JOINT MULTI-PITCH DETECTION AND SCORE TRANSCRIPTION FOR POLYPHONIC PIANO MUSIC

Lele Liu, Veronica Morfi, and Emmanouil Benetos

Centre for Digital Music, Queen Mary University of London, UK

ABSTRACT

Research on automatic music transcription has largely focused on multi-pitch detection; there is limited discussion on how to obtain a machine- or human-readable score transcription. In this paper, we propose a method for joint multi-pitch detection and score transcription for polyphonic piano music. The outputs of our system include both a piano-roll representation (a descriptive transcription) and a symbolic musical notation (a prescriptive transcription). Unlike traditional methods that further convert MIDI transcriptions into musical scores, we use a multitask model combined with a Convolutional Recurrent Neural Network and Sequence-to-sequence models with attention mechanisms. We propose a Reshaped score representation that outperforms a LilyPond representation in terms of both prediction accuracy and time/memory resources, and compare different input audio spectrograms. We also create a new synthesized dataset for score transcription research. Experimental results show that the joint model outperforms a single-task model in score transcription.

***Index Terms*—** Automatic music transcription, sequence-to-sequence models, score transcription.

1. INTRODUCTION

Automatic music transcription (AMT), whose aim is to convert music audio recordings into some form of music notation, is a core problem in the field of music signal analysis [1]. It is usually considered as the music equivalent of automatic speech recognition (ASR), and can be useful for applications such as music search, music tutoring, music accompaniment, and building music archives. AMT is generally considered as a combination of several subtasks including multi-pitch detection, onset/offset detection, voice separation, rhythm quantisation, score formatting. A large part of work in AMT falls under the tasks of multi-pitch detection and onset/offset detection (e.g. [2, 3]), which are often referred to jointly as *note tracking*.

Due to the increased performance of note tracking systems, the problem of *audio-to-score transcription* (A2S), or *complete music transcription*, has started to attract research interest. A2S, targeting at estimating human- or machine-readable scores from music audio, is an under-explored problem and a challenging task because it includes other tasks beyond multi-pitch detection and onset/offset detection, such as meter estimation, rhythm quantisation, note duration estimation and voice separation.

The recent literature has mainly focused on two approaches for complete transcription: 1) *traditional* methods transcribe music audio step by step in the order of subtasks [4, 5], and 2) *end-to-end* methods design algorithms that directly convert an audio input to a score format in the form of Western staff notation [6, 7, 8, 9]. Compared to traditional methods, end-to-end methods do not face the risk of accumulated errors in different steps. The first attempt towards A2S with end-to-end methods adopts sequence-to-sequence models as presented in [6], which proves A2S can be performed in an end-to-end manner. However, the work is limited to monophonic or two-melody polyphonic music, constrained note durations and 4/4 time signatures. Another end-to-end A2S system developed in [7, 8, 9] makes use of a convolutional recurrent neural network (CRNN) with connectionist temporal classification (CTC) loss. It uses a CTC-friendly score data representation to achieve higher performance. However, the system is still limited to monophonic transcription or a simple case of polyphonic transcription (string quarters or Bach chorales). Another limitation among those above mentioned end-to-end A2S methods is that they only output a beat-quantised musical notation, lacking important descriptive information in music analysis such as absolute time for note onsets and offsets.

In this work, we extend the use of end-to-end methods for A2S to the more general application scenario of polyphonic piano music with varying polyphony levels. Furthermore, we support the estimation of music performance characteristics in a piano-roll format by using a multitask end-to-end model composed of convolutional and recurrent neural networks and sequence-to-sequence models with attention mechanism.

Our main contributions are listed as follows: 1) we propose the first holistic model that transcribes polyphonic piano music into both a piano-roll format (corresponding to a descriptive notation of the music audio) and a score in Western staff notation (corresponding to a prescriptive notation of the musical audio); 2) we propose a new score representation for modelling polyphonic music that learns and predicts 7 times faster and performs better than the LilyPond format [10] score representation on this model; 3) we evaluate the effect of using different input time-frequency representations, including the Short Time Fourier Transform, Mel Spectrogram, Constant-Q Transform, Variable-Q Transform [11] and Harmonic Constant-Q Transform [12]; and 4) to allow end-to-end training for an A2S system, we create and publish a new dataset with collected score files from MuseScore and synthesised music recordings for A2S research.

2. DATA REPRESENTATIONS

2.1. Audio data representations

Like in most AMT systems, we use as input a time-frequency representation of the audio signal, and we compare commonly used time-frequency representations. All representations are log-valued and the signals are resampled to ensure the hop size of every spectrogram being equal to 10ms, which means equal length in the model input. The time-frequency representations we compare are:

L. Liu is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by the China Scholarship Council and Queen Mary University of London.

Thanks to Mary Pilataki-Manika for her invaluable feedback and Simin Yang for proofreading.

- *STFT* - Magnitude spectrogram from the Short Time Fourier Transform with a Hanning window and FFT window length in {1024, 2048}. Signal sampling rate is 44.1kHz.
- *Mel Spectrogram* - Mel Spectrogram with different FFT window length in {1024, 2048} and different number of Mel bands in {128, 192, 256}. Signal sampling rate is 44.1kHz.
- *Constant-Q Transform (CQT)* - Spectrogram obtained from the Constant-Q Transform [13], with bins per octave in {12, 24, 36, 48, 60}, number of octaves in {7, 8} and lowest frequency equal to pitch A0=27.5 Hz, which is the lowest pitch in piano. Signals are resampled at 25.6 kHz to fit a hop length of 256.
- *Harmonic Constant-Q Transform (HCQT)* - Spectrogram from the Harmonic Constant-Q Transform proposed in [12], which is a 3-dimensional spectrogram with CQTs based on shifted harmonics. The parameters we select from are bins per octave in {36, 60}, number of octaves in {5, 6} and number of harmonics in {4, 5, 6}. Signals are resampled at 25.6kHz.
- *Variable-Q Transform (VQT)* - Spectrogram calculated from Variable-Q Transform proposed in [11]. We select γ values in {10, 20, 30}, number of bins per octave in {36, 60} and number of octaves in {7, 8}. Signals are resampled at 25.6kHz.

2.2. Score data representations

One of the major challenges in developing an end-to-end A2S system is selecting an output representation that can support polyphonic music and includes various cues present in music scores. Compared to sentence outputs in ASR tasks, music notation is much more structured and complex. Some of the most commonly used symbolic music score encoding formats are MusicXML, **Kern, LilyPond, ABC and PAE [14]. However, MusicXML is a verbose music encoding, and the **Kern format only supports monophonic music per voice. All the other three formats support polyphonic music and encode music scores into strings. Here, we use the LilyPond [10] format as a base representation for our score data representation.

Although much more concise than the MusicXML format, LilyPond encoding is still complex with hierarchical structures such as parts and voices. To make the transcription task simpler, we assume that there are only two hand parts in piano music and only one voice per hand, where each voice can have multiple concurrent notes. We consider the left hand part and right hand part scores as two outputs predicted jointly. In this way, we discard the hierarchical structure of the LilyPond format and keep the most essential information in two strings. We assume our model to predict one bar at a time, this means that we only take into account the notes and rests in our score data representation, no barline or key/time signature symbols are included. We do not consider playing techniques such as trios, vibratos nor rhythm structures such as triplets or quintuplets. The symbols we use in LilyPond format are:

- *Pitch* - Combined with pitch chroma (e.g. 'c' for C, 'cis' for $C\sharp$ and 'ces' for $C\flat$) and pitch height (e.g.' $'$ ' for higher octave and ' , ' for lower octave, duplicate e.g. ' $''$ ' for double octaves).
- *Duration* - We use numbers to represent durations, e.g. '8' for an 8th note duration (duration symbol can be omitted for a 4th note duration). The same duration representation is used for chords and rests. ' . ' is added for dotted notes - resulting in e.g. '4.'.
- *Rest/Note/Chord* - A rest is represented as 'r' followed by its duration symbol. A note/chord is represented by its pitch(es) and duration. Chord pitches are grouped by brackets (e.g. pitches for a C major starting with a middel C is '$\langle\ c'\ e'\ g'\ \rangle$')
- *Tie* - Ties are represented using '$\sim$', added to its start note, such as '$c4\ \sim\ c8$' for a tied note c, or '$\langle c\ \sim\ \ e\ g\rangle 4\langle c\ f\ a\rangle 2$' for a tied c in chord.

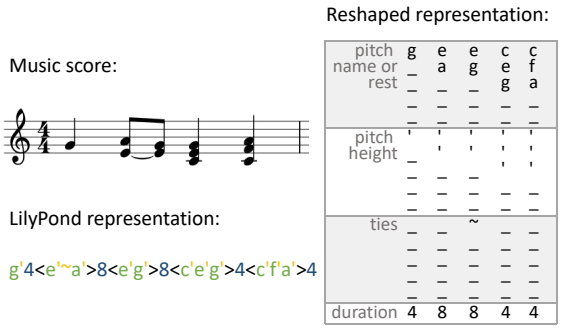

Fig. 1. Example music score and corresponding LilyPond and Reshaped representation.

Based on the above defined musical symbols, we compare the following two score data representations; an example for the two score representations can be seen in Figure 1.

- *LilyPond representation* - A representation based on LilyPond encoding by removing extra symbols and keeping only the described necessary symbols to reconstruct a musical score.
- *Reshaped representation* - Considering the length of a LilyPond score representation and the difficulty in learning structural information, we propose a Reshaped data representation based on the LilyPond representation that describes a score in a 2D matrix of symbols. We assume a maximum of five concurrent notes per hand, one for each finger, and the 2D matrix of symbols is indexed by symbol index and time, where each time step consists of (5+5+5+1=)16 symbols corresponding to five symbols for each one of pitch names or rest, pitch heights, ties and one symbol for duration. Each column of the matrix can reconstruct a rest, note or a chord in a music score.

3. MODEL ARCHITECTURE

Based on the data representations described above, we build a model[1] that predicts a piano-roll Y_p and a music score representation $Y_s = \{S_r, S_l\}$ from a music audio spectrogram input X, where S_r and S_l are the sequences of score symbols for the right and left hand parts. That is to find:

$$Y_p, \{S_r, S_l\} = \underset{Y_p \in \mathcal{U}_p, S_r \in \mathcal{U}_s, S_l \in \mathcal{U}_s}{\operatorname{argmax}} P(Y_p, \{S_r, S_l\}|X) \quad (1)$$

where $\mathcal{U}_p$ is the universal set of possible piano-rolls and $\mathcal{U}_s$ is the universal set of score representation sequences for one hand part. Although Y_p and X are strictly time-aligned, S_r, S_l are not aligned with X and have different lengths.

We design the model as a multitask learning model [15] with a shared convolutional stack and three separate sequential models, corresponding to Y_p, S_r, and S_l. The convolutional stack of the model has four convolutional layers, with a pooling layer in the middle of the stack following the second convolutional layer, and a linear layer following the last convolutional layer. We use two layers of bidirectional Gated Recurring Units (GRU) followed by a linear output layer with ELU activation for the piano-roll prediction, resulting in a Convolutional Recurrent Neural Network (CRNN) together with the shared convolutional stack. For scores, since we cannot ensure the

[1] https://github.com/cheriell/ICASSP2021-A2S

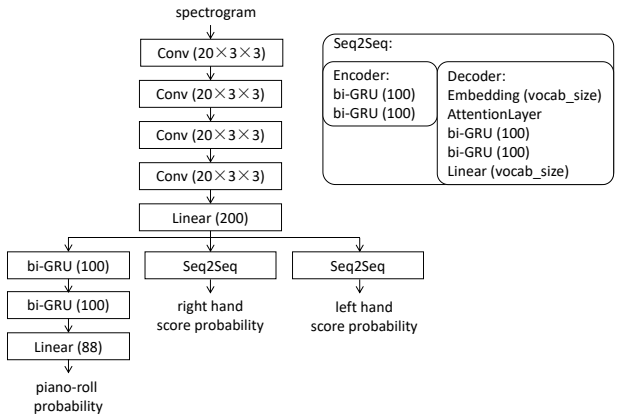

Fig. 2. Model structure with layer sizes.

LilyPond score representation is always shorter than the input audio spectrogram, which is a pre-requisite of using a CTC loss, we adopt the more universal sequence-to-sequence model originally proposed in neural machine translation [16], and use an attention mechanism [17] to make the model capable of learning long sequences. Two sequence-to-sequence models are added following the shared convolutional stack, for the right hand score S_r and the left hand score S_l respectively. The encoder of a sequence-to-sequence model is composed of two bi-directional GRU layers, and the decoder consists of one (for LilyPond score representation, or 16 for Reshaped score representation) embedding layer(s), one attention layer that attends to the encoder output, two one-directional GRU layers and one (for LilyPond representation, or 16 for Reshaped representation) linear output layer(s) with log-valued softmax activation. A diagram explanation of our proposed model architecture and the dimensions of each layer can be found in Figure 2.

4. EXPERIMENTS

4.1. Data

To obtain data to develop our model, we create a synthesized dataset[2] using scores collected from the MuseScore website[3]. We do this as a starting point and because there is a lack of AMT datasets that provide score ground truth on both physical and musical time. A dataset that best fits this task is the recently published ASAP dataset [18], which will be investigated as future work. We collect scores in MusicXML format, convert them to MIDI files, and synthesize audio files using four piano models using the Native Instruments Kontakt Player[4]. The scores we collect cover various key and time signatures, tempos, modes and polyphony levels, but do not contain grace notes, triplets, arpeggios, trios or other complex playing techniques. Some statistics on the dataset are in Table 1.

We use a train/validation/test ratio of 8:1:1 in our experiments, which corresponds to 168 pieces for training, 21 pieces for validation and 21 pieces for testing. Among the four piano models in the dataset, we use three piano models in training and validation, and all four piano models in testing. We keep the same train/validation/test split throughout all experiments.

[2]MuseSyn dataset: https://zenodo.org/record/4527460

[3]https://musescore.com/hub/piano

[4]https://www.native-instruments.com/en/products/komplete/samplers/kontakt-6-player/

Table 1. Dataset Statistics. For polyphony levels, the numbers out of brackets are calculated without adding piano pedals, and the numbers in brackets are calculated with piano pedals.

Number of music pieces	210
Total hours	9.62 $\times$ 4 piano models
Total notes	222,219
Use of piano pedal	29% (61 pieces)
Maximum polyphony level	13 (26)
Average polyphony level	2.87 (3.21)
Time signatures	4/4, 3/4, 5/4, 6/8, 9/8, etc.
Key signatures	5 flats $\sim$ 6 sharps

4.2. Training and inference

In our experiments, all the input audio spectrograms are calculated using librosa [19]. Python package pretty_midi [20] is used to extract information from MIDI files. We use the default velocity-valued piano-roll in pretty_midi as our piano-roll reference, and the ground truth downbeat times to cut audio recordings into bars. Zero padding is added to the input audio spectrograms and piano-rolls. For score representations, we split them into lists of symbols and consider the symbols as tokens in a sentence like tokens in natural language processing problems [21]. SOS, EOS and PAD symbols are added to the sentences. Symbols are encoded by one-hot encoding. For the Reshaped representation, the symbols are separately one-hot encoded, that is, we use separate index dictionaries for pitch name, pitch height, tie and duration symbols.

For the piano-roll part, a minimum squared error loss is used for the output during training. To obtain a binary piano-roll during inference, we threshold the model's output with a velocity value of 30, since low velocities are not audible. No post-processing steps like smoothing, minimum duration pruning, gap filtering are applied. The obtained note sequences form the binary piano rolls for note-level evaluations.

For the score part, a negative log-likelihood loss is used for the score representation. A 50% teacher forcing ratio [21] is used in the training process. This means the models use the ground truth in the decoding process with 50% chance, otherwise use the most probable output symbol. During inference, a 0% teacher forcing ratio is used, and we simply adopt a greedy decoding to obtain the predicted output sequence. To further obtain a full score transcription from the models, we combine the predicted scores for all bars, and post-process the scores by obtaining the most probable time signature, adding missing rests and removing extra rests. The final score representation can be directly decoded to MusicXML format.

4.3. Experimental setup

Comparison of time-frequency representations: In this experiment, we compare the performance on the use of the STFT, Mel Spectrogram, CQT, HCQT and VQT as input representations and their different parameters described in Section 2.1 for multi-pitch detection. We assume the best input representation for multi-pitch detection will perform well on score transcription, since the two tasks are highly related. Hence, we use only the piano-roll part of our proposed model, removing the pooling layer in the convolutional stack to achieve higher resolution. Results for this comparison are evaluated only on the three piano models used in model training.

Comparison of score representations: We use only the shared and score component to compare the two score representations we describe in Section 2.2, keeping the convolutional stack and the two sequence-to-sequence models for score prediction. We call the two

Table 2. F-measure of piano-roll prediction on different input representations and models. F_f: frame-level; F_{on}: note-level onset only; F_{onoff}: note-level onset and offset. The last two models use VQT as input, and are evaluated on all four pianos in the dataset.

Input representations/Models	F_f	F_{on}	F_{onoff}
STFT	89.5	81.0	61.7
Mel Spectrogram	89.0	82.1	63.0
CQT	**91.9**	85.4	67.4
HCQT	91.0	84.1	65.3
VQT	**91.9**	**85.7**	**68.5**
Piano-roll only	86.4	**67.6**	52.0
Joint	**88.0**	66.7	**53.6**

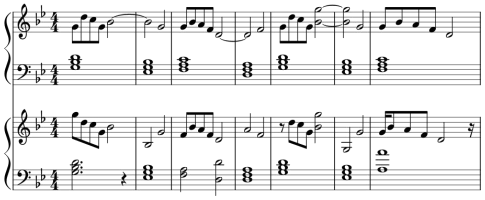

Fig. 3. Example transcription output. Upper grand staff: ground truth. Lower grand staff: transcription output, ground truth key signature is used for visualisation purposes.

models Score-only model with LilyPond or Reshaped representation. The best performing spectrogram from the previous comparison is used as our input representation.

Combination of piano-roll and symbolic score: Using the best input and score representation from the above, and the full multi-task model, we train a Joint model that simultaneously predicts a piano-roll transcription and a symbolic score transcription. We compare this new model with Piano-roll only and Score-only models to see how the two tasks influence each other.

4.4. Evaluation metrics

For the input representation experiment, we evaluate the performance with the AMT benchmark frame-level and note-level onset only and onset-offset evaluation metrics [22]. We use an onset tolerance of 50ms and keep the offset tolerance to be 20% of the note duration, or 50ms, which ever is larger, as is used in the MIREX public evaluations [22].

For score evaluation, since there is no existing standard A2S evaluation metric, we use the following two metrics as an indication of the system's performance:

- *Word error rate (WER)* of the LilyPond representation, adopted from neural machine translation tasks [21]. For the Reshaped representation, we first reconstruct the output to the LilyPond format and then calculate the word error rate to the ground truth LilyPond representation.
- *MV2H metric* proposed in [23] for complete AMT evaluation. We use four of the sub-metrics: multi-pitch detection accuracy (F_p), voice separation accuracy (F_{voi}), metrical alignment accuracy (F_{met}), note value detection accuracy (F_{val}). Submetric harmonic analysis is not included since we do not include key detection and chord estimation. The overall accuracy of this metric (F_{MV2H}) is the average over the four submetrics. In this work, we use the v1.0 of this metric, assuming our transcription and the input audio are time-aligned.

Table 3. Word error rates and MV2H results in percentage for different models. LilyPond: Score-only model with LilyPond representation; Reshaped: Score-only model with Reshaped representation; Joint: Joint model with Reshaped representation. Models evaluated on four pianos in the dataset.

WER	wer_{right}		wer_{left}		wer
LilyPond	38.0		39.0		38.5
Reshaped	37.8		**34.5**		**36.2**
Joint	**37.6**		35.3		36.5
MV2H	F_p	F_{voi}	F_{met}	F_{val}	F_{MV2H}
LilyPond	66.7	90.3	94.8	93.2	86.3
Reshaped	69.6	89.7	94.8	93.7	86.9
Joint	**71.1**	**90.8**	**94.9**	**94.4**	**87.8**

4.5. Results

Table 2 shows the F-scores of benchmark frame-level and note-level evaluation metrics on the model predictions on different input audio spectrograms with their best performing parameters, and on the Piano-roll only model and Joint model under the best input representation we find. Among the five spectrogram types, VQT shows the best performance, with a γ value of 20, and 8 octaves $\times$ 60 bins per octave in the frequency axis. No large performance difference on this metric is found between the Piano-roll only model and the Joint model. More detailed results on other parameters and precision/recall are described in our experiment report[5].

Table 3 shows the WER evaluation and MV2H evaluation results on score prediction for the two Score-only models and the Joint model we described in Section 4.3. All inputs are VQT spectrograms with 8×60 frequency bins and a γ value equal to 20. Results show the Reshaped representation is slightly better than the LilyPond representation in both metrics. The Reshaped representation also outperforms the LilyPond representation in terms of the time and memory resources required (around 7 times faster and half the memory). The Joint model performs better than the Score-only model with Reshaped representation in terms of MV2H evaluation metric, which suggests an advantage in adding the piano-roll prediction task.

One example ground truth score and its transcription from the Joint model is in Figure 3. In the example, the model does well in predicting meters and harmonies and can transcribe the melodies in general. Main errors include octave errors, note duration errors, extra/missing note errors and voice separation errors.

5. CONCLUSION

In this paper, we propose a model as a first step of joint multi-pitch detection and score transcription based on CRNN and Sequence-to-sequence models with an attention mechanism. We propose a matrix-like score data representation that is more efficient in learning and prediction compared to a LilyPond representation. We use the VQT spectrogram as the input representation for its best performance on multi-pitch detection among five types of commonly used representations. We train and evaluate our model on a new synthesized dataset we create for A2S research. The model we currently developed uses ground truth downbeat times to split bars and is not fully automatic. In future work, we will try our model in other datasets with real recordings (e.g. MAPS [24] and ASAP [18]) as well as use a beat tracker or similar to estimate downbeat times. Additionally, including skip connections between the piano-roll and score components can possibly help the tasks learn more from each other.

[5] https://cheriell.github.io/documents/report/A2S_report.pdf

6. REFERENCES

[1] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri, "Automatic music transcription: Challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[2] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck, "Onsets and frames: Dual-objective piano transcription," in *ISMIR, International Society for Music Information Retrieval Conference*, 2018, pp. 50–57.

[3] Rainer Kelz, Sebastian Bock, and Gerhard Widmer, "Deep Polyphonic ADSR Piano Note Transcription," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, vol. 2019-May, pp. 246–250.

[4] Eita Nakamura, Emmanouil Benetos, Kazuyoshi Yoshii, and Simon Dixon, "Towards Complete Polyphonic Music Transcription: Integrating Multi-Pitch Detection and Rhythm Quantization," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*. 2018, vol. 2018-April, pp. 101–105, IEEE.

[5] Kentaro Shibata, Eita Nakamura, and Kazuyoshi Yoshii, "Non-Local Musical Statistics as Guides for Audio-to-Score Piano Transcription," *arXiv preprint arXiv:2008.12710*, 2020.

[6] Ralf Gunter Correa Carvalho and Paris Smaragdis, "Towards End-to-End Polyphonic Music Transcription: Transforming Music Audio Directly to A Score," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2017, vol. 2017-Octob, pp. 151–155.

[7] Miguel A Román, Antonio Pertusa, and Jorge Calvo-Zaragoza, "An End-To-End Framework for Audio-To-Score Music Transcription on Monophonic Excerpts," in *ISMIR, International Society for Music Information Retrieval Conference*, 2018, pp. 34–41.

[8] Miguel A Román, Antonio Pertusa, and Jorge Calvo-zaragoza, "A Holistic Approach to Polyphonic Music Transcription with Neural Networks," in *ISMIR, International Society for Music Information Retrieval Conference*, 2019, pp. 731–737.

[9] Miguel A Román, Antonio Pertusa, and Jorge Calvo-zaragoza, "Data representations for audio-to-score monophonic music transcription," *Expert Systems With Applications*, vol. 162, pp. 113769, 2020.

[10] Han-Wen Nienhuys and Jan Nieuwenhuizen, "Lilypond, a System for Automated Music Engraving," in *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, 2003, pp. 167–171.

[11] Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Dörfler, "A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Proceedings of the AES International Conference*, 2014, pp. 232–239.

[12] Rachel M. Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan P. Bello, "Deep salience representations for F0 estimation in polyphonic music," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, 2017, pp. 63–70.

[13] Judith C. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[14] Meinard Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*, Springer, 2015.

[15] Sebastian Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," *arXiv preprint arXiv:1706.05098*, 2017.

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4, pp. 3104–3112, 2014.

[17] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–15.

[18] Francesco Foscarin, Andrew Mcleod, Philippe Rigaux, Florent Jacquemard, Francesco Foscarin, Andrew Mcleod, Philippe Rigaux, Florent Jacquemard, Masahiko Sakai Asap, Francesco Foscarin, Andrew Mcleod, and Philippe Rigaux, "ASAP : a dataset of aligned scores and performances for piano transcription," in *ISMIR, International Society for Music Information Retrieval Conference*, 2020.

[19] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa/librosa: 0.8.0 (Version 0.8.0)," http://doi.org/10.5281/zenodo.3955228, 2020.

[20] Colin Raffel and Daniel P. W. Ellis, "Intuitive analysis, creation and manipulation of midi data with pretty_midi," in *Proceedings of the 15th International Conference on Music Information Retrieval Late Breaking and Demo Papers*, 2014.

[21] Yoav Goldberg, *Neural Network Methods for Natural Language Processing*, Morgan & Claypool, 2017.

[22] Mert Bay, Andreas F. Ehmann, and J. Stephen Downie, "Evaluation of multiple-F0 estimation and tracking systems," in *ISMIR, International Society for Music Information Retrieval Conference*, 2009, pp. 315–320.

[23] Andrew Mcleod and Mark Steedman, "Evaluating Automatic Polyphonic Music Transcription," in *ISMIR, International Society for Music Information Retrieval Conference*, 2018, pp. 42–49.

[24] Valentin Emiya, Roland Badeau, and Bertrand David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.