Audio Engineering Society

Conference Paper

Presented at the Conference on
Semantic Audio
2017 June 22 – 24, Erlangen, Germany

Deep Learning for Jazz Walking Bass Transcription

Jakob Abeßer[1,3], Stefan Balke[2], Klaus Frieler[3], Martin Pfleiderer[3], and Meinard Müller[2]

[1]*Semantic Music Technologies Group, Fraunhofer IDMT, Germany*
[2]*International Audio Laboratories, Erlangen, Germany*
[3]*Jazzomat Research Project, University of Music Franz Liszt, Weimar, Germany*

Correspondence should be addressed to Jakob Abeßer (jakob.abesser@idmt.fraunhofer.de)

ABSTRACT

In this paper, we focus on transcribing walking bass lines, which provide clues for revealing the actual played chords in jazz recordings. Our transcription method is based on a deep neural network (DNN) that learns a mapping from a mixture spectrogram to a salience representation that emphasizes the bass line. Furthermore, using beat positions, we apply a late-fusion approach to obtain beat-wise pitch estimates of the bass line. First, our results show that this DNN-based transcription approach outperforms state-of-the-art transcription methods for the given task. Second, we found that an augmentation of the training set using pitch shifting improves the model performance. Finally, we present a semi-supervised learning approach where additional training data is generated from predictions on unlabeled datasets.

1 Introduction

In many jazz styles, bass players commonly play walking bass lines, which consist mostly of quarter notes and few rhythmic variations. Since these notes coincide with beat positions, the bass supports the drums in creating a rhythmic pulse. At the same time, walking bass lines contribute to the harmonic fundament by emphasizing important chord tones (root, third, fifth) on metrically accented beats. Thus, they provide a rhythmic and harmonic backbone for many jazz tunes and are of genuine interest to jazz researchers.

Since jazz improvisers often relate to the chords of a composition, knowing the correct harmonic structure is of crucial interest for analyzing solo improvisations. However, while taking the composition as a foundation for their playing, jazz musicians often modify the original chord progressions. In order to determine the actually played chords from a jazz recording, automatic music transcription methods could be applied. But a reliable polyphonic transcription of the piano or guitar from ensemble recordings can still be considered an unsolved problem [1]. To mitigate this problem, walking bass lines could be used to provide initial cues for a full harmonic analysis and also to validate external chord annotations such as currently used in the Weimar Jazz Database [2].

In this paper, we propose to transcribe the underlying (monophonic) walking bass line to obtain initial cues for a subsequent harmonic analysis in jazz recordings. Instead of performing a full transcription, we aim to extract a *bass salience representation*. The bass salience measures a kind of likelihood that the bass instrument plays certain pitches over time. We represent

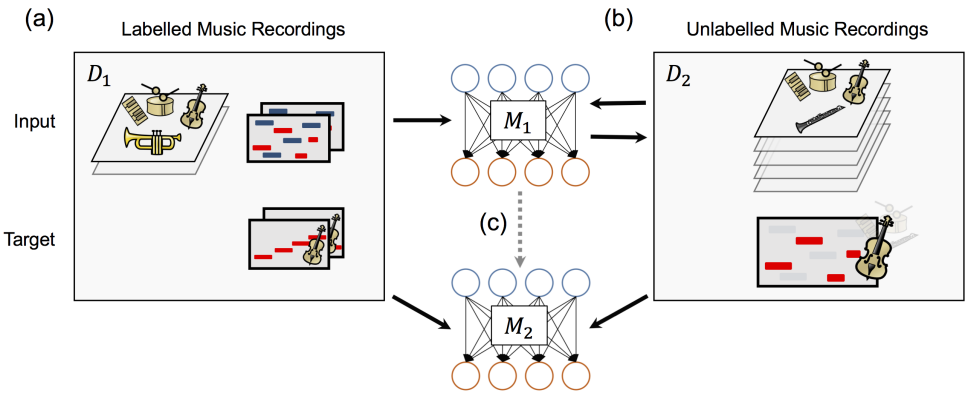

Fig. 1: Visualization of the proposed system. (a) The labelled dataset D_1 is used for training the DNN to derive model M_1. (b) M_1 is used to create labels for the unlabelled music recordings in dataset D_2. (c) D_1 and D_2 are used as a combined training set to derive the DNN model M_2.

the walking bass line as a sequence of *beat-wise pitch values*, which we obtain by frame-wise aggregation of extracted bass saliences between annotated beat times.

Our proposed system is based on training a deep neural (DNN) on jazz music recordings equipped with walking bass line transcriptions, as shown in Figure 1(a). Currently, we only have a small number of 41 jazz music recordings (D_1) with ground truth transcriptions. In addition, we aim to exploit another 237 jazz music recordings (D_2) of similar music style without bass transcriptions in order to implement a semi-supervised learning procedure.

In this paper, our first contribution is the adaptation of a previously proposed deep neural network architecture [3, 4] to the task of walking bass line transcription. This technique allows for an automated learning of a mapping from a given spectrogram representation of the music recording to a pitch salience representation of the bass instrument. Furthermore, we investigate two data augmentation techniques to further improve our trained DNN. First, we increase the labelled data by simply shifting the input features to make the DNN more robust to key changes. Second, we apply a bootstrapping approach inspired by [5] to examine the benefit of using unlabelled data to further generalize our trained DNN. At the same time, we evaluate the performance of state-of-the-art bass transcription methods for the given task. Note that this paper has an accompanying website at [6] where one can find further audio examples and details about the annotations used in this paper.

2 Related Work

Several bass transcription algorithms were proposed in the MIR literature so far which approach the bass part as the lowest dominant melody line with fundamental frequency (f_0) values between around 40 Hz and 400 Hz. Considering the low frequency range, low-pass filtering in combination with downsampling is often the first processing stage [7, 8, 9], which accelerates the transcription by filtering out harmonic components from instruments in higher frequency ranges. Instead of fixing an upper f_0 limit, Ryynänen and Klapuri [10] adapt the frequency band of interest by dynamically estimating the f_0 range of the bass line. Some authors apply source separation techniques to filter out interfering instruments prior to the bass transcription. For instance, Tsunoo et al. [11] use the harmonic-percussive sound separation (HPSS) algorithm to attenuate spectral components of percussive instruments.

In terms of suitable spectral representations for bass transcription, the short-time Fourier transform (STFT) is often applied [7, 12, 10]. However, due to the limited temporal-spectral resolution in lower frequency ranges, different methods such as the instantaneous frequency (IF) spectrogram [7, 8] or the constant-Q spectrogram [13] are also common. For details on different spectral representations used in MIR, see e. g., [14]. A subsequent *note detection* stage is performed either in the time domain [7, 12] using envelope extraction methods or in the frequency domain by grouping frame-wise f_0 estimates to note events [10, 8].

Dataset	Usage	Ann.	# Files	# Notes	Duration [h]
D_1	Training	✓	31	3899	0.43
D_1^+	Training	✓	93	11697	1.30
D_2	Training	-	237	-	7.16
D_2^+	Training	-	711	-	21.49
D_3	Test	✓	10	1101	0.12

Table 1: Summary of the datasets. In the forelast two columns, the number of audio files and number of notes are given before and after data augmentation, respectively.

Hyperparameter	Values
# Hidden layers	3, **4**, 5
# Context frames	1, 3, **5**
Dropout (%)	0, **25**, 50
L_2 weight regularization	**disabled**, 10^{-3}

Table 2: Variations over model hyperparameters. Optimal parameter values are given in bold print.

Our approach is inspired by previously proposed algorithms based on a *harmonic salience function*, which provides a likelihood measure for different pitch candidates. Klapuri and Ryynänen compute an f_0 candidate's harmonic salience by summing up the spectral energies at corresponding harmonic frequencies [10]. In contrast, Salamon et al. use a logarithmic frequency representation in combination with instantaneous frequency estimation methods and harmonic summation [9, 15].

Recent advances in DNNs stimulated new progress in automatic music processing algorithms. Böck and Schedl [16] used a recurrent neural network (RNN) based on Long Short-Term Memory (LSTM) units to obtain a pitch salience representation from polyphonic piano recordings. Recently, Kelz et al. [17] systematically evaluated different network hyperparameters on the performance of DNNs for frame-wise polyphonic piano transcription. The authors compared three different network configurations based on feed-forward and convolutive DNNs, which outperformed an alternative piano transcription algorithm proposed by Sigtia et al. in [18], which uses an RNN to model the combination of an acoustic and a musical language model.

3 Proposed Method

3.1 Datasets

Our dataset is a subset of 41 jazz solo recordings taken from the Weimar Jazz Database (WJD)[1], which contains 456 high-quality solo melody transcriptions and covers numerous jazz performers and styles. For this work, musicology students transcribed excerpts of walking bass lines of 41 recordings using the Sonic Visualiser software [19]. As shown in Table 1, we put aside 10 files as final test set D_3. The remaining 31 files are used as labeled training set D_1. Furthermore, we add another 237 recordings from the WJD database as unlabeled training set D_2 without any bass annotation available for semi-supervised learning as will be explained in Section 3.4 and Section 3.6. These recordings have the same music style (swing feeling with walking bass lines) as the annotated files.

3.2 Data Augmentation

In order to enlarge the datasets D_1 and D_2, we perform *data augmentation* and created two additional versions of each audio recording by applying pitch shifting using the *sox* audio library[2] one semitone upwards and downwards, respectively. As a side effect, this procedure balances the overall pitch distribution in the training set. The enlarged datasets are denoted as D_1^+ and D_2^+ and summarized in Table 1.

3.3 Input Features & Targets

We resample each audio signal to a sampling rate of 22.05 kHz and compute the constant-Q magnitude spectrogram using the *librosa* python library [20] with a hopsize of 1024 (46.4 ms) and a frequency resolution of 12 bins per octave as input features. We consider the pitch range of a double bass ranging from MIDI pitch 28 and 67 (f_0 values from 41.2 Hz to 392 Hz). Pitch annotations are converted to binary pitch saliency vectors, which serve as target representation for multilabel classification. Both the input features and target values have the same dimensionality of $N = 40$.

3.4 Model Training

In the following, we introduce a bass transcription algorithm, which builds upon previous work from Uhlich et al. [3], who used a DNN to learn a mapping

[1] http://jazzomat.hfm-weimar.de/dbformat/dboverview.html

[2] http://sox.sourceforge.net/

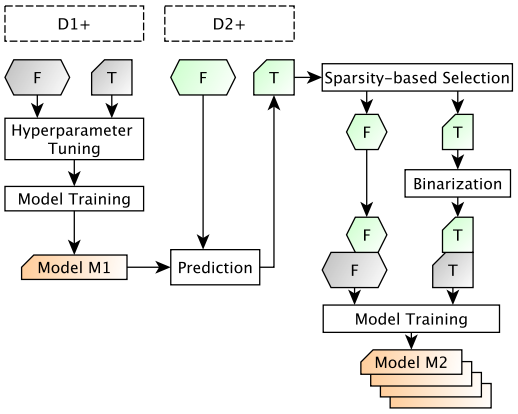

Fig. 2: Proposed semi-supervised training procedure for the DNN model. Feature and target matrices are denoted as F and T, respectively.

from a mixture magnitude spectrogram to the magnitude spectrogram of a single instrument in a source separation context. The DNN architecture has several fully-connected hidden layers with the same size as the output layer. Rectified linear units (ReLU) are used as activation function for the hidden layers and the sigmoid function is used for the output layer. Uhlich et al. [3] propose to train the DNN layers in succession. For each layer training, least-squares estimates based on the output of the previous layer are used to initialize the corresponding weight values [21]. This approach was adapted by Balke et al. [4], who changed the mapping objective to obtain a pitch salience representation of the melody instrument from multi-instrumental jazz recordings. In this work, we adapt and apply this model to learn a salience representation for the bass line from a mixture spectrogram.

Our proposed semi-supervised learning procedure is illustrated in Figure 1 and Figure 2. First, we use the labeled dataset D_1^+ to train an initial encoder model M_1^+ based on the architecture proposed in [4]. For the optimization, we use the ADADELTA [22] algorithm, a mini-batch size of 500 (samples per gradient update), 500 epochs (gradient updates) for the training of each layer, and the mean squared error loss function as provided by the *keras* python library[3]. During initial tests on dataset D_1, we found the best training convergence for a learning rate of 1. Since we focus on frame-wise pitch saliency modeling, we normalize all feature vectors by their Euclidean norm before they are input to the DNN model.

[3] https://github.com/fchollet/keras

3.5 Hyperparameter Tuning

For the hyperparameter tuning, we compare 54 different model parameterizations as shown in Table 2 inspired by [23, 17]. Each parametrization is evaluated using a 3-fold cross validation and the final validation loss is averaged over all folds and minimized to obtain the optimal training configuration. In addition to different number of layers, we investigate, whether frame stacking improves the model performance. We compare different context frame sizes ($N_C \in [1,3,5]$). For example, for $N_C = 5$, we concatenate each spectral frame with the two preceding and two succeeding spectral frames to a feature vector of dimension $N_C \cdot N = 200$ (where $N = 40$, see Section 3.3). Also, we investigate, whether dropout between the hidden layers and an L_2 weight regularization improves the model's robustness.

Our experiments showed that a network with 4 layers, 5 context frames, 25 % dropout, and no weight regularization showed the best performance on the dataset D_1^+. The optimal number of layers is close to the 5 layers used by Balke et al. in [4] for melody pitch salience estimation. The incorporation of temporal context (frame stacking) seems beneficial for our application scenario. One possible reason could be that most bass notes in the walking bass style are relatively long (quarter notes) and have a stable pitch contour.

3.6 Sparseness-Based Selection

Dittmar et al. [5] proposed to include predicted feature vectors from unlabeled test data using a bootstrapping approach. Using the augmented training set, the authors performed a re-training of the model to adapt it to the targeted test data. We adopt these approaches to implement a semi-supervised learning procedure as follows.

We denote a feature vector (model input) as $F \in \mathbb{R}^{N \cdot N_C}$ and the corresponding target vector (for model training) or prediction of the model as $T \in \mathbb{R}^N$, respectively. Using the model M_1^+, we first obtain predictions on the unlabeled dataset D_2^+. In order to measure whether a given pitch salience vector shows only one or multiple salient pitches, we use the following *sparseness* measure s proposed in [24], which

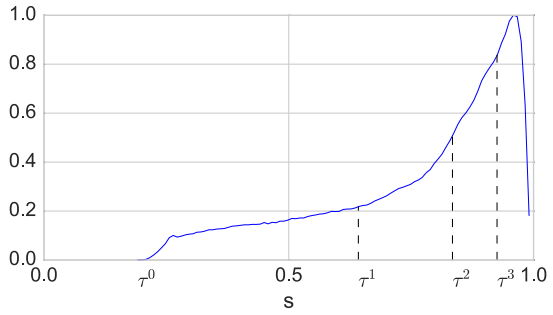

Fig. 3: Normalized histogram over sparseness values s of predictions of model M_1^+ for all files in dataset D_2^+. Sparseness thresholds τ^q are shown as vertical lines.

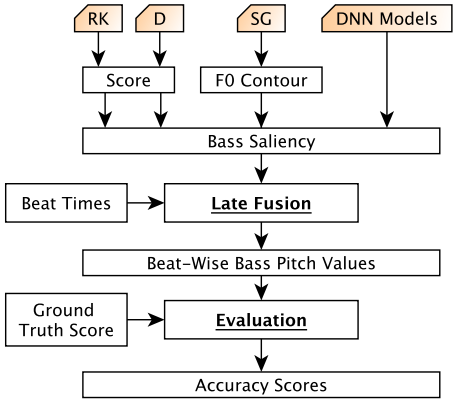

Fig. 4: Sketch of the experimental procedure.

is based on the relationship between the L_1 norm and L_2 norm of a given (predicted) pitch salience vector $T = (T_1, T_2, \ldots, T_N)^{\mathrm{T}}$:

$$s(T) = \frac{\sqrt{N} - (\sum_{i=1}^{N} T_i)/(\sqrt{\sum_{i=1}^{N} T_i^2})}{\sqrt{N} - 1} \text{ with } s \in [0,1]. \tag{1}$$

If T has only a small number of non-zero components, s will be close to 1. In that case, we assume that T corresponds to a voiced time frame with one bass pitch being predominant, which can be used as additional training data. In contrast, if s is close to 0, we assume that no bass pitch is predominant in this time frame and T represents either an unvoiced frame or a unconfident model prediction for a voiced frame. We select all predicted pitch salience vectors T with sparseness values $s(T) > \tau$ greater than a given threshold $\tau > 0$ as additional training data to train M_2^+.

In the experiments discussed in Section 4, we evaluate six different variants of the DNN model. We compare the model M_1^+ trained on the dataset D_1^+ and the model M_1 trained on the dataset D_1 that excludes the data augmented files. Also, we include four different variants of model M_2^+, which we obtain by varying the sparseness threshold $\tau = \tau^q$. We compute τ^q from the q-th quartile Q_q of the distribution of the sparseness values s over all predictions of the model M_1^+ over the dataset D_2^+. We investigate all three quartiles Q_1, Q_2, and Q_3 (denoted as $q \in [1,2,3]$) as well as the special case of using all frames of D_2^+ as additional data (denoted as $q = 0$).

Figure 3 shows a histogram over the sparseness values s over all predictions from model M_1^+ on the unlabeled dataset D_2^+ as well as the corresponding thresholds τ^q. The thresholds obtained from the distribution percentile values are illustrated as vertical lines. By increasing τ, we reduce the size of additional training data at the one hand but focus on more confident annotations on the other hand.

3.7 Binarization

Before including the additional training data to the initial training set D_1^+, we binarize the corresponding predictions T in such a way that only the largest entry remains 1 and all others become 0. This leads to annotations similar to those in D_1^+ with only one pitch being marked. After adding these frames to D_1^+, we use the trained model M_1^+ as initialization and perform 500 additional training epochs on all layers using the combined training set. Depending on the applied threshold τ_q (compare Section 3.6), the resulting models are labeled as $M_2^{q,+}$.

3.8 Beat-Informed Late Fusion

We aim to represent the walking bass line as sequence of beat-wise pitch values. Usually, automatic beat-detection methods need to be applied to estimate the beat times in a given audio recording. In this paper, we circumvent this step by using manually annotated beat times provided by the WJD database. After we estimate a pitch salience representation for a given audio recording, we average the pitch salience values between successive beat times and obtain a pitch estimate by taking the most salient pitch as bass pitch for the current beat.

Alg.	Frame-wise		Beat-wise		Sparseness
	A	A_{PC}	A	A_{PC}	s
SG	0.28 (0.14)	0.39 (0.15)	0.68 (0.22)	0.75 (0.21)	-
RK	0.12 (0.13)	0.18 (0.14)	0.60 (0.27)	0.64 (0.26)	-
D	0.37 (0.20)	0.41 (0.19)	0.72 (0.16)	0.75 (0.15)	-
M_1	0.31 (0.09)	0.43 (0.10)	0.71 (0.17)	0.78 (0.14)	0.684 (0.035)
M_1^+	**0.57** (0.13)	0.70 (0.11)	**0.83** (0.13)	0.89 (0.11)	0.761 (0.018)
$M_2^{0,+}$	0.54 (0.12)	0.68 (0.11)	0.81 (0.14)	0.88 (0.12)	**0.954** (0.010)
$M_2^{1,+}$	0.54 (0.13)	0.70 (0.11)	0.81 (0.14)	0.89 (0.11)	0.935 (0.015)
$M_2^{2,+}$	0.55 (0.12)	0.71 (0.11)	0.82 (0.14)	0.89 (0.12)	0.922 (0.019)
$M_2^{3,+}$	0.56 (0.12)	0.70 (0.11)	0.82 (0.14)	0.88 (0.12)	0.862 (0.030)

Table 3: Mean pitch detection accuracy values A and A_{PC} and mean frame-wise sparseness values s averaged over all test files (standard deviation values given in brackets). Both accuracy measures are computed frame-wise and beat-wise. Highest accuracy values A and sparseness values are denoted in bold print.

4 Evaluation

4.1 Experimental Procedure

Figure 4 gives an overview over the experimental design. The dataset D_3 is used as test set. We obtain bass salience predictions from the six models M_1, M_1^+, $M_2^{0,+}$, $M_2^{1,+}$, $M_2^{2,+}$, and $M_2^{3,+}$ described in Section 3.6. The state-of-the-art bass transcription algorithms by Ryynänen & Klapuri (RK) [10] and Dittmar et al. (D) [7] output a list of note events (score) whereas the algorithm from Salamon & Gomez (SG) [9] outputs a frame-wise f_0 contour of the bass line.[4]

We first convert all transcription results into corresponding bass salience representations using a fixed temporal resolution of 46.4 ms and a pitch resolution of one semitone in order to have a comparable signal representation across all algorithms. In the next step, we use annotated beat times taken from the WJD database to aggregate the bass salience representations over frames between successive pairs of beats as described in Section 3.8.

From the obtained beat-wise bass pitch estimates, we compute the accuracy A as the percentage of correctly estimated pitch values. Similarly, we compute the accuracy A_{PC} by considering only the correct pitch class and disregarding the octave information. Both measures are computed on a frame-level, taking only voiced frames with available ground truth annotation into account, and on a beat-level, taking only beats with a ground-truth annotation into account.

[4] It must be noted that the algorithm SG is limited to a two-octave pitch range between the MIDI pitch values 21 and 45 (f_0 values between 27.5 Hz - 110 Hz).

5 Results

Table 3 summarizes the accuracy values for frame-wise and beat-wise evaluation across the three state-of-the-art algorithms and the 6 proposed models. First, it can be observed that the beat-informed late fusion step discussed in Section 3.8 significantly improves accuracy values across all algorithms. Second, the DNN-based algorithms clearly outperform the state of the art algorithms. One important reason is that the state-of-the-art algorithms are not at all tailored towards jazz music while the proposed models are trained on music recordings with similar music style as the test data. Third, by disregarding octave errors, accuracy values consistently rise by around 5 % to 10 %.

With respect to the proposed data augmentation methods, adding additional training data using the semi-supervised training procedure described in Section 3.6 does not improve the accuracy values for the given transcription task. However, we found that the predictions obtained from all variants of model M_2, which incorporate additional training data, show higher sparseness values than the predictions obtained from the initial model M_1. The last column of Table 3 shows the average sparseness values of the pitch salience predictions from all proposed models for the test set. It can be observed that the average sparseness increases with decreasing threshold τ^q.

This is also apparent in Figure 5, which shows the predicted pitch salience representations from all compared algorithms for the bass line excerpt from 0:04 to 0:09 of Chet Baker's Solo on "Let's Get Lost". We interpret

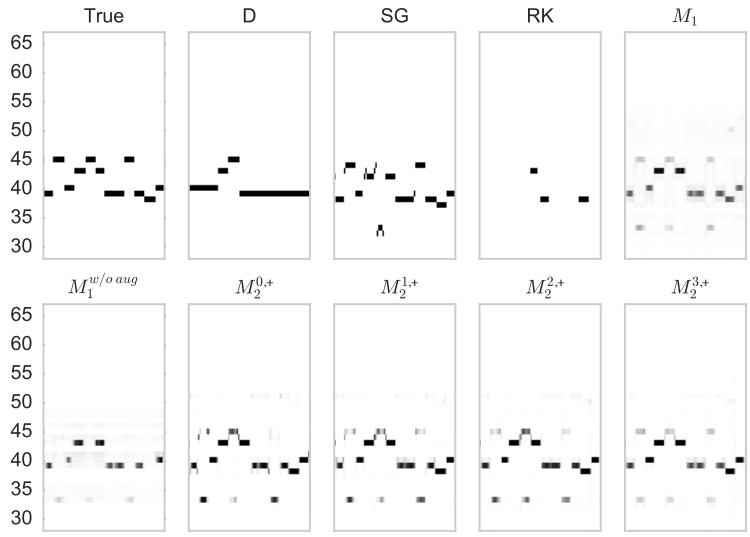

Fig. 5: Excerpts from bass pitch salience representations for excerpt from 0:04 to 0:09 of Chet Baker's Solo on "Let's Get Lost". "True" illustrates the ground truth pitch salience. Pitch salience matrices for deep learning methods are squared for better visibility. The MIDI pitch is shown on the vertical axis.

this as an indicator that these models are more confident in predicting the pitch salience of monophonic bass lines and show less confusion to other pitches. While this is not beneficial in the given transcription task, we belief that it has potential to improve source separation algorithms.

6 Conclusions

In this paper, we presented a transcription algorithm for walking bass lines in jazz recordings based on deep neural networks. We adapted a method from the literature to learn a mapping from a mixture spectrogram to a bass salience representation. By using manually tapped beat times as additional clues, we applied late-fusion to extract beat-wise pitch values, which are a musically meaningful representation of a walking bass line. We investigated two data augmentation techniques to increase the training dataset of the model. First, we added pitch-shifted versions of the initial training set and second, we applied a semi-supervised learning scheme, where we selected predictions of the model on unlabeled data as additional training data. Using pitch shifting as data augmentation technique clearly improves the results. The proposed semi-supervised learning leads to models that generate sparser pitch salience predictions. While this property does not increase the transcription accuracy, we believe that it can help to improve the performance of DNN-based source separation algorithms.

Acknowledgements The Jazzomat Research Project is supported by the German Research Foundation (DFG-PF 669/7-2). S. Balke and M. Müller are supported by DFG MU 2686/6-1. The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Fraunhofer-Institut für Integrierte Schaltungen IIS. The authors would like to thank Stylianos Ioannis Mimilakis for fruitful discussions and Justin Salamon, Christian Dittmar, and Anssi Klapuri for providing reference transcriptions of our test set. A special thank goes to all jazz and musicology students who transcribed the walking bass lines.

References

[1] Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., and Klapuri, A., "Automatic music transcription: challenges and future directions," *J. of Intelligent Inf. Systems*, pp. 1–28, 2013.

[2] Abeßer, J., Frieler, K., Pfleiderer, M., and Zaddach, W.-G., "Introducing the Jazzomat project - Jazz solo analysis using Music Inf. Retrieval methods," in *Proc.*

of the Int. Symposium on Computer Music Multidisciplinary Research (CMMR) Sound, Music and Motion, pp. 187–192, Marseille, France, 2013.

[3] Uhlich, S., Giron, F., and Mitsufuji, Y., "Deep neural network based instrument extraction from music," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2135–2139, Brisbane, Australia, 2015.

[4] Balke, S., Dittmar, C., Abeßer, J., and Müller, M., "Data-Driven Solo Voice Enhancement for Jazz Music Retrieval," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017.

[5] Dittmar, C., Lehner, B., Prätzlich, T., Müller, M., and Widmer, G., "Cross-Version Singing Voice Detection in Classical Opera Recordings," in *Proc. of the Int. Society for Music Inf. Retrieval Conf. (ISMIR)*, pp. 618–624, Málaga, Spain, 2015.

[6] Abeßer, J., Balke, S., Frieler, K., Pfleiderer, M., and Müller, M., "Accompanying Website: Deep Learning for Jazz Walking Bass Transcription," `http://www.audiolabs-erlangen.de/resources/MIR/2017-AES-WalkingBassTranscription/`, 2017.

[7] Dittmar, C., Dressler, K., and Rosenbauer, K., "A Toolbox for Automatic Transcription of Polyphonic Music," *Proc. of the Audio Mostly Conf.*, pp. 58–65, 2007.

[8] Goto, M., "A Real-Time Music-Scene-Description System - Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals," *Speech Communication*, 43(4), pp. 311–329, 2004.

[9] Salamon, J., Serrà, J., and Gómez, E., "Tonal Representations for Music Retrieval: From Version Identification to Query-by-Humming," *Int. J. of Multimedia Inf. Retrieval*, 2, pp. 45–58, 2013.

[10] Ryynänen, M. P. and Klapuri, A., "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music J.*, 32, pp. 72–86, 2008, ISSN 1520-6149.

[11] Tsunoo, E., Ono, N., and Sagayama, S., "Musical Bass-Line Pattern Clustering and its Application to Audio Genre Classification," in *Proc. of the Int. Society for Music Inf. Retrieval Conf. (ISMIR)*, pp. 219–224, Kobe, Japan, 2009.

[12] Hainsworth, S. W. and Macleod, M. D., "Automatic bass line transcription from polyphonic music," in *Proc. of the Int. Computer Music Conf. (ICMC)*, pp. 431–434, La Habana, Cuba, 2001.

[13] Tsunoo, E., Tzanetakis, G., Ono, N., and Sagayama, S., "Beyond Timbral Statistics: Improving Music Classification Using Percussive Patterns and Bass Lines," *IEEE Trans. on Audio, Speech, and Language Processing*, 19(4), pp. 1003–1014, 2011.

[14] Müller, M., *Fundamentals of Music Processing*, Springer Verlag, 2015, ISBN 978-3-319-21944-8.

[15] Salamon, J. and Gómez, E., "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics," *IEEE Trans. on Audio, Speech, and Language Processing*, 20(6), pp. 1759–1770, 2012.

[16] Böck, S. and Schedl, M., "Polyphonic piano note transcription with recurrent neural networks," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 121–124, Kyoto, Japan, 2012.

[17] Kelz, R., Dorfer, M., Korzeniowski, F., Böck, S., Arzt, A., and Widmer, G., "On the Potential of Simple Framewise Approaches to Piano Transcription," in *Proc. of Int. Society for Music Inf. Retrieval Conf. (ISMIR)*, New York, USA, 2016.

[18] Sigtia, S., Benetos, E., and Dixon, S., "An End-to-End Neural Network for Polyphonic Piano Music Transcription," *IEEE Trans. on Audio, Speech, and Language Processing*, 24(5), pp. 927–939, 2016.

[19] Cannam, C., Jewell, M. O., Rhodes, C., Sandler, M., and d'Inverno, M., "Linked Data And You: Bringing music research software into the Semantic Web," *J. of New Music Research*, 39(4), pp. 313–325, 2010.

[20] McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., and Nieto, O., "librosa: Audio and Music Signal Analysis in Python," in *Proc. of the Scientific Computing with Python conference (Scipy)*, Austin, Texas, 2015.

[21] Erdogmus, D., Fontenla-Romero, O., Principe, J. C., Alonso-Betanzos, A., and Castillo, E., "Linear-least-squares initialization of multilayer perceptrons through backpropagation of the desired response," *IEEE Trans. on Neural Networks*, 16(2), pp. 325–337, 2005.

[22] Zeiler, M. D., "ADADELTA: An Adaptive Learning Rate Method," *CoRR*, abs/1212.5701, 2012.

[23] Korzeniowski, F. and Widmer, G., "Feature Learning for Chord Recognition: The Deep Chroma Extractor," in *Proc. of the Int. Society for Music Inf. Retrieval (ISMIR)*, New York, USA, 2016.

[24] Hoyer, P. O., "Non-negative Matrix Factorization with Sparseness Constraints," *J. of Machine Learning Research*, 5, pp. 1457–1469, 2004.