (S) [11]	80.12	63.91	71.10	x
INN + biGRU (L) [11]	**81.72**	64.81	72.29	
CNN + RNN-NADE [5,13]	71.99	**73.32**	72.22	x
ESN 24 000b, 5000b	81.06	66.73	**73.20**	x
CNN + LSTM [5]	88.53	**70.89**	**78.30**	–
ADSRNet [9]	**90.73**	67.85	77.16	–

The models [5,9] are quite different in terms of utilizing additional information about on- and offset etc., and can thus be not entirely compared to the proposed ESN model. Table 1 shows that the additional information is very useful for piano transcription and significantly improves the recognition results. In the future, we will extend the current approach towards incorporating additional information in a similar way. We have not listed the results of [22] because in those experiments there is some overlap between the training and testing data.

5 Conclusions and Outlook

Our proposed ESN-based acoustic model for piano transcription, which relies on simple spectral features, outperformed a wide variety of deep learning models, such as CNN, INN, and hybrid models which benefit from combination of acoustic and language models. All approaches have in common that they purely rely on spectral feature vectors, from which note likelihoods are computed. Encouraged by promising results for onset detection using ESNs [14], one way to move forward would be to incorporate additional information about on-/offsets [5,6,9] in the current system and also to use the MAESTRO dataset. In [18], data augmentation was also shown to improve multipitch tracking.

Furthermore, we will investigate the length of effective future information in bidirectional ESNs. If only a limited window of future frames is required, ESNs could be incorporated into real-time systems for piano transcription.

Acknowledgement. The parameter optimizations were performed on a Bull Cluster at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden. This research was also partially funded by Ghent University (BOF19/PDO/134).

References

1. Bay, M., Ehmann, A.F., Downie, J.S.: Evaluation of multiple-F0 estimation and tracking systems. In: Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, 26–30 October 2009, Kobe, Japan, pp. 315–320 (2009). http://ismir2009.ismir.net/proceedings/PS2-21.pdf
2. Böck, S., Schedl, M.: Polyphonic piano note transcription with recurrent neural networks. In: ICASSP 2012–2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 121–124, March 2012
3. Cheng, T., Mauch, M., Benetos, E., Dixon, S.: An attack/decay model for piano transcription. In: Mandel, M.I., Devaney, J., Turnbull, D., Tzanetakis, G. (eds.) Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, 7–11 August 2016, New York City, USA, pp. 584–590 (2016). https://archives.ismir.net/ismir2016/paper/000085.pdf
4. Emiya, V., Badeau, R., David, B.: Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. IEEE Trans. Audio Speech Lang. Process. **18**(6), 1643–1654 (2010)
5. Hawthorne, C., et al.: Onsets and frames: dual-objective piano transcription. In: Gómez, E., Hu, X., Humphrey, E., Benetos, E. (eds.) Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, 23–27 September 2018, Paris, France, pp. 50–57 (2018). http://ismir2018.ircam.fr/doc/pdfs/19_Paper.pdf
6. Hawthorne, C., et al.: Enabling factorized piano music modeling and generation with the MAESTRO dataset. In: International Conference on Learning Representations (2019). https://openreview.net/forum?id=r1lYRjC9F7
7. Jaeger, H.: The echo state approach to analysing and training recurrent neural networks. Technical report GMD Report 148, German National Research Center for Information Technology (2001). http://www.faculty.iu-bremen.de/hjaeger/pubs/EchoStatesTechRep.pdf
8. Jalalvand, A., Demuynck, K., Neve, W.D., Martens, J.P.: On the application of reservoir computing networks for noisy image recognition. Neurocomputing **277**, 237–248 (2018). hierarchical Extreme Learning Machines
9. Kelz, R., Böck, S., Widmer, G.: Deep polyphonic ADSR Piano note transcription. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 246–250, May 2019
10. Kelz, R., Dorfer, M., Korzeniowski, F., Böck, S., Arzt, A., Widmer, G.: On the potential of simple framewise approaches to piano transcription. In: Mandel, M.I., Devaney, J., Turnbull, D., Tzanetakis, G. (eds.) Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, 7–11 August 2016, New York City, United States, pp. 475–481 (2016). https://wp.nyu.edu/ismir2016/wp-content/uploads/sites/2294/2016/07/179_Paper.pdf
11. Kelz, R., Widmer, G.: Towards interpretable polyphonic transcription with invertible neural networks. In: Flexer, A., Peeters, G., Urbano, J., Volk, A. (eds.) Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, 4–8 November 2019, Delft, The Netherlands, pp. 376–383 (2019). http://archives.ismir.net/ismir2019/paper/000044.pdf

12. Raffel, C., et al.: `mir_eval`:a transparent implementation of common MIR metrics. In: Wang, H., Yang, Y., Lee, J.H. (eds.) Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, 27–31 October 2014, pp. 367–372 (2014). https://archives.ismir.net/ismir2014/paper/000320.pdf
13. Sigtia, S., Benetos, E., Dixon, S.: An end-to-end neural network for polyphonic piano music transcription. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(5), 927–939 (2016)
14. Steiner, P., Jalalvand, A., Stone, S., Birkholz, P.: feature engineering and stacked echo state networks for musical onset detection. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 9537–9544, January 2021
15. Steiner, P., Jalalvand, A., Stone, S., Birkholz, P.: PyRCN: Exploration and Application of ESNs (2021)
16. Steiner, P., Stone, S., Birkholz, P.: Note Onset Detection using Echo State Networks. In: Böck, R., Siegert, I., Wendemuth, A. (eds.) Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020, pp. 157–164. TUDpress, Dresden (2020)
17. Steiner, P., Stone, S., Birkholz, P., Jalalvand, A.: Multipitch tracking in music signals using Echo state networks. In: 2020 28th European Signal Processing Conference (EUSIPCO), pp. 126–130 (2020). https://www.eurasip.org/Proceedings/Eusipco/Eusipco2020/pdfs/0000126.pdf
18. Thickstun, J., Harchaoui, Z., Foster, D.P., Kakade, S.M.: Invariances and data augmentation for supervised music transcription. In: ICASSP 2018–2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2241–2245, April 2018
19. Thickstun, J., Harchaoui, Z., Kakade, S.M.: Learning features of music from scratch. In: 5th International Conference on Learning Representations, ICLR 2017, 24–26 April 2017, Toulon, France, Conference Track Proceedings (2017). https://openreview.net/forum?id=rkFBJv9gg
20. Triefenbach, F., Jalalvand, A., Schrauwen, B., Martens, J.P.: Phoneme recognition with large hierarchical reservoirs. In: Advances in Neural Information Processing Systems 23, pp. 2307–2315. Curran Associates, Inc. (2010). http://papers.nips.cc/paper/4056-phoneme-recognition-with-large-hierarchical-reservoirs.pdf
21. Wang, Q., Zhou, R., Yan, Y.: A two-stage approach to note-level transcription of a specific piano. Appl. Sci. **7**(9), 901 (2017)
22. Wu, Y., Chen, B., Su, L.: Polyphonic music transcription with semantic segmentation. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 166–170, May 2019