Predominant Jazz Instrument Recognition: Empirical Studies on Neural Network Architectures

Jakob Abeßer
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany
jakob.abesser@idmt.fraunhofer.de

Jaydeep Chauhan
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany

Prateek Pradeep Pillai
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany

Michael Taenzer
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany

Stylianos I. Mimilakis
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany

***Abstract*—Musicological studies on jazz performance analysis commonly require a manual selection and transcription of improvised solo parts, both of which can be time-consuming. In order to expand these studies to larger corpora of jazz recordings, algorithms for automatic content analysis can accelerate these processes. In this study, we aim to detect the presence of predominant music instruments in jazz ensemble recordings. This information can guide a structural analysis in order to detect improvised solo parts. As the main contribution, we perform a comparative study on predominant automatic instrument recognition (AIR) in jazz ensembles using a taxonomy of 11 common instruments including singing voice. We compare the performance of three state-of-the-art convolutional neural networks (CNNs) including a recurrent variant and one with an attention mechanism. Our main finding is that while all networks perform comparably, the attention-based model learns the most compact feature representation as it is by orders of magnitude smaller than the other models.**

***Index Terms*—automatic instrument recognition, convolutional neural networks, deep learning, attention, jazz analysis**

I. Introduction

Automatic Instrument Recognition (AIR) is one of the central tasks in Music Information Retrieval (MIR). By identifying the active instruments in a music recording, AIR can further guide algorithms for source separation [1] or music transcription [2]. An automated analysis of music recordings can support and facilitate large-scale musicological research such as in music performance analysis.

Algorithms for AIR face several challenges such as the spectral overlap between simultaneously played instruments as well as the large variety of instrument sounds caused by different sound production mechanisms. When analyzing jazz recordings, two specific challenges exist. As a first challenge, the recording quality of early jazz recordings from the beginning of the 20th century is poorer than in contemporary recordings. This directly affects the data distribution of derived audio features, which are processed by classification algorithms such as deep neural networks. In the research field of sound event detection (SED), this phenomenon is known as covariate shift and its compensation is subject to current research [3]. A second challenge arises from the inherent hierarchy of instruments [4] in jazz ensemble recordings. While predominant solo instruments such as trumpet or saxophone are audible in the foreground, rhythm section instruments such as piano, double bass, and drums often remain in the background. In this paper, we focus on predominant instrument recognition in jazz recordings and extend the instrument taxonomy previously investigated by Gomez et al. [5] from six to eleven instruments.

As the main contributions of this paper, we evaluate three variants of convolutional and convolutional recurrent neural networks previously used for AIR and sound event detection (SED) for the purpose of instrument recognition in jazz ensemble recordings. We conduct systematic experiments to determine the influence of different data splitting strategies between training and test sets as well as threshold techniques to obtain the final predictions. This paper is structured as follows: We first review recent deep-learning based AIR algorithms in Section II. Section III provides details about the data processing pipeline and the evaluated neural network architectures. Section IV discusses the applied datasets before Section V explains the experimental procedure and summarizes the results. Finally, Section VI concludes this work.

II. Related Work

Traditional AIR algorithms included audio pre-processing, feature extraction, classification and post-processing of prediction results [6]. Such algorithms combined domain expert knowledge with signal processing and machine learning techniques. In contrast, recent AIR algorithms focus on signal representations, which are learnt in a data-driven fashion from large datasets using deep neural networks.

The first CNN-based AIR algorithms were introduced by Park & Lee [7] and Li et al. [8]. The former model uses both magnitude and phase information as input, whereas the latter processes raw audio in an end-to-end learning fashion. A CNN architecture for predominant instrument recognition, which

includes multiple pairs of convolutional layers without intermediate max pooling as inspired by the VGGNet architecture, was proposed by Han et al. [9]. This model was further used by Gomez et al. [5] for solo instrument recognition in jazz ensemble recordings. The authors investigated harmonic/percussive and solo/accompaniment source separation as pre-processing prior to the instrument recognition stage. Taenzer et al. also built upon this model and systematically investigated the influence of data augmentation and normalization strategies for the use-case of instrument family recognition in classical recordings [10].

A current research trend is to approach AIR jointly with additional classification tasks in a multitask learning fashion. Yu et al. grouped instruments based on similar onset characteristics or instrument family membership to jointly predict the instrument as well as its corresponding group [11]. Hung & Yang approached AIR in classical ensemble recordings using a CNN model that jointly learns to predict the activity of instruments and pitches [12]. Another trend is to investigate alternative signal representations for AIR such as the Hilbert Spectral Analysis (HSA) [13] and the Hilbert-Huang Transform [14].

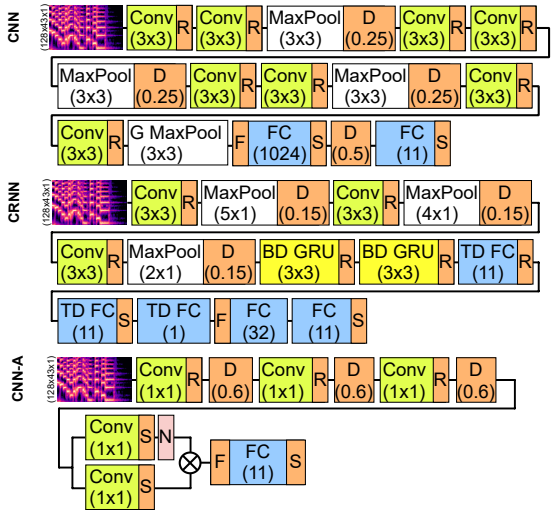

Fig. 1. Comparison of three model architectures CNN, CRNN, and CNN−A (from top to bottom). Abbreviations used: Convolutional layer (Conv), Max pooling layer (MaxPool), ReLU (R), Dropout (D), Flatten (F), Fully-connected layer (FC), Sigmoid (S), Normalization (N), Bi-directional Gated Recurrent Unit (BD GRU), Time-distributed (TD).

III. Methodology

Our AIR approach consists of two steps. In the first step, music signals are pre-processed and converted into a two-dimensional signal representation as described in Section III-A. This representation is then processed by a deep neural network (compare Section III-B), which computes predictions concerning the activity of different predominant jazz instruments. Finally, the predicted instrument activities are post-processed as discussed in Section III-C.

A. Audio Representation & Pre-processing

We resample every audio file at 22.05 kHz and normalize the yielded signal to a maximum absolute amplitude of 1. We then compute a 128-band mel-spectrogram[1] of each audio signal using a window and a hop size of 2048 and 512 samples, respectively. Following [9], we apply logarithmic magnitude scaling to each computed mel-spectrogram and split it into one second long sub-sequences, i. e., patches of 43 (time) frames. Based on the findings presented in [10], every patch is further standardized using the global mean and standard deviation.

B. Neural Network Architectures

Our study considers three neural network architectures based on convolutional neural networks (CNN). We compare a regular CNN architecture (CNN) [9] and one with an attention mechanism [16] (CNN−A), which were both previously used for AIR. As a third architecture, we investigate a convolutional recurrent neural network (CRNN) originally proposed for sound event detection [17]. The model architectures are illustrated in Figure 1 and are detailed below. Despite their architectural differences, all models share a final fully-connected feed-forward (dense) layer that uses the sigmoid activation function to predict (weak) instrument activity labels on a song-level.

1) CNN [9]: The original CNN model by Han et al. [9] includes 4 pairs of convolutional layers with symmetric 3x3 filters. The intermediate max pooling operations of size 3x3 implement a feature abstraction across time and frequency. At the same time, the number of filters increases from 32 to 256. Finally, after a global max pooling layer, two dense layers are used to compute the final class predictions.

2) CRNN [17]: In the CRNN model proposed by Adavanne & Virtanen [17], the front-end includes three convolutional layers and intermediate max pooling operations, which operate solely across frequency to retain the initial temporal resolution. In the back-end, the output of two bi-directional gated recurrent unit layers are first processed by three time-distributed dense layers before the temporal context is flattened and two additional dense layers generate the final predictions. The second network output branch for generating strong label predictions is neglected here.

3) CNN-A [16]: The third model CNN−A [16] combines a CNN architecture with an attention mechanism. The model front-end consists of three convolutional layers with 128 1x1 filters, ReLU activation functions, and an intermediate dropout of 0.6. This way, the original spectrogram is transformed into an 128-dimensional embedding space while maintaining the original time-frequency resolution. In the back-end of the network, both the instance-level scores and attention-level weights are computed in two parallel branches before they are combined. The main intuition of this attention mechanism is to enable a weighted aggregation of frame-level predictions based on the internal embedding vectors. The resulting tag-level predictions allow for a training using weakly-labeled

[1]Mel-spectrograms are computed using the *librosa* library [15].

Model	Number of Parameters
CNN	1,449,963
CRNN	522,502
CNN-A	52,769

TABLE I
NUMBER OF PARAMETERS FOR EACH CNN ARCHITECTURE.

Strategy	Datasets			
	IRMAS	MedleyDB	WJD/JSD	DTL
DS1	Training (80 %) - Validation (20 %)			Test
DS2	Training (100 %)		Training (70 %) - Validation (30 %)	Test

TABLE II
DATASET SPLIT STRATEGIES AS DISCUSSED IN SECTION IV.

datasets.

It must be noted that in the original study [16], the authors used blocks of 10 VGGish embedding [18] frames as input to the attention-based CNN, which correspond to a temporal context of 9.6 seconds of audio. As discussed in Section III-A, the spectrogram patches used in our experiments only cover a temporal context of around one second of audio.

C. Post-processing

We use the "S2" evaluation strategy used by Han et al. [9] for late fusion from frame-level to recording-level predictions $p \in \mathbb{R}^N$ $(0 \leq p_i \leq 1)$ with N denoting the number of instrument classes. For this purpose, frame-level predictions are averaged over the duration of each audio recording and then normalized by their maximum value. We use a threshold τ to obtain recording-level binary instrument activity predictions $a \in \mathbb{Z}^N$ as

$$a_i = \begin{cases} 0 & p_i < \tau_i \\ 1 & \text{else.} \end{cases} \quad (1)$$

We consider two approaches to derive the threshold τ: In the *fixed threshold* approach, we use $\tau = 0.5$ for all classes. In the *variable threshold* approach, we determine the optimal threshold for each class to maximize the F-score of this class on the validation set.

IV. DATASETS & TAXONOMY

A. Source Datasets

In this section, we will briefly review four existing AIR datasets, from which we compiled our task-specific datasets.

1) IRMAS: The Instrument Recognition in Music Audio Signals (IRMAS) dataset was created with a focus on predominant instrument recognition [19]. We used data from both the original training set (single-labeled) and test sets (multi-labeled). The dataset includes music recordings from various decades across the past century which naturally differ in recording quality. Here, we use recordings of the instrument classes clarinet, flute, trumpet, and vocals.

2) MedleyDB: MedleyDB 2.0 is a dataset of royalty-free multitrack music recordings [20]. The dataset covers a wide distribution of genres and primarily consists of full-length songs with professional or near-professional audio quality. All recordings are multi-labeled. We use recordings of the instrument classes clarinet, tenor saxophone, flute, trumpet, vibraphone, trombone, and vocals.

3) WJD/JSD: The Weimar Jazz Dataset (WJD) was published by the Jazzomat Research Projekt [21] and includes manual solo melody transcriptions of 456 jazz solos in commercial jazz recordings of various epochs. In a follow-up work, Balke et al. [22] compiled the Jazz Structure Dataset, which includes structural annotations (chorus boundaries) as well as chorus-level instrument activity annotations.

4) DTL: As a second dataset of solo sections from jazz ensemble recordings, we use a subset of 607 files from the "The DTL1000 Jazz Solo Dataset". This dataset was provided to us by the Dig That Lick research project[2] and includes recordings from all relevant epochs of jazz history. In our experiments, we used a subset of the DTL audio recordings database with annotations of the predominant solo instruments. Both the WJD/JSD and DTL datasets include all instrument classes considered in this work (compare Sec. IV-B).

B. Instrument Taxonomy & Data Distribution

In [23], Gomez et al. investigated AIR in jazz recordings for the six instruments trumpet (*tp*), clarinet (*cl*), trombone (*tb*), alto saxophone (*as*), tenor saxophone (*ts*), and soprano saxophone (*ss*). In this paper, we extend this selection to eleven instruments by adding the instrument classes baritone saxophone (*bs*), flute (*flu*), vibraphone (*vib*), cornet (*cor*), and (male and female) singing voice (*voi*). Compared to other instruments, the singing voice is used less often as a solo instrument, but still plays an important role in jazz.

The total duration of annotated audio recordings per dataset and instrument class is summarized in Fig. 2. One can observe that the data distribution is heavily imbalanced. For instance, only four of the eleven instrument classes (*cla*, *flu*, *tp*, *voi*) can be found in all four datasets while the four classes *ss*, *as*, *bs*, and *cor* are only included in the WJD/JSD and DTL datasets. Furthermore, the total duration of available data varies strongly across different instruments, with fewer data for the instruments *ss*, *bs*, and *cor* and significantly more data for the instruments *ts*, *tp*, and *voi*. To a certain extent, this correlates to their common frequency of appearance as solo instruments in jazz recordings.

V. EVALUATION

A. Data Split Strategies

As shown in Tab. II, we compare two strategies on splitting the four datasets into training, validation, and test sets. The DTL dataset is consistently used solely as test set to compare

[2]http://dig-that-lick.eecs.qmul.ac.uk/

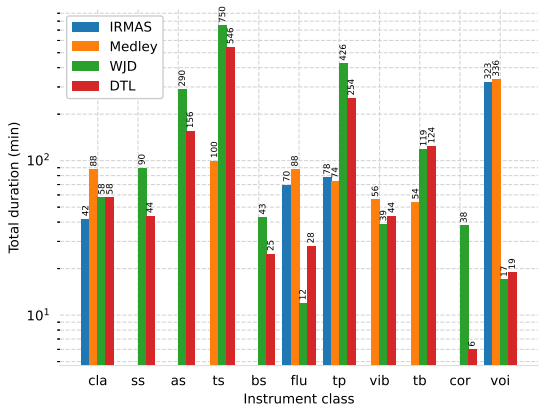

Fig. 2. Overall duration (min) of annotated audio recordings per dataset and per instrument class. The instrument class abbreviations are introduced in Section IV-B.

both strategies. For strategy `DS1`, we randomly shuffle the IRMAS, MedleyDB, and the WJD/JSD datasets and use 80% as training data and the remaining 20% as validation data. In contrast, for strategy `DS2`, we derive a validation set from 30% of the WJD dataset and use the remaining 70% as well as the full IRMAS and MedleyDB datasets as training data. The intuition behind strategy `DS2` is to use a validation set that resembles the musical content of the test set (DTL), i. e., ensemble recordings from all jazz epochs and covering all instruments considered in this work.

B. Experimental Procedure

1) Neural Network Training: We use the Adam optimizer [24] to train all neural networks. An initial learning rate of 10^{-4} is halved every five epochs. We use early stopping if no improvement on the validation loss is observed for 20 epochs. The total number of epochs is set to 400. Binary crossentropy is used as loss function since all datasets are multi-labeled. For each experimental configuration, one model training is performed.

2) Threshold Optimization: We compare two thresholding strategies. As a first strategy, we use a fixed threshold of $\tau_c = 0.5$ for all classes. As a second strategy, we optimize the decision thresholds for each class c based on the validation set. We select τ_c by maximizing the F-score of class c on the validation set using a grid search between $\tau \in [0, 1]$ with a step-size of 0.001.

C. Metrics

We compute both the micro-averaged F-score F_{micro}, which averages performance over all test items, and the macro-averaged F-score F_{macro}, which averages over class-wise F-scores due to the class imbalance discussed in Section IV.

D. Results

Table III summarizes the evaluation metrics, from which we make the following observations. First, the class imbalance discussed in Section IV-B is confirmed, since the F_{micro} scores are on average 0.2-0.3 larger than the F_{macro} scores. As a second observation, the best performance of $F_{\text{micro}} = 0.81$ and $F_{\text{macro}} = 0.52$ was achieved for the fixed decision threshold and the data split strategy `DS2` using the `CNN` model. However, the lead is very small compared to the other models using the fixed decision threshold. Therefore, it must be stated that all three neural network architectures perform comparably well for the given AIR scenario. These results show that the `CNN-A` is most effective in learning suitable features for this AIR task, as it is by two orders of magnitudes smaller than the best-performing `CNN` model (compare Table I).

Data Split Strategy	Model	Fixed Threshold		Variable Threshold	
		F_{micro}	F_{macro}	F_{micro}	F_{macro}
`DS1`	`CNN`	0.80	0.50	0.73	0.49
	`CNN-A`	0.80	0.48	0.73	0.49
	`CRNN`	0.80	0.50	0.72	0.51
`DS2`	`CNN`	**0.81**	**0.52**	0.60	0.46
	`CNN-A`	0.77	0.47	0.65	0.46
	`CRNN`	0.81	0.47	0.64	0.46

TABLE III
SUMMARY OF F-SCORE RESULTS FOR BOTH DATA SPLIT STRATEGIES AND BOTH TYPES OF DECISION THRESHOLDS.

As a third observation, using a fixed decision threshold consistently outperformed variable (class-dependent) decision thresholds. For the variable threshold case, data split strategy `DS1` clearly outperforms `DS2`. Since the class-wise decision thresholds are optimized on the validation sets, this indicates that using the WJD/JSD dataset as a "representative" validation set for the jazz ensemble recordings in the DTL test set (`DS2`) is not the better strategy here. As a final observation, Table IV shows that the recognition performance varies strongly between different instrument classes. For many instrument classes such as *as*, *bs*, *flu*, and *vib*, the F-scores using the fixed threshold are significantly bigger than for the variable thresholds. To our surprise, the performance of the frequently appearing classes *ts* and *tp* is clearly lower compared to the other instruments with F_{micro} values of around 0.5.

VI. Conclusion

In this paper, we report on a comparative study of three convolutional neural network architectures for the task of predominant instrument recognition in jazz ensemble recordings. Based on short one-second long excepts of mel-spectrograms, the networks were evaluated for predicting the instruments' activity. In our experiments, we furthermore investigate two dataset split strategies differing in the composition of the validation set and two thresholding approaches to binarize the network predictions. While our results show that all models perform comparatively well, the smallest convolutional neural network variant, which includes an attention mechanism, learns the most efficient feature representation for the given task. Furthermore, we find that a fixed decision threshold

Instrument Class	Fixed Threshold		Variable Threshold	
	F_{micro}	F_{macro}	F_{micro}	F_{macro}
Clarinet (*cla*)	0.83	0.59	0.80	0.61
Soprano Saxophone (*ss*)	0.95	0.49	0.73	0.46
Alto Saxophone (*as*)	0.71	0.56	0.38	0.37
Tenor Saxophone (*ts*)	0.53	0.43	0.56	0.50
Baritone Saxophone (*bs*)	0.96	0.49	0.35	0.29
Flute (*flu*)	0.93	0.50	0.29	0.25
Trumpet (*tp*)	0.48	0.47	0.50	0.50
Vibraphone (*vib*)	0.94	0.49	0.76	0.53
Trombone (*tb*)	0.80	0.60	0.70	0.58
Cornet (*cor*)	0.94	0.49	0.90	0.50
Singing Voice (*voi*)	0.81	0.58	0.61	0.44

TABLE IV
INSTRUMENT-LEVEL F-SCORE RESULTS FOR THE BEST PERFORMING CONFIGURATION OF THE CNN MODEL USING DATASET SPLIT STRATEGY DS2.

is preferable over variable decision thresholds, which are individually optimized for each instrument class. Future work must further investigate why particular instrument classes such as trumpet and tenor saxophone achieve lower recognition scores, although they appear more frequently in jazz recordings. Furthermore, a detailed ablation study and model inspection analysis potentially allows for a more detailed model comparison for the given task.

ACKNOWLEDGEMENTS

This work has been supported by the German Research Foundation (AB 675/2-1). We would like to express our gratitude to the members of the Jazzomat and the Dig That Lick project to share with us the WJD and DTL dataset. Furthermore, our thanks go to Stefan Balke, Julian Reck, Christof Weiß, and Meinard Müller for sharing the JSD dataset annotations.

REFERENCES

[1] O. Slizovskaia, G. Haro, and E. Gómez, "Conditioned Source Separation for Music Instrument Performances," *arXiv preprint arXiv:2004.03873*, 2020.

[2] Y.-N. Hung and Y.-H. Yang, "Frame-Level Instrument Recognition by Timbre and Pitch," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 135–142.

[3] K. Drossos, P. Magron, and T. Virtanen, "Unsupervised Adversarial Domain Adaptation based on the Wasserstein Distance for Acoustic Scene Classification," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2019, pp. 259–263.

[4] S. Essid, G. Richard, and B. David, "Instrument Recognition in Polyphonic Music Based on Automatic Taxonomies," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 68–80, 2006.

[5] J. Gomez, J. Abeßer, and E. Cano, "Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.

[6] M. Grasis, J. Abeßer, C. Dittmar, and H. Lukashevich, "A Multiple-Expert Framework for Instrument Recognition," *Sound, music, and motion. 10th International Symposium, CMMR 2013 : Marseille, France, October 15 - 18, 2013; Revised selected papers, Lecture Notes in Computer Science*, vol. 8905, pp. 619–634, 2014.

[7] T. Park and T. Lee, "Musical Instrument Sound Classification with Deep Convolutional Neural Networks using Feature Fusion Approach," *arXiv preprint arXiv:1512.07370*, 2015.

[8] P. Li, J. Qian, and T. Wang, "Automatic Instrument Recognition in Polyphonic Music Using Convolutional Neural Networks," *arXiv preprint arXiv:1511.05520*, 2015.

[9] Y. Han, J. Kim, and K. Lee, "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 1, pp. 208–221, 2017.

[10] M. Taenzer, J. Abeßer, S. I. Mimilakis, C. Weiß, M. Müller, and H. Lukashevich, "Investigating CNN-Based Instrument Family Recognition for Western Classical Music Recordings," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 612–619.

[11] D. Yu, H. Duan, J. Fang, and B. Zeng, "Neural Network With Auxiliary Classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 852–861, 2020.

[12] Y.-N. Hung, Y.-A. Chen, and Y.-H. Yang, "Multitask Learning for Frame-level Instrument Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, may 2019, pp. 381–385.

[13] D. Kim, T. T. Sung, S. Y. Cho, G. Lee, and C. B. Sohn, "A single predominant instrument recognition of polyphonic music using CNN-based timbre analysis," *International Journal of Engineering Technology*, vol. 7, no. 3.34, pp. 590–593, 2018.

[14] X. Li, K. Wang, J. Soraghan, and J. Ren, "Fusion of hilbert-huang transform and deep convolutional neural network for predominant musical instruments recognition," in *Proceedings of the International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*, Seville, Spain, 2020, pp. 80–89.

[15] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference (SciPy)*, vol. 8, Austin, Texas, 2015, pp. 18–25.

[16] S. Gururani, M. Sharma, and A. Lerch, "An Attention Mechanism for Musical Instrument Recognition," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019, pp. 83–90.

[17] S. Adavanne and T. Virtanen, "Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events (DCASE)*, Munich, Germany, 2017.

[18] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 131–135.

[19] J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012, pp. 559–564.

[20] R. M. Bittner, J. Wilkins, H. Yip, and J. P. Bello, "MedleyDB 2.0: New Data and a System for Sustainable Data Collection," in *Late-Breaking Demo Session of the International Conference on Music Information Retrieval (ISMIR)*, New York, NY, USA, 2016.

[21] M. Pfleiderer, K. Frieler, J. Abeßer, W.-G. Zaddach, and B. Burkhart, Eds., *Inside the Jazzomat - New Perspectives for Jazz Research*. Schott Campus, 2017.

[22] S. Balke, J. Reck, C. Weiß, J. Abeßer, and M. Müller, "JSD: A dataset for structure analysis in jazz music," in *to be published*.

[23] J. S. Gómez, J. Abeßer, and Estefanía Cano, "Jazz Solo Instrument Classification with Convolutional Neural Networks, Source Separation, and Transfer Learning," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 577–584.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.