DEEP POLYPHONIC ADSR PIANO NOTE TRANSCRIPTION

Rainer Kelz[⋆]*, Sebastian Böck*[⋆]*, Gerhard Widmer*[⋆,†]

[⋆]Austrian Research Institute for Artificial Intelligence, Vienna
[†]Johannes Kepler University, Linz

ABSTRACT

We investigate a late-fusion approach to piano transcription, combined with a strong temporal prior in the form of a handcrafted Hidden Markov Model (HMM). The network architecture under consideration is compact in terms of its number of parameters and easy to train with gradient descent. The network outputs are fused over time in the final stage to obtain note segmentations, with an HMM whose transition probabilities are chosen based on a model of attack, decay, sustain, release (ADSR) envelopes, commonly used for sound synthesis. The note segments are then subject to a final binary decision rule to reject too weak note segment hypotheses. We obtain state-of-the-art results on the MAPS dataset, and are able to outperform other approaches by a large margin, when predicting complete note regions from onsets to offsets.

Index Terms— Convolutional Neural Networks, Polyphonic Transcription, Probabilistic Models

1. INTRODUCTION

Polyphonic transcription is the task of extracting a symbolic score from an audio recording, regardless of how many instruments or notes are playing concurrently. For each note sounding in the recording, we would like to obtain a tuple (s, e, n, v), denoting start, end, MIDI note number and optionally volume. We tackle a somewhat easier subproblem, and attempt to transcribe polyphonic recordings of a single instrument, the piano. We will ignore volume too for now, extracting only (s, e, n) tuples. To this end, we pursue a multi-task deep learning approach with late fusion of the neural networks' predictions in time. Transcription of polyphonic piano music, as well as the deep learning aspect of it, is well studied in the literature [1–7].

In multi-task learning, one attempts to predict multiple targets with a shared representation [8]. This can lead to improved generalization, because representations that are helpful in predicting targets for one task can be utilized to predict targets for other tasks. In our scenario, the tasks are indeed highly related, and there is much potential for representation reuse, as we train a deep convolutional neural network to simultaneously predict the onsets, intermediate note phases and offsets of piano notes. This trick of using the same groundtruth to define multiple targets at different points in time was already mentioned in [8], chapter 8. The network architecture we use is simple, produces musically interpretable features, and has a small number of parameters. Interpreting the network outputs as emission probabilities of an HMM, combined with transition probabilities that directly encode the temporal relationships of different note phases, allows us to obtain plausible note candidates (s, e, n). We can efficiently filter these candidates after decoding to discard a large amount of false positives. This combination of multi-task learning and handcrafted, causal probabilistic temporal model yields state-of-the-art performance on extracting complete notes on the widely used MAPS piano transcription dataset [9].

2. RELATION TO PREVIOUS WORK

It could be shown that modelling different note phases in time with different neural network outputs can be advantageous [2, 4, 5, 8]. The piano transcription approach in [4] uses two separate, bi-directional long-short term recurrent neural networks (BLSTMs) to train a pitched onset detector together with a framewise pitch detector. The onset targets and the intermediate note phase targets are supplied to separate parts of the network, with the onset predictions feeding into the BLSTMs responsible for the final note predictions. This can encourage the suppression of spurious note activities by potentially making them dependent on the presence of an onset. The authors give a few examples demonstrating such a suppression mechanism at work. However, as there are no constraints mentioned to force the desired behavior, the BLSTMs responsible for final note predictions could just as well decide to ignore the onset predictions of the second network.

In a similar vein, [2, 5] use three (or more) separate neural networks for onsets, intermediate note phases and offsets. Predictions are fused either via a handcrafted rule-based method [5], or another neural network on top [2] to obtain symbolic notes.

We borrow this idea of using separate targets for different phases of a note, but drastically simplify the architecture to a much smaller convolutional neural network with a common representation that branches out after a few shared layers, and predicts onsets, intermediate frames, and offsets. This is in contrast to the aforementioned architectures, which neglect any potential for feature reuse, by having completely separate

networks for each task (up to 6 different networks in [2]).

Instead of using BLSTMs or rule-based systems, we pick a different route and post-process the predictions of the network with a handcrafted HMM to obtain individual note segmentations. The states of the HMM roughly correspond to *attack*, *decay*, *sustain* and *release* (ADSR) phases of a note, along with an additional state indicating that no note is currently sounding. ADSR envelopes, as shown in Figure 2a, are commonly used in sound synthesis, governing the volume of a note from onset to offset (ADS) and a brief period afterwards (R). The HMM is not fitted using data, instead we select the transition probabilities of the model manually, and interpret the outputs of the network as emission probabilities. After the decoding step, which yields many candidate note regions, a final decision rule is subsequently applied to each note segment, taking into account the raw network outputs and discarding segments with too small activations within a segment. ADSR envelope inspired mechanisms have been used previously, for example in [10] to model the temporal evolution of spectral envelopes directly.

Post-processing raw, framewise transcriptions with HMMs is a practice widely reported in the literature. The approach in [11] uses Independent Subspace Analysis to extract raw note transcription, and HMMs to model their durations. Similarly, [12] uses two-state HMMs to post-process raw transcriptions obtained using a variety of NMF- and PLCA-based methods. In [13], the temporal evolution of note spectra is modeled via factorial scaled HMMs. The authors in [14] use an HMM variant that explicitly models the duration of staying in a particular state. These are only a small sample of a great variety of related approaches, as can be found in [9, 15–19].

3. MODELS

When predicting multiple targets simultaneously with neural networks, one can consider two ends of a spectrum. One could either branch out immediately after the input layer, and thus have a separate network for each target, or one could branch out immediately before the output layers and have a shared network for all targets. We opt to use a model somewhere in the middle of this spectrum. The first few layers compute a shared representation. Based upon this representation, separate networks branch out, enabling each branch to specialize to the nature of the target it is connected to.

3.1. Deep convolutional neural network

A conceptual drawing of our model architecture is depicted in Figure 1. The network input $\mathbf{x}_t \in \mathbb{R}^{c \times b}$ is a small spectrogram snippet, where c denotes the number of context frames in the time dimension, and b denotes the number of bins in the frequency dimension. The number b is the result of passing a linear magnitude spectrogram through a filterbank with semi-logarithmically spaced, triangular filters. The filterbank has

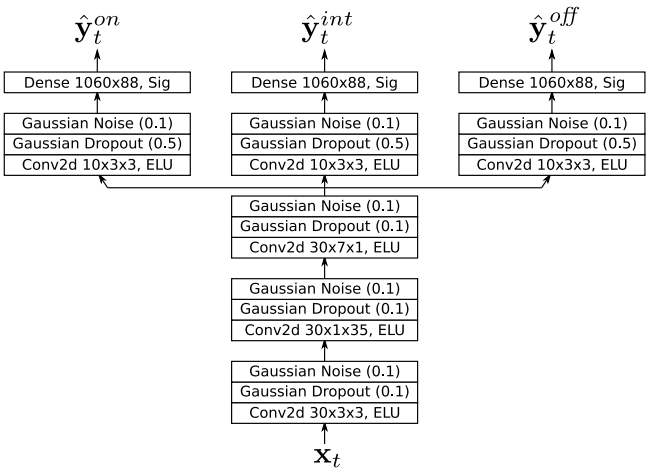

Fig. 1: The architecture of our model. Arrows indicate information flow. The sizes of the convolutional kernels are given as triples $\mathrm{C} \times \mathrm{T} \times \mathrm{F}$, denoting number of channels, elongation in time and elongation in frequency dimension, respectively. The sizes of the dense layers are given as $\mathrm{I} \times \mathrm{O}$, denoting input and output dimensions, and result from concatenating all feature maps of the previous layer into a flat vector of size I.

a linear response and lower resolution for the lower frequencies, and a logarithmic response for the higher frequencies. The resolution of the filtered spectrogram is approximately two bins per semitone. For all our experiments, we selected $c = 11, b = 144$. The temporal resolution of the model is chosen to be $50\,[\mathrm{frames/s}]$. Finally, we compute the logarithm of all magnitude bins, approximately modelling human loudness perception. The target matrix $\mathbf{y}_t \in \{0,1\}^{88\times 3}$ decomposes into vectors $\mathbf{y}_t^{on}, \mathbf{y}_t^{int}$, and $\mathbf{y}_t^{off}$ respectively, denoting the presence of an onset, intermediate note phase, and offset for each note in the center frame within the context window c. Assuming our instrument has K keys, we denote the targets and predictions for the individual keys $k \in \{0..K-1\}$ of the instrument at time t as $\mathbf{y}_t^k \in \{0,1\}^{1\times 3}$. The ground-truth annotation comes in the form of MIDI data, temporally aligned to the accompanying audio recordings. From this annotation, all three different targets are derived, as shown in Figure 2b, where we can also observe that for targets such as onsets and offsets, which have event character, the targets are elongated in time by one frame, to provide a denser learning signal for events.

All nonlinearities are ELUs [20], except for sigmoid functions in the three output layers. The network is composed of small blocks with similar structure: a layer with trainable parameters, such as a convolutional or dense layer, a nonlinearity, followed by a small amount of multiplicative gaussian noise, also called "Gaussian Dropout" [21], which is sampled from $\mathcal{N}(1, (\frac{p_m}{1-p_m})^{1/2})$, followed by a small amount of additive gaussian noise, sampled from $\mathcal{N}(0, p_a)$. Please note that

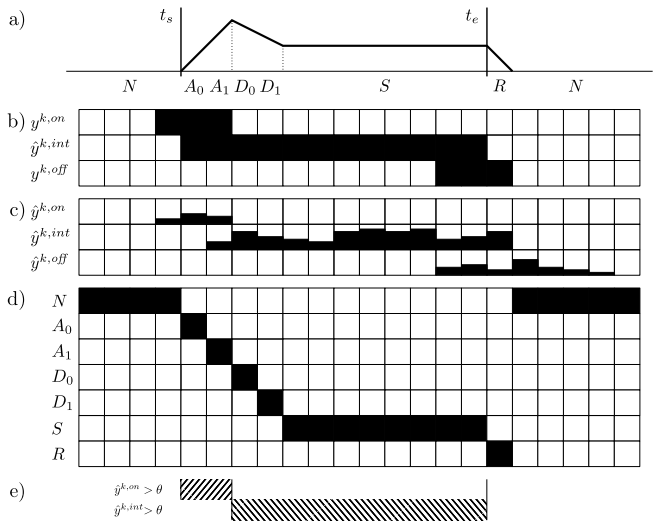

Fig. 2: **a**) An idealized $ADSR$-envelope, describing the different phases of a note in time. Solid vertical lines denote the startpoint t_s and endpoint t_e, extracted from the annotation. **b**) The targets as they are shown to the network during training. Targets with event character, such as onsets and offsets, are elongated in time to three frames. **c**) A sketch of the predictions of the network $\hat{y}^k \in [0, 1]$. **d**) The state space trajectory of the HMM, given the predictions. Note segmentations reach from the start of A_0 to the start of the R state. **e**) The binary decision rule for the two different parts of the note segmentation.

noise is only injected during training [1].

The related approaches in [2, 5] use three or more separate networks to obtain their predictions, which we found to be detrimental. We choose the dimensions for the convolutional kernels based on certain expectations of what features we want the network to emphasize. Kernels elongated in the time direction are supposed to emphasize loudness variations for onsets and offsets, whereas kernels elongated in frequency direction should emphasize overtone structure. The final kernel sizes settled upon can be seen in Figure 1. For an in-depth discussion on musically motivated convolutional kernel shapes see [23]. The network is also compact in terms of the number of parameters, which comes down to $N = 326.394$ for the best performing model.

3.2. Note decoding

We will now describe the HMM-based note decoding stage for individual keys k. After training the network on targets $\mathbf{y}_t$, the predicted pseudo probabilities for each individual instrument key $\hat{\mathbf{y}}_t^k$ are interpreted as emission probabilities of an HMM. The structure of this HMM is depicted in Figure 3. The transition probabilities are determined manually on the

[1]Code to reproduce results is available at `https://github.com/rainerkelz/ICASSP19`, a pretrained model is made available in the *madmom* library [22].

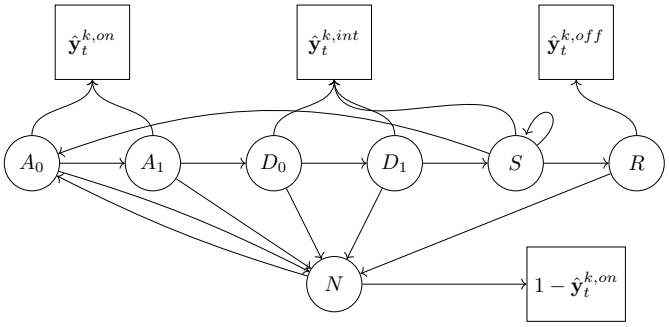

Fig. 3: The $ADSR$-HMM model has seven states: N for no note, $A_{0,1}$ for attack, $D_{0,1}$ for decay, S for sustain, R for release.

training data set, and are shared across all keys. Each key k is decoded into a sequence of note segments individually, however.

For a better understanding of the HMM structure, a sketch of an $ADSR$ envelope is provided in Figure 2a. We will call $\{A_{0,1}, D_{0,1}, S, R\}$ the sounding states, and $\{N\}$ the non-sounding state. The transition probabilities are chosen in such a way that for large pseudo probabilities for onsets, intermediate note phases and offsets, the HMM transitions through all four sounding states, as shown in Figure 2d. Smaller emission probabilities, or even larger gaps in the predictions can lead to transitions that return to the non-sounding state earlier, as we can see from the multiple arrows going into the N state. We allow for the possibility to immediately return to the A_0 state from S, starting a new note segment without going through R. The reason for this is, that pressing the same key rapidly in sequence leads to audio for which the network outputs only very low offset pseudo probabilities. After decoding, we keep all segments that transition at least from A_0 to S for further processing. This effectively establishes a lower bound on the note length, given by the state sequence $\{A_{0,1}, D_{0,1}, S\}$, which yields a minimum note length of $0.1[s]$ at a framerate of $50[\mathrm{frames/s}]$. After note segmentations have been obtained, a final rule is applied to each note segment, utilizing the raw predictions $\hat{\mathbf{y}}_t^k$, with $t \in [\mathrm{frame}(A_0), \mathrm{frame}(\mathrm{last}(S))]$, for that segment, where $\mathrm{frame}(\cdot)$ returns the frame number of an HMM state, and $\mathrm{last}(\cdot)$ returns the last state in a sequence of recurring states. If there is at least one pseudo probability $\hat{y}_t^{k,on} \geq \theta$ during the $\{A_0, A_1\}$ phases, and at least one pseudo probability $\hat{y}_t^{k,int} \geq \theta$ during the $\{D_{0,1}, S\}$ phases, the segment is kept, otherwise it is discarded. An illustration of this mechanism is shown in Figure 2e.

4. EXPERIMENTS

We use the MAPS dataset [24] to train and select models. The dataset contains 210 recordings of classical piano music, rendered using 7 samplebank-based synthesizers. Additionally, there are two sets of recordings of a reproducing Disklavier

	Frames			Note Onsets			Complete Notes		
Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
BLSTM [4]	88.53	70.89	78.30	84.24	80.67	82.29	51.32	49.31	50.22
ADSRNet	90.73	67.85	77.16	90.15	74.78	81.38	61.93	51.66	56.08

Table 1: Experimental results on the Disklavier recordings

piano: 30 recordings from a microphone in close proximity to the piano, and 30 recordings from a microphone farther apart, capturing additional ambient acoustic conditions, such as room reverberations.

All neural network models are trained, compared and selected *only* on the synthetic sources, and finally evaluated on the Disklavier recordings, for which results are reported in Table 1. Additionally, we *remove* any *musical overlap* from the trainset, yielding only 137 musical recordings. We agree with [4] in this regard, and see this as an important step to reduce trainset bias. As the neural network model is fairly small in terms of parameters, and the training loss never reaches zero, we can assume that the model is underfitting the data to some extent. This encouraged us to hand-tune the transition probabilities of the HMM towards best performance *only* on the predictions obtained from the training set, assuming the error behavior and output distribution of the neural network will be similar enough on unseen data, due to the loose fit.

Note transcription performance is determined with the same audio aligned ground-truth annotations and evaluation protocol as in [4], scoring each musical piece individually and averaging performance measures across all pieces. We utilize the `mir_eval` [25] library with the following parametrization: onsets are counted as correctly transcribed, if they are within a $\pm 50[\mathrm{ms}]$ range of the annotated onset. An offset is counted as correct if it is within $\pm 50[\mathrm{ms}]$ or $\pm 20\%$ of the note length, whichever happens to be larger.

We can see from Table 1, that the trained network outputs fairly precise predictions for all three targets, and the majority of keys. As we directly condition the start of note segments on the presence of onset predictions, this necessitates conservatism and confidence in onset and offset predictions. Even though the recall suffers for most of the measurements, due to the small size of the network, and considerable differences in acoustic conditions between train- and testset, the directly enforced constraints on what note segments should look like in time, manage to boost the recognition performance for complete notes considerably.

Results from [5–7] were omitted from Table 1, because their trainsets (called "Configuration II") contain significant musical overlap with the testset, biasing the results.

5. CONCLUSION AND FUTURE WORK

We have shown that simple, small convolutional neural networks with multiple outputs for different temporal phases of a note, together with sequential probabilistic models can achieve state-of-the-art results on a widely used piano transcription dataset.

Some potential improvements for the future include: a global model for typical note lengths, with the help of hierarchical HMMs, trying to infer fingering information from the networks' predictions, which could lead to improvements in transcribing keys which are pressed and released together.

Additionally, we would like to incorporate a *post-hoc*, linear analysis of the volume a note was played at, and only then mapping it to a MIDI velocity number. We believe this to be a better model for volume, than trying to directly predict this quantity with neural networks, as done in [4].

6. ACKNOWLEDGMENTS

This work is supported by the European Research Council via ERC Grant Agreement 670035, project CON ESPRESSIONE and the Austrian Promotion Agency (FFG) under the "BASIS, Basisprogramm" umbrella program. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

7. REFERENCES

[1] Sebastian Böck and Markus Schedl, "Polyphonic Piano Note Transcription with Recurrent Neural Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, Kyoto, Japan, March 25-30*, 2012, pp. 121–124.

[2] Anders Elowsson, "Polyphonic Pitch Tracking with Deep Layered Learning," *CoRR*, vol. abs/1804.02918, 2018.

[3] Carl Thomé and Sven Ahlbäck, "Polyphonic Pitch Detection with Convolutional Recurrent Neural Networks," in *MIREX 2017 abstracts*, 2017.

[4] 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 *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27*, 2018.

[5] Fu'ze Cong, Shuchang Liu, Li Guo, and Geraint A. Wiggins, "A Parallel Fusion Approach to Piano Music Transcription based on Convolutional Neural Network," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, Calgary, AL, Canada, April 15-20*, 2018.

[6] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon, "An End-to-End Neural Network for Polyphonic Piano Music Tran-

scription," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.

[7] Rainer Kelz, Matthias Dorfer, Filip Korzeniowski, Sebastian Böck, Andreas Arzt, and Gerhard Widmer, "On the Potential of Simple Framewise Approaches to Piano Transcription," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 475–481.

[8] Rich Caruana, "A Dozen Tricks with Multitask Learning," in *Neural Networks: Tricks of the Trade - Second Edition*, pp. 163–189. 2012.

[9] Valentin Emiya, Roland Badeau, and Bertrand David, "Automatic Transcription of Piano Music based on HMM Tracking of jointly-estimated Pitches," in *2008 16th European Signal Processing Conference, EUSIPCO 2008, Lausanne, Switzerland, August 25-29*, 2008, pp. 1–5.

[10] Cheng-Te Lee, Yi-Hsuan Yang, and Homer H. Chen, "Multipitch Estimation of Piano Music by Exemplar-Based Sparse Representation," *IEEE Trans. Multimedia*, vol. 14, no. 3-1, pp. 608–618, 2012.

[11] Emmanuel Vincent and Xavier Rodet, "Music Transcription with ISA and HMM," in *Independent Component Analysis and Blind Signal Separation, Fifth International Conference, ICA Granada, Spain, September 22-24, Proceedings*, 2004, pp. 1197–1204.

[12] Dorian Cazau, Yuancheng Wang, Olivier Adam, Qiao Wang, and Grégory Nuel, "Improving Note Segmentation in Automatic Piano Music Transcription Systems with a Two-State Pitch-Wise HMM Method," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 2017, pp. 523–530.

[13] Alexey Ozerov, Cédric Févotte, and Maurice Charbit, "Factorial Scaled Hidden Markov Model for Polyphonic Audio Representation and Source Separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA '09, New Paltz, NY, USA, October 18-21*, 2009, pp. 121–124.

[14] Emmanouil Benetos and Tillman Weyde, "Explicit Duration Hidden Markov Models for Multiple-Instrument Polyphonic Music Transcription," in *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013*, 2013, pp. 269–274.

[15] Emmanouil Benetos and Tillman Weyde, "An Efficient Temporally-Constrained Probabilistic Model for Multiple-Instrument Music Transcription," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, 2015, pp. 701–707.

[16] M. P. Ryynanen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct 2005, pp. 319–322.

[17] Graham E. Poliner and Daniel P. W. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP J. Adv. Sig. Proc.*, vol. 2007.

[18] Tal Ben Yakar, Roee Litman, Pablo Sprechmann, Alexander M. Bronstein, and Guillermo Sapiro, "Bilevel Sparse Models for Polyphonic Music Transcription," in *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR, Curitiba, Brazil, November 4-8*, 2013, pp. 65–70.

[19] Gautham J. Mysore, Paris Smaragdis, and Bhiksha Raj, "Nonnegative Hidden Markov Modeling of Audio with Application to Source Separation," in *Latent Variable Analysis and Signal Separation - 9th International Conference, LVA/ICA 2010, St. Malo, France, September 27-30, 2010. Proceedings*, 2010, pp. 140–148.

[20] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *CoRR*, vol. abs/1511.07289, 2015.

[21] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in *Proceedings of the 24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, 10 2016, pp. 1174–1178.

[23] Jordi Pons, Thomas Lidy, and Xavier Serra, "Experimenting with Musically Motivated Convolutional Neural Networks," in *14th International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2016, pp. 1–6.

[24] Valentin Emiya, Roland Badeau, and Bertrand David, "Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle," *IEEE Trans. Audio, Speech & Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[25] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis, "MIR_EVAL: A Transparent Implementation of Common MIR Metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, 2014, pp. 367–372.