IMPROVING MUSIC TRANSCRIPTION BY PRE-STACKING A U-NET

Fabrizio Pedersoli, George Tzanetakis, Kwang Moo Yi

Department of Computer Science, University of Victoria

ABSTRACT

We propose to pre-stack a U-Net as a way of improving the polyphonic music transcription performance of various baseline Convolutional Neural Networks (CNNS). The U-Net, a network architecture based on skip-connections between layers acts as a *transformation* network followed by a *transcription* network. Notably, we do not introduce any additional loss terms specific to the *transformation* network, but instead, jointly train the entire combined model with the original loss function that was designed for the back-end transcription network. We argue that this U-Net network transforms the input signal into a representation that is more effective for transcription, and thus enables the observed improvements in accuracy. We empirically confirm with several experiments using the MusicNet dataset, that the proposed configuration consistently improves the accuracy of transcription networks. This enhancement cannot be achieved by simply introducing more neurons or more layers to the baseline CNNs. Moreover, we show that using the proposed architecture we can go beyond general music transcription and perform transcription in an instrument-specific fashion. By doing so, the original general transcription performance is also increased.

Index Terms— Automatic Music Transcription, Deep Learning, Deep Architecture, U-Net

1. INTRODUCTION

Automatic Music Transcription (AMT) is an important and open problem in Music Information Retrieval. It consists of detecting which music instrument pitches are present at any particular time by analyzing the acoustic audio signal. AMT has been used in interactive music systems [1] but it remains a challenge especially when there is limited prior knowledge for constraining the algorithms.

Recently AMT, also referred to as polyphonic music transcription, has been treated as a multi-label classification problem with Deep Neural Networks (DNN) [2–4]. Typically, a DNN is trained to predict the active notes within a short analysis block of the audio signal. In most cases, these DNNs are Convolutional Neural Networks (CNNs) and operate on log-spaced (magnitude/power) spectrograms. The choice of using log-spaced spectrograms as input to CNNs for music tasks [4] is motivated by the fact that patterns in the log-spaced frequency domain are shift invariant to pitch changes. A basic approach to enhancing the accuracy of deep networks is to make them deeper [5]. However, as we show experimentally, deeper networks do not always guarantee better performance in AMT.

In this paper, we propose to pre-stack a U-Net in front of existing neural network architectures to enhance polyphonic music transcription performance. The U-Net architecture was initially developed for medical image segmentation [6], because of its ability to reproduce tiny details, and the robustness arising from skip connections. We show empirically that having such an architecture as the first stage of a transcription network with joint training results in improved performance. We believe that the jointly trained U-Net acts as a transformation network, and enhances the input signal in a way that helps the training of the following transcription network.

We show the benefit of the proposed architecture through experiments on the MusicNet dataset [7], focusing both on instrument and instrument-agnostic transcription. For the transcription network, we investigate neural network architectures that are popular for computer vision tasks, as well as one that has been recently suggested and shown to work well for music transcription. We empirically verify that pre-stacking a U-Net in the front always provides enhanced or comparable performance, regardless of the architecture.

2. RELATED WORK

2.1. Traditional methods

Initial approaches to automatic music transcription were mainly unsupervised and based on spectral factorization techniques. In these approaches, the goal is to factorize the magnitude spectra into two components in such a way that one component is related to the frequency profile of each note, and the other one is related to the activation in time of each note. Smaragdis et al. [8] used non-negative matrix factorization (NMF) on the magnitude spectrogram. Although their method had to make some assumptions about the number of unique notes presented in the analyzed audio segment, it showed initial promising results for both monophonic and polyphonic music.

Smaragdis et al. [9] proposed the use of Probabilistic Latent Component Analysis (PLCA) for spectorgram decomposition. This statistical framework models the spectra as a multi-dimensional distribution, which is approximated by a mixture of marginal distribution products. These marginal distributions are estimated using a variant of the Expectation Maximization (EM) algorithm. Smaragdis et al. [10] modified the standard PLCA model to detect multiple local shift invariant patterns. According to this shift invariant model, the marginal distributions are defined in terms of convolutions. Grindaly et al. [11] extended the PLCA model to multiple polyphonic sources. A set of training instruments is used to learn a sparse model space with NMF. This model is then used to learn the distributions of pitches conditioned on the sources. Benetos et al. [12] extends the shift-invariant PCLA to support the use of multiple spectral template per pitch and per instrument. The time varying pitch contribution of each source is also considered by the proposed model extension.

Rather than using spectrogram factorization, Poliner et al. [13] proposed a discriminative model for polyphonic piano transcription in which a Support Vector Machine (SMV) is trained on spectral features, and used to classify frame-level notes instances. In addition, a Hidden Markov Model (HMM) is used to temporally constrain the SVM outputs. Instead of relying on hand-crafted features, Nam et al. [14] use Deep Belief Networks (DBM) to learn feature representations of notes and jointly train classifiers for multiple notes.

Other techniques for AMT rely on multiple-F_0 estimation, in-

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

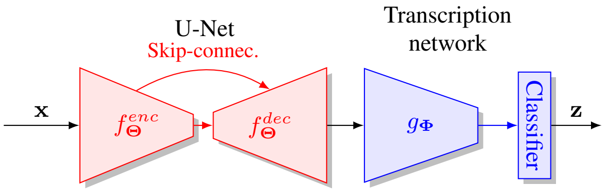

Fig. 1. Proposed instrument-agnostic transcription architecture.

stead of spectrogram factorization. Multiple-F_0 estimation is not reliable enough to provide good transcription results so it is often combined with additional processing stages that model other musical aspects. Ryynanen et al. [15] proposed a music transcription system composed of: multiple-F_0 estimation, an acoustic model, and a musicological model. The acoustic model, takes as input three features extracted from the multiple-F_0 estimates and calculates the likelihoods of different notes and performs temporal segmentation of notes. The musicological model estimates the musical key and controls the transition between notes. The final transcription results are obtained by searching the best paths through the note models. Multiple-F_0 estimation is also used in the work of Benetos et al. [16], where it is combined with note onset/offset detection. The input of the transcription system is the resonator time-frequency image (Zhou et al. [17]). A pitch salience function is extracted for each frame, and onset detection is computed through a spectral flux feature. Finally, a pitch set score function is used for each segment defined by two onsets to estimate the pitch of the current frame.

2.2. Deep learning-based methods

Recently, AMT techniques have been mostly based on deep neural networks, for both the acoustic model and the musical model. The acoustic model consists of neural network operating on a time-frequency representation and trained to predict the active pitches within each frame. For the musical model, the HMM is replaced by a neural network that models temporal dependencies, such as Recurrent Neural Networks (RNNs), or similar architectures. Bittner et al. [2] proposed a fully convolutional neural network for learning salience and estimating fundamental frequencies. The network is trained on a large scale, semi-automatically generated f_0 dataset. In order to better capture harmonic relationships, the authors used a harmonic constant-Q transform as the input representation. Sigtia et al. [3] proposed an architecture comprised of an acoustic model, and a music model for polyphonic piano music transcription. The acoustic model is a neural network that estimates pitch probabilities for a given audio frame. The musical model is an RNN that models temporal dependencies of pitches. The predictions of the two models are combined using a probabilistic graphical model, and the beam search algorithm is used to perform inference.

Thickstun et al. [7] proposed a convolutional architecture for polyphonic music transcription, that extracts features from raw audio rather than using a time-frequency representation as input. A convolutional layer is used as a learned filter-bank that computes a spectrogram-like representation. After a pooling layer, a linear classifier predicts the probabilities of notes active within the considered audio block. This work also introduced the MusicNet dataset that we have used for our experiment. More recently, a deep neural network trained with a loss function that combines onset and frame information has achieved state-of-the-art performance in automatic piano transcription [18]. The U-Net architecture was originally proposed for image segmentation [6]. There has been some limited work using this architecture for music information retrieval tasks such as singing voice separation [19], and audio source separation [20]. To the best of our knowledge, the usage of a U-Net in AMT has not been explored especially for multiple instrument transcription.

3. PRE-STACKING ARCHITECTURES

3.1. Instrument-agnostic transcription

The core idea of our method is a two-stage architecture where we pre-stack a U-Net in front of a typical CNN-based transcription network. We first describe the more general instrument-agnostic case, and then also propose an instrument-specific architecture. If we denote U-Net as $f_{\mathbf{\Theta}}(\cdot)$ and the transcription network as $g_{\mathbf{\Phi}}(\cdot)$, then for a given input $\mathbf{x}$, the proposed architecture computes the label estimates $\mathbf{z}$ using the composition of these two functions (See Fig. 1). Mathematically we write: $\mathbf{z} = g_{\mathbf{\Phi}}\left(f_{\mathbf{\Theta}}(\mathbf{x})\right)$.

Intuitively, the pre-stacked U-Net $f_{\mathbf{\Theta}}(\cdot)$ acts as a *transformation* network that modifies the input signal into a *deep-network-friendly* representation for AMT. Although the architecture can be largely divided into two components, the training of this architecture remains end-to-end. In other words, we train our model with a single cross-entropy loss at the output of the transcription back-end network. Specifically the loss is expressed by the following equation:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_n \log(\hat{\mathbf{y}}_n) + (1-\mathbf{y}_n)\log(1-\hat{\mathbf{y}}_n)\ , \qquad (1)$$

where $\mathbf{y}_n$ are the true labels for the n-the sample in each mini-batch – either one or zero – and $\hat{\mathbf{y}}_n = 1/(1-e^{\mathbf{z}_n})$ – are transformed into a bound signal from $\mathbf{z}$ through a sigmoid. Notice that by doing this we do not constrain the U-Net's output which is treated as a latent variable that is discovered naturally as the training progresses.

It is worth noting that use of a U-Net architecture with skip connections is crucial. Initial empirical investigations revealed severe performance degradation when removing skip connections, which effectively makes the pre-stacked network an Auto Encoder. We omit these results from this paper due to space constraints. Based on this observation, we argue that this implies that a pathway for unhindered input data to go through is essential for the prefix U-Net. We stipulate that the skip-connections act as *anchors* that prevent the output from drifting too far from the original input, without requiring any explicit constraint in the optimization. In short, the prefix U-Net placed in front of a transcription network behaves as a *learned* pre-processing step, that transforms the input in such a way that more meaningful features can be easily extracted by the transcription network, providing overall better classification performance when compared to the sole transcription network. This improvement can be observed for different back-end transcription networks after pre-stacking them with a U-Net (see Section 5).

3.2. Multi-instrument transcription

The MusicNet dataset used in this work is labelled on an instrument basis, that is, each note in the ground truth is annotated with the corresponding instrument. Several tracks are also recordings of ensemble performances in which multiple instruments play together. We leverage this additional information to extend music transcription to instrument-specific transcription. We first identify and separate the instruments, creating a meta-representation for each instrument that no longer has instrument specific characteristics. This results

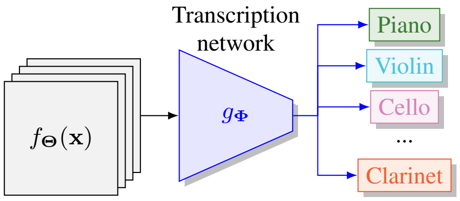

Fig. 2. Instrument-wise transcription architecture. We apply the U-Net front-end to create per-instrument meta-representation, which we then pass on to the *same* transcription network to obtain per-instrument labels (see Section 3.2).

in making the task simpler for the deep networks and increases the variability of data from the point of view of the transcriber.

In more detail, we extend the proposed architecture to perform instrument-wise transcription, as shown in Fig. 2. We apply the front-end U-Net to create multiple outputs, each dedicated for a specific instrument. We then run our transcription (classification) network multiple times, once for each of these outputs to obtain per-instrument transcription (classification) results. Interestingly, our formulation is end-to-end differentiable, and optimal intermediate representations can be found implicitly during training. Therefore, it is not necessary to provide a supervision signal for how the front-end U-Net should transform the data for each instrument.

3.3. Network specifics

Here, we summarize the architectures of the back-end networks considered in our study: 2LR (Eq. (2)) [7], VGG (Eq. (3)) [21], ResNet18 (Eq. (4)) [22], ResNet34 (Eq. (5)) [22]. We use the notation ${}^{\text{\#filters}}C^{\text{kernel}}_{\text{stride}}$, $MP^{\text{kernel}}_{\text{stride}}$, $AP^{\text{kernel}}_{\text{stride}}$, ${}^{\text{\#filters}}R^{\text{kernel}}_{\text{stride}}$, and ${}^{\text{\#units}}L$, to respectively define: convolutional, max pooling, average pool, residual block and, linear layer. The symbol $\circ$ means function and composition, and raise to n-th power means concatenating the considered layer n times.

$$\text{2LR} := {}^{128}C^{32\times1}_{1\times1} \circ {}^{256}C^{1\times32}_{1\times1} \circ {}^{88}L \tag{2}$$

$$\begin{aligned}\text{VGG} := {} & \left({}^{64}C^{3\times3}_{1\times1} \circ MP^{2\times2}_{2\times2}\right)^2 \circ \left({}^{128}C^{3\times3}_{1\times1}\right)^2 \circ MP^{2\times2}_{2\times2} \\ & \circ \left({}^{256}C^{3\times3}_{1\times1}\right)^4 \circ MP^{2\times2}_{2\times2} \circ \left({}^{512}C^{3\times3}_{1\times1}\right)^2 \circ MP^{2\times2}_{2\times2} \\ & \circ {}^{4096}L \circ {}^{4096}L \circ {}^{88}L\end{aligned} \tag{3}$$

$$\begin{aligned}\text{ResNet18} := {} & {}^{64}C^{7\times7}_{1\times1} \circ {}^{64}R^{3\times3}_{1\times1} \circ {}^{64}R^{3\times3}_{2\times2} \circ {}^{128}R^{3\times3}_{1\times1} \\ & \circ {}^{128}R^{3\times3}_{2\times2} \circ {}^{256}R^{3\times3}_{1\times1} \circ {}^{256}R^{3\times3}_{2\times2} \circ {}^{512}R^{3\times3}_{1\times1} \\ & \circ {}^{512}R^{3\times3}_{2\times2} \circ AP^{\vdots}_{\vdots} \circ {}^{88}L\end{aligned} \tag{4}$$

$$\begin{aligned}\text{ResNet34} := {} & {}^{64}C^{7\times7}_{1\times1} \circ {}^{64}R^{3\times3}_{1\times1} \circ {}^{64}R^{3\times3}_{2\times2} \circ \left({}^{128}R^{3\times3}_{1\times1}\right)^3 \\ & \circ {}^{128}R^{3\times3}_{2\times2} \circ \left({}^{256}R^{3\times3}_{1\times1}\right)^5 \circ {}^{256}R^{3\times3}_{2\times2} \circ \left({}^{512}R^{3\times3}_{1\times1}\right)^2 \\ & \circ {}^{512}R^{3\times3}_{2\times2} \circ AP^{\vdots}_{\vdots} \circ {}^{88}L\end{aligned} \tag{5}$$

For U-Net, we apply a standard architecture starting with base 64 channels on the base layer, with a depth of four, where the number of channels is doubled at every depth. Each layer consists of two 3×3 convolutions with stride of one. We further utilize transposed convolutions for upsampling with a stride of two.

4. METHODOLOGY

We experiment with four CNNs for polyphonic music transcription using the MusicNet dataset [7]. We formulate music transcription as a multi-label classification problem, where multiple notes can be active at any particular time. We establish the baseline transcription performances of the "original" CNNs, and then evaluate the performance improvement of pre-stacking with a U-Net.

4.1. Dataset and Input Representation

The MusicNet dataset [7] is a large scale dataset of classical music specifically designed for AMT. The dataset consists of 330 freely licensed recordings (2048 minutes, 1 299 329 labels) of classical music with a variety of instruments arranged in small chamber ensembles under various condition of studio and microphone. The dataset is skewed towards Beethoven (1085 minutes, 736 072 labels) and to solo piano (1346 minutes, 794 532 labels). The MusicNet labels are structured according to the format: starting/ending time, instrument, note, measure, beat, and note value. The labels are retrieved from digital MIDI scores, collected from various archives, and aligned to the recordings using techniques of Turetsky and Ellis [23] with an error rate of 4%. To make our results comparable to Thickstun et al. [7] we use the same test set as theirs: Bach's Prelude in D major for Solo piano, Mozart's Serenade in E-flat major, and Beethoven's String Quartet No.13 in B-flat major.

The dataset is preprocessed by computing the CQT magnitude spectrogram of each recording. CQT spectrograms are computed on 7 octaves with 24 bins per octave with a minimum frequency 32.7 Hz which yields 168 frequency bins in total. In order to be comparable with the previous work of Thickstun et al. [4, 7] we adopt an equivalent setup for input preprocessing, rescaled to our sampling frequency of 11.025 kHz. Specifically, audio and labels are re-sampled to 11.025 kHz using an implementation of the band-limited $sinc$ interpolation method for sampling rate conversion as described by Smith [24]. The CQT spectrogram is computed with an hop length of 128 samples ($\approx 12\,\mathrm{ms}$). Finally, a context window of 32 frames is used for training and testing the neural networks.

4.2. Training

For adapting the networks to the multi-label classification scenario, instead of taking the softmax at the output, we compute the sigmoid, and treat each individual element as a probability value. We use the cross-entropy loss, and optimize the network parameters with the Adam algorithm [25]. For all the experiments, the learning rate is fixed to 1×10^{-4} and the batch size is set to 32. In order to avoid overfitting, the training batch is created by randomly choosing a track, and, randomly choosing a spectrogram frame (and associated context window) within the track using a uniform random distribution. By doing so, we observed no overfitting, thus did not use a validation split to maximize the number of training samples. All the models are trained until convergence, which was determined by observing the performance metric on the training set, and stopping training when no change is observed. We use the micro average-precision (μAP) metric to report classification performance. Each output class is treated independently as a binary prediction. For all the possible threshold values n, we compute precision P_n and recall R_n. This precision-recall curve is summarized as the weighted mean:

$$\mu AP = \sum_n P_n (R_n - R_{n-1}) \tag{6}$$

Table 1. "Instrument-agnostic" transcription results in terms of μAP (%). Our proposed setup (+U-Net) performs best in all cases.

	2LR	VGG	Res18	Res34
Orig.	74.21	74.91	75.65	76.08
+U-Net	75.63	75.05	76.40	76.83

Table 2. "Piano"/"non-piano" transcription results in terms of μAP (%). The proposed architecture with U-Net always outperform the original architecture. Note that best performance is achieved with Res18+U-Net which actually is shallower than Res34.

		2LR	VGG	Res18	Res34
Orig.	Piano	72.96	78.60	79.75	78.20
	¬ Piano	67.21	68.92	70.57	72.83
	Avg.	70.08	73.76	75.16	75.51
+U-Net	Piano	79.01	79.38	80.47	79.83
	¬ Piano	70.03	70.66	72.85	72.28
	Avg.	74.52	75.02	76.66	76.06

Table 3. All instrument transcription results in terms of μAP (%). Best performance is achieved with Res34+U-Net. Note that except Res18, pre-stacking U-Net always help.

		2LR	VGG	Res18	Res34
Orig.	Piano	67.98	74.60	77.70	80.19
	Violin	48.64	44.46	46.80	50.73
	Viola	32.53	31.80	36.35	33.72
	Cello	32.73	38.33	38.56	38.75
	Horn	67.53	64.38	70.38	63.20
	Bassoon	71.61	69.91	73.43	72.11
	Clarinet	64.79	63.98	67.89	68.53
	Avg.	55.12	55.35	58.73	58.18
+U-Net	Piano	78.71	78.42	77.50	79.28
	Violin	49.30	50.10	47.30	52.40
	Viola	32.73	36.80	37.99	36.11
	Cello	37.16	43.37	37.80	40.55
	Horn	68.12	68.45	72.10	72.76
	Bassoon	69.35	72.94	68.09	66.71
	Clarinet	68.09	69.56	67.91	72.63
	Avg.	57.85	59.95	58.38	60.06

A similar methodology for training, testing, and evaluation was utilized in Thickstun et al. [7]. As our focus is frame-level transcription we do not consider note-level metrics.

5. RESULTS

We first report on the "instrument-agnostic" based transcription. In this case the instrument labels are not utilized, that is, the transcribed notes are not associated with a particular instrument. We then analyze the performance of the proposed deep architecture for instrument-wise music transcription. Due to data imbalances of the instrument labels, we first discuss transcription results considering only the "piano" and "non-piano" case, which is roughly balanced.. We then discuss the obtained transcription results using all the instruments classes in the imbalanced data scenario.

5.1. Instrument-agnostic transcription

As shown in Table 1, the proposed architecture based on pre-stacking a U-Net, provides improved accuracy compared to the baseline results of the original architectures. The performance improvement is $\approx 1\%$ absolute, where 2LR, is the most improved network, and VGG the least improved network. Importantly, this improvement is *architecture agnostic* and consistent over all four configurations. Also note that, without our method, Res34 performs worse than Res18 – which hints saturation. Also note that our method applied on Res18 – Res18+U-Net – performs better than Res34, which is shallower. This shows that simply making the network deep, and therefore more complex, is not an as effective solution as our proposed approach.

5.2. Instrument-wise transcription

Table 2 reports transcription results for "piano", and "non-piano" instruments. As in the instrument-agnostic case in Table 1, the proposed architecture based on U-Net improves classification performance for all the considered architectures. The performance improvement is large for the shallow models, $\approx 5\%$ absolute, and it reduces as the model gets deeper, becoming less than 1% absolute for ResNet34. Interestingly, Res18 with our method – Res18+U-Net – performs best. Given that Res34 is already very deep, and near-doubling of the number of layers from Res18 to Res34 provided only small improvement, it is highly unlikely that even a deeper ResNet will be able to achieve the same performance as our method.

Table 3 shows transcription results when all instrument labels are used. Also in this case the proposed architecture based on U-Net achieved an improvement with respect to the baseline models. The improvement is more evident when all the instruments are transcribed, and it ranges from $\approx$4% absolute for VGG network, to $\approx$2% absolute for ResNet34. In case of Res18, comparable performance is achieved with our method to the original, indicating that our method does not significantly harm the performance of the original architecture even in the worst case.

6. CONCLUSION

We have proposed pre-stacking a U-Net architecture before different transcription networks for improving polyphonic transcription at the frame level. Our experiments show that the proposed architecture improves transcription performance of the baseline for both the "instrument-agnostic" and "instrument-specific" scenarios.

We plan to investigate more thoroughly the use of U-Net pre-stacking for other MIR tasks such as audio source separation and singing voice separation, with more experiments, additional data sets, and metrics, including those that relate to statistical significance. Applying our approach to the more complex network of [18] that takes into account both onsets and frames is also an interesting challenge for future work. Finally, it would be interesting to investigate if pre-stacking a U-Net architecture can lead to improvements in tasks beyond MIR such as image classification.

7. REFERENCES

[1] Masataka Goto, "A real-time music-scene-description system: Predominant-f0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[2] Rachel M Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan Pablo Bello, "Deep salience representations for f0 estimation in polyphonic music.," in *ISMIR*, 2017, pp. 63–70.

[3] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 5, pp. 927–939, 2016.

[4] John Thickstun, Zaid Harchaoui, Dean P Foster, and Sham M Kakade, "Invariances and data augmentation for supervised music transcription," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2241–2245.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[7] John Thickstun, Zaid Harchaoui, and Sham Kakade, "Learning features of music from scratch," *arXiv preprint arXiv:1611.09827*, 2016.

[8] Paris Smaragdis and Judith C Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.* IEEE, 2003, pp. 177–180.

[9] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka, "A probabilistic latent variable model for acoustic modeling," *Advances in models for acoustic processing, NIPS*, vol. 148, pp. 8–1, 2006.

[10] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka, "Sparse and shift-invariant feature extraction from non-negative data," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 2069–2072.

[11] Graham C Grindlay and Daniel PW Ellis, "A probabilistic subspace model for multi-instrument polyphonic transcription," 2010.

[12] Emmanouil Benetos and Simon Dixon, "A shift-invariant latent variable model for automatic music transcription," *Computer Music Journal*, vol. 36, no. 4, pp. 81–94, 2012.

[13] Graham E Poliner and Daniel PW Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 048317, 2006.

[14] Juhan Nam, Jiquan Ngiam, Honglak Lee, and Malcolm Slaney, "A classification-based polyphonic piano transcription approach using learned feature representations.," in *ISMIR*, 2011, pp. 175–180.

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

[16] Emmanouil Benetos and Simon Dixon, "Polyphonic music transcription using note onset and offset detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 37–40.

[17] Ruohua Zhou and Marco Mattavelli, "A new time-frequency representation for music signal analysis: Resonator time-frequency image," in *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*. IEEE, 2007, pp. 1–4.

[18] C Hawthorne, E. Elsen, J. Song, A. Roberts, C. Raffel, J. Engel, S. Ooore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *ISMIR*, 2018.

[19] Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde, "Singing voice separation with deep u-net convolutional networks," 2017.

[20] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," in *ISMIR*, 2018.

[21] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] Robert J Turetsky and Daniel PW Ellis, "Ground-truth transcriptions of real music from force-aligned midi syntheses," 2003.

[24] Julius O. Smith, *Digital Audio Resampling Home Page Center for Compute Research in Music and Acoustics*, 2015 (accessed Dec 12, 2018), `http://ccrma.stanford.edu/~jos/resample/`.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.