Mobile-AMT: Real-time Polyphonic Piano Transcription for In-the-Wild Recordings

Yuta Kusaka, Akira Maezawa
Music Informatics Group, R&D Division
Yamaha Corporation
Shizuoka, Japan
{yuta.kusaka, akira.maezawa}@music.yamaha.com

***Abstract*—This paper introduces a polyphonic piano transcription framework that can robustly transcribe piano notes from real-world audio recordings in real time. State-of-the-art (SoTA) automatic music transcription (AMT) methods lack real-time processing capabilities and generalization to unseen recording conditions, making them difficult to deploy in a real-world scenario. To address these challenges, we propose *mobile-AMT*, a new AMT framework that consists of 1) an online and lightweight network architecture with efficient recurrent and convolutional layers, and 2) a data augmentation scheme to enhance robustness against out-of-domain recordings. The mobile-AMT model reduces the computational cost by 82.9% while retaining comparable accuracy to the recent SoTA method, allowing real-time AMT on mobile devices. The proposed augmentation improves the note F1-score by 14.3 points when evaluated on realistic audio with various recording conditions.**

***Index Terms*—piano transcription, deep learning, data augmentation, real-time systems**

I. Introduction

Automatic music transcription (AMT) is a task that predicts a symbolic representation of musical performances from audio recordings [1]. In particular, polyphonic piano transcription has been studied as a challenging task [2]–[5]. AMT is useful for applications such as practice assistance [6] and automatic accompaniment [7]. AMT can potentially expand the scope of existing MIDI-driven music information retrieval systems from digital to acoustic instruments. For AMT to be employed in such interactive usages in the real world, the model must be able to transcribe piano music in real time under various recording conditions.

Deploying existing AMT models for *real-time* and *real-world* applications is challenging. First, real-time applications require an *online* model, which sequentially predicts musical notes from an audio stream with latency in tens of milliseconds. However, most AMT methods [2]–[5] have focused on *offline* models, which employ non-causal models to pursue SoTA accuracy at the sacrifice of real-time processing. In addition, these models exhibit slow inference speed due to their complex architecture, prohibiting real-time inference on resource-limited deployment targets such as mobile phones.

Second, an AMT system for real-world applications should be robust against variations in recording conditions and noises. For example, an AMT model should perform equally well when transcribing a clean piano solo in a quiet concert hall recorded with professional recording equipment or a home piano lesson recorded in a noisy living room with a smartphone. However, AMT models trained on datasets with limited recording conditions, such as the MAESTRO [8] dataset, are prone to result in poor generalization to such unseen recording conditions in the real world.

To meet these requirements, we present a new AMT framework named *mobile-AMT*, which robustly predicts piano notes' onset, offset, and velocity from in-the-wild audio recordings. Our framework is compatible with offline and online AMT tasks; in particular, it can perform real-time online inference even on resource-constrained devices such as mobile phones or tablets. This paper's contributions are:

1) We extend the existing offline model [3] to a lightweight online model by incorporating efficient convolutional layers from computer vision literature [9], [10] and causal recurrent layers. These modifications reduce the computational cost by 82.9% while maintaining comparable accuracy to the original model. Our model demonstrates the real-time factor (RTF) less than 1 on a single CPU core of mobile devices.
2) We develop a data augmentation scheme to improve robustness against out-of-domain recordings. The augmentation process consists of countermeasures against four distribution shifts in a real-world AMT scenario: timbre, room acoustics, background noise, and recording setup. We evaluate our scheme on the realistic dataset [11], which includes various recording conditions.

II. Related Work

Online AMT has received less attention than offline AMT [2]–[5]. Two notable examples of online AMT are the autoregressive multi-state note model (AR model) [12] and Onsets & Velocities (O&V) [13]. The AR model accomplished a low-latency inference of $160\,\mathrm{ms}$, and O&V achieved accuracy close to SoTA offline models. In these models, the output timing precisions are bounded by the hop size of time-frequency analysis; our model predicts precise onset timing based on regression outputs used in [3].

Lightweight network architecture is essential for low-latency online inference. Notes and Multipitch [14] explored a unique and efficient architecture tailored to AMT, enabling on-device processing. In contrast, we take inspiration from

TABLE I
SPECIFICATION FOR THE ACOUSTIC MODEL. THE SYMBOLS c AND t ARE THE OUTPUT CHANNELS AND EXPANSION RATIO OF MBCONVS. SE DENOTES WHETHER A SQUEEZE AND EXCITATION IS PRESENT.

Input	Operator	Kernel	c	Stride	t	SE
(1, 229, T)	Conv2d	(3, 3)	32	(2, 1)	-	-
(32, 115, T)	MBConv	(3, 1)	16	(1, 1)	1	-
(16, 115, T)	MBConv	(3, 3)	32	(2, 1)	6	-
(32, 58, T)	MBConv	(3, 1)	32	(1, 1)	6	-
(32, 58, T)	MBConv	(3, 3)	48	(2, 1)	6	-
(48, 29, T)	MBConv	(3, 1)	48	(1, 1)	6	-
(48, 29, T)	MBConv	(3, 3)	64	(2, 1)	6	✓
(64, 15, T)	MBConv	(3, 1)	64	(1, 1)	6	✓
(64, 15, T)	Flatten	-	960	-	-	-
(960, T)	Linear	-	512	-	-	-
(512, T)	Uni-GRU×2	-	256	-	-	-
(256, T)	Linear	-	88	-	-	-

an approach in computer vision to replace convolutional neural network (CNN) layers with computationally efficient alternatives [9], [10].

The methodology to enhance robustness against out-of-domain recordings has been extensively studied in speech processing. One common approach is data augmentation [15], [16]. For AMT tasks, it is known that data augmentation improves accuracy on different domain datasets [8]. A systematic analysis of various augmentation techniques for AMT has recently been reported [17]. However, a more detailed investigation of the real-world robustness on AMT is needed.

III. PIANO TRANSCRIPTION FRAMEWORK

A. Lightweight Model Architecture

The mobile-AMT model is derived from the regression-based model [3], one of the advanced offline-AMT models. The regression model takes log mel-spectrogram frames $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T) = \boldsymbol{X} \in \mathbb{R}^{F \times T}$, where F and T are the numbers of frequency bins and time frames, as inputs and estimates note-{onset, offset, frame, velocity} frames with CNNs and gated recurrent units (GRU). We construct the mobile-AMT model by modifying the regression model, mainly focusing on the CNNs and GRUs.

In order to enable online inference, we convert all the bi-directional GRUs into uni-directional ones. This modification relaxes the model's non-causality and allow the model to predict output frames at a time frame t using input frames up to $t + \tau$. Here, τ is the size of required future frames. If given r, the size of the CNN's temporal receptive field, the latency introduced by the CNN is computed as follows:

$$L_{\text{CNN}} = N/2 + H\tau = N/2 + H(r-1)/2, \qquad (1)$$

where N and H are the short-time Fourier transform's (STFT) window and hop size in audio samples.

In parallel, we configure the CNNs to accelerate the inference speed. The shorter hop size brings about the lower latency as shown in (1), while making it more challenging to satisfy the RTF < 1. Mobile-AMT achieves low-latency inference under this RTF constraint via its lightweight architecture. Specifically, MBConv [9], [10] replaces the naive CNNs in the model. MBConv is an efficient convolution operator that consists of point-wise and depth-wise convolutions, which can reduce the computational cost while preserving accuracy. Table I shows the MBConv-based acoustic model. Several kernels of MBConv have a size only along the frequency axis, leading to the receptive field size $r = 9$. This kernel design is due to suppressing the latency as calculated in (1).

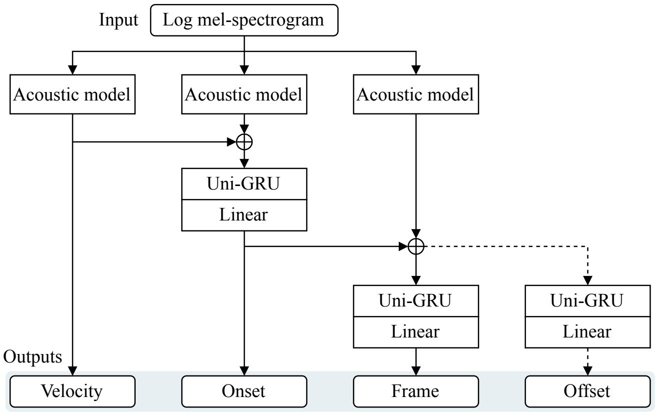

Fig. 1. Overview of the *Mobile-AMT* architecture. Log mel-spectrogram frames are fed into three acoustic models. Their output features are then concatenated and used to infer the note onset, offset, frame, and velocity output. The forward path for offset (dashed line) is skipped during inference.

In summary, Figure 1 illustrates the mobile-AMT architecture. Mobile-AMT omits the offset acoustic model to save the computational cost. The MBConv-based CNN and the entire model reduces the multiply-accumulate operations by $36.0\,\%$ and $82.9\,\%$ compared to the original model, respectively. This efficient architecture design enables mobile-AMT to perform online inference on mobile devices.

After training the model, mobile-AMT performs the online inference in the following steps. First, mel-spectrogram frames are computed frame-by-frame and added to an input buffer of size r. Next, the network predicts the output frames centered at the index of the input buffer. Finally, the output frames undergo the post-processing the same as the regression model in order to obtain precise onset positions [3]. Considering the latencies introduced by the peak detection and time shift in the post-processing, the total latency is computed as follows:

$$L = L_{\text{CNN}} + H + H/2 = N/2 + H(r+2)/2. \qquad (2)$$

B. Data Augmentation Scheme

We categorize distribution shifts in a piano recording process into four major types. Table II summarizes the types of shifts and their examples, taking the MAESTRO dataset [8] and a real-world recording situation as references. The MAESTRO dataset exhibits consistent recording conditions; on the other hand, real-world recordings express various conditions in each category.

The mobile-AMT framework enhances the robustness to such distribution shifts based on data augmentation. The

TABLE II
DISTRIBUTION SHIFTS IN A PIANO RECORDING PROCESS.

	MAESTRO [8]	REAL WORLD
Timbre		
Types of piano	Disklavier	Grand/Upright/Digital piano
Piano tuning	Consistent	Inconsistent
Room acoustics		
Types of room	Concert hall	Living room, classroom, ...
Background noise		
Speech	No	Crowd, instructions, ...
Environment	No	Air conditioner, wind, ...
Recording setup		
Types of mic	Professional mic	Consumer/Mobile-device mic
Clipping	No	Occasionally

TABLE III
DATA AUGMENTATION PIPELINE.

Operation	Scale	Range	Probability
Piano source sampling	-	-	$\boldsymbol{p}_{\text{piano}}$
Pitch shift	cent	[-10, 10]	0.50
Speech noise	SNR (dB)	[0, 20]	0.50
Environment noise	SNR (dB)	[0, 20]	0.50
RIR convolution	-	-	1.00
Stationary noise	SNR (dB)	[15, 25]	0.50
DIR convolution	-	-	1.00
Clipping	ratio	[0, 10]	0.05

proposed augmentation pipeline consists of several countermeasure methods against each shift. The details of each augmentation are as follows:

Timbre. Synthesized piano sounds rendered by software synthesizers augment training data [17]. Given M types of piano sources, a categorical distribution samples a training example with probabilities $\boldsymbol{p}_{\text{piano}} = (p_1, p_2, \ldots, p_M)$. The selected example is then pitch-shifted to simulate various tuning [8].

Room acoustics. We employ the augmentation technique using room impulse responses (RIR) to simulate recordings in different rooms [15]. A pre-recorded RIR signal can apply audio signals to its acoustic characteristics by convolution.

Background noise. Speech and environmental noise are added to audio signals in order to produce noisy recordings. According to [15], we divide noise into point-source and stationary. Point-source noise (i.e., speech and foreground environmental noise) is mixed with signals before the RIR augmentation.

Recording setup. We exploit the augmentation method based on device impulse responses (DIR) [16]. A DIR convolution applies microphones' frequency characteristics to audio signals, increasing the variety of recording devices. Finally, a random percentage of the audio samples are clamped to reproduce hard-clipped recordings.

Table III illustrates the complete augmentation pipeline. An audio example sampled from multiple piano sources passes through the listed augmentations from top to bottom. The pipeline performs each augmentation probabilistically and chooses an augmentation amount uniformly from the range shown in Table III.

IV. EXPERIMENTAL SETUP

A. Datasets and Metrics

We use the MAESTRO dataset and its variant for experiments. The original v3.0.0 dataset [8] includes 200 hours of piano solos with the condition described in Table II. The Studio MAESTRO dataset [17] is a re-recording of the original dataset. It is less noisy and reverberant, which is helpful for applying our augmentation scheme.

The evaluation of the augmentation scheme employs the IDMT-PIANO-MM dataset [11], which contains 432 piano solos recorded in various conditions. In the dataset, nine musical pieces were played in eight different rooms with room volume range of $40 - 4400\,\mathrm{m}^3$, each on upright, digital, and grand pianos. All performances were recorded with one stereo microphone, two smartphones, and three tablets.

Our augmentation scheme adopts the following datasets. Modartt Pianoteq, using presets "Steinway Model D (New York, Hamburg)" and "YC5" without reverbs or EQs, renders synthetic piano sounds from the MIDI files contained in the MAESTRO dataset. Speech and environmental noise are taken from the train-clean-100 subset in LibriSpeech [18] and Room Impulse Response and Noise Database [15]. RIRs are drawn from the IR Survey dataset [19], which includes 271 RIRs recorded in everyday life places. DIRs come from micIRP[1] and Tiny-IRs[2], which contain DIRs from 66 vintage microphones and five daily drivers such as a smartphone, respectively.

Transcription accuracy is evaluated using note-level metrics. The *note*, *note with offset*, and *note with offset + velocity* scores are computed by `mir_eval` [20] with the default parameters. For the IDMT dataset, only the note score is obtained because the offset labels included in this dataset is not precise.

B. Model and Training Configurations

The input log mel-spectrogram is computed as follows. All audio signals are converted to mono and resampled to $16\,\mathrm{kHz}$, and then a random 10-second segment is extracted from the signals. The log mel-spectrogram is computed with a 229-dimensional filter bank, using a STFT computed using a Hann window of $N = 2048$, $H = 320$, after passing the signal through a high-pass filter with a cutoff frequency of $30\,\mathrm{Hz}$. These parameters lead to the hop duration of $20\,\mathrm{ms}$ and the latency of $174\,\mathrm{ms}$. The sharpness parameter for onsets and offsets [3] is set to 3.

Table I summarizes the hyperparameters of the acoustic model. The reduction ratio for the squeeze and excitation is set to 8. Batch normalization and ReLU activation are followed by each layer in the model, except for the output layers. The output layers comprise a uni-GRU with a hidden size 256, a single linear layer, and a sigmoid activation. The loss function is the same as the original model, except that the velocity

[1] https://micirp.blogspot.com/

[2] https://collectedtransients.com/product/tiny-irs-impulse-responses/

TABLE IV
TRANSCRIPTION RESULTS EVALUATED ON THE MAESTRO DATASET.

	Online	Latency	Params.	NOTE			NOTE W/ OFFSET			NOTE W/ OFFSET & VEL.		
				P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
Mobile-AMT	✓	174 ms	5.9M	98.26	94.46	96.30	78.35	75.35	76.80	77.04	74.12	75.53
Regression [3]	-	-	20.2M	98.17	95.35	96.72	83.68	81.32	82.47	82.10	79.80	80.92
Onsets & Frames [8]	-	-	18.3M	98.27	92.61	95.32	82.95	78.24	80.50	79.89	75.37	77.54
AR model (five small) [12]	✓	160 ms	6.1M	98.89	90.43	94.38	77.55	70.93	74.03	-	-	-
Onsets & Velocities [13]	✓	4 – 9 s	3.1M	98.58	95.07	96.78	-	-	-	-	-	-

loss is replaced with mean squared error. The post-processing employs the onset threshold $\theta_{\mathrm{on}} = 0.3$ unless otherwise stated.

The model is trained with a batch size of 16, using the Adam optimizer with a learning rate of 0.001 annealed to 0 with cosine scheduler [21]. The number of epochs required for training depends on the dataset and the augmentations used. For reference, it took $3\,\mathrm{k}$ epochs to train the model with no augmentation (1.1 days on a NVIDIA A100 80GB GPU).

The `SoX` and `audiomentations`[3] library implements the pitch shift and the other augmentations, respectively. The piano-source sampling probabilities $\boldsymbol{p}_{\mathrm{piano}}$ are uniform in the set of MAESTRO types (original, studio, and Pianoteq) and also in Pianoteq sources. Speech and foreground environmental noise are mixed as a burst sound with $1-4\,\mathrm{s}$ pauses between. Background environmental noise and stationary noise are mixed to cover an entire signal. All impulse responses are resampled to $16\,\mathrm{kHz}$, and their leading silence is manually trimmed to eliminate delays introduced by convolution.

C. Evaluation Scheme

To evaluate mobile-AMT, we conducted four experiments. First, mobile-AMT was compared with the offline [2], [3] and online [12], [13] baseline models in terms of transcription accuracy, using the original MAESTRO dataset. The mobile-AMT model was trained for $3\,\mathrm{k}$ epochs without augmentation. Note that the version of MAESTRO differs according to the models.

Next, we assessed the on-device inference performances of the proposed lightweight architecture. Mobile-AMT was implemented with ONNX Runtime [22], a cross-platform machine learning accelerator, and deployed on a 2019 MacBook Pro (2.4 GHz 8core Intel Core i9 64 GB 2667 MHz DDR4), 2019 iPad Pro, and Pixel6. Each device performed online inference using a single CPU core. The inference performance was measured in RTF and memory consumption.

Then, we evaluated the proposed augmentation scheme. The mobile-AMT model was trained on the {original, studio, mix} MAESTRO dataset, where the "mix" represents a mixture of the other two datasets, with the full augmentation. The number of epochs depended on the scale of the training set. The models were tested on the original MAESTRO and IDMT datasets. Only for the Studio MAESTRO model, θ_{on} was set to 0.45.

TABLE V
EVALUATION RESULTS WITH THE DATA AUGMENTATION. EACH VALUE DENOTES THE MEAN AND STANDARD DEVIATION OF NOTE F1-SCORE.

Training Scheme	Epochs	NOTE F1 (%)	
		MAESTRO	IDMT
Original MAESTRO	3 k	96.30 (2.85)	78.84 (14.23)
Original MAESTRO + Aug.	10 k	95.93 (3.02)	92.94 (5.00)
Studio MAESTRO + Aug.	10 k	93.35 (3.85)	92.20 (5.26)
Mix MAESTRO + Aug.	13 k	95.93 (3.00)	93.15 (4.97)

Finally, we conducted an ablation study of the augmentation pipeline to determine the individual contribution of each augmentation toward the robustness improvements. The mobile-AMT models were trained on Studio MAESTRO, skipping a given augmentation each time, and then tested on the IDMT dataset. Supplementarily, we compared the fully-augmented and speech-ablated models to investigate the impact of the speech augmentation. Both models were tested on the IDMT dataset accompanied by the LibriSpeech test-clean set [18]. The speech noises from LibriSpeech were added to the test set in a similar manner as the speech augmentation, with a fixed signal-to-noise ratio (SNR) of $10\,\mathrm{dB}$.

V. RESULTS

A. Comparison with Baseline Models

Table IV summarizes the results from mobile-AMT and the baseline models. The first two rows highlight the successful conversion of the regression model into an online model accompanied by acceptable degradations, especially with a drop in the note F1-score by 0.42 points. Among the online models, mobile-AMT outperformed the AR model in the note and note with offset F1-scores and showed comparable accuracy with O&V. This indicates a favorable balance of accuracy and latency for our model.

B. On-device Inference Performance

The RTFs of mobile-AMT, measured on the MacBook, iPad, and Pixel6, were 0.25, 0.35, and 0.6, respectively. The peak memory observed on the MacBook was $129\,\mathrm{MB}$. Thanks to the frame-by-frame online inference, mobile-AMT consumes considerably less memory than the offline model. These measurements suggest the real-time processing capability of our lightweight architecture on mobile devices.

[3] https://github.com/iver56/audiomentations

TABLE VI
ABLATION STUDY OF THE DATA AUGMENTATION PIPELINE. EACH VALUE DENOTES THE STATISTICS OF NOTE F1-SCORE (%).

Training Scheme	**Mean (std)**	**Quartile**		
		25%	50%	75%
Studio MAESTRO + Aug.	92.20 (5.26)	88.95	93.33	96.23
w/o piano source sampling	86.43 (7.83)	81.42	87.84	92.17
w/o pitch shift	90.71 (6.12)	87.25	91.55	95.22
w/o speech noise	92.37 (5.31)	89.21	93.40	96.45
w/o environmental noise	90.94 (5.82)	87.78	91.78	95.17
w/o RIR convolution	90.20 (5.88)	86.67	91.07	94.71
w/o DIR convolution	92.04 (5.57)	88.89	92.92	96.31
w/o clipping	91.98 (5.56)	88.80	93.09	96.20

C. Effectiveness of the Data Augmentation

Table V presents the transcription results using the proposed training scheme. The first row shows the severe performance degradation on the IDMT dataset, suggesting that a model trained on a dataset with limited conditions exhibits poor performance under real-world conditions. On the other hand, the augmented models achieved better scores than the non-augmented model; in particular, the mix MAESTRO plus augmentation model improved the note F1-score by 14.3 points compared to the original MAESTRO model. The reduced standard deviations imply that the models became more stable in predicting piano notes in unseen conditions.

D. Ablation Study of Data Augmentation

Table VI presents the result of the ablation study. All the augmentations, except the speech augmentation, improved the note F1-score on the IDMT dataset. Meanwhile, the model without speech augmentation experienced a slight decrease in accuracy. We presume this is because speech noise is absent from the IDMT dataset. In the supplementary experiment using the noisy IDMT dataset, the fully-augmented and speech-ablated models achieved note F1-scores of 89.73% and 79.47%, respectively. Considering the note F1-score on the clean IDMT dataset (92.20%), the speech augmentation mitigates performance decline in noisy conditions. Section V-C and V-D conclude that all of the augmentations in the pipeline contribute to enhancing real-world robustness.

VI. CONCLUSION

In this study, we proposed *mobile-AMT*, a robust real-time polyphonic piano transcription framework. Our lightweight architecture can transcribe piano notes in real time, even on mobile devices. The proposed training scheme using several data augmentation techniques enhanced the AMT model's robustness against out-of-domain recordings. The experiments have shown that our model is accurate, lightweight, and robust for practical real-time applications. We plan to explore more efficient architecture and data augmentation pipelines.

REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic Music Transcription: An Overview," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 20–30, Jan. 2019.

[2] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and Frames: Dual-Objective Piano Transcription," in *Proc. 19nd Int. Society for Music Information Retrieval Conf.*, Sep. 2018, pp. 50–57.

[3] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution Piano Transcription with Pedals by Regressing Onset and Offset Times," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, pp. 3707–3717, Oct. 2021.

[4] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, "Sequence-to-Sequence Piano Transcription with Transformers," in *Proc. 22nd Int. Society for Music Information Retrieval Conf.*, Nov. 2021, pp. 246–253.

[5] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. Liao, and Y. Mitsufuji, "Automatic Piano Transcription with Hierarchical Frequency-Time Transformer," in *Proc. 24th Int. Society for Music Information Retrieval Conf.*, Nov. 2023, pp. 215–222.

[6] R. Guo, J. Cui, W. Zhao, and S. Li, "Ai and ar based interface for piano training," in *2020 Int. Conf. on Virtual Reality and Visualization (ICVRV)*, Nov. 2020, pp. 328–330.

[7] S. Sagayama, T. Nakamura, E. Nakamura, Y. Saito, H. Kameoka, and N. Ono, "Automatic music accompaniment allowing errors and arbitrary repeats and jumps," *Proc. Mtgs. Acoust.*, vol. 21, p. 035003, Dec. 2014.

[8] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," in *7th Int. Conf. on Learning Representations*, May 2019.

[9] A. Howard, R. Pang, H. Adam, Q. V. Le, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, and Y. Zhu, "Searching for MobileNetV3," in *IEEE/CVF Int. Conf. on Computer Vision*, Oct. 2019, pp. 1314–1324.

[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th Int. Conf. on Machine Learning*, Jul. 2019, pp. 6105–6114.

[11] J. Abeßer, F. Bittner, M. Richter, M. Gonzalez Rodriguez, and H. Lukashevich, "A Benchmark Dataset to Study Microphone Mismatch Conditions for Piano Multipitch Estimation on Mobile Devices," in *Proc. Digital Music Research Network One-day Workshop (DMRN+16)*, Dec. 2021.

[12] T. Kwon, D. Jeong, and J. Nam, "Polyphonic Piano Transcription Using Autoregressive Multi-State Note Model," in *Proc. 21th Int. Society for Music Information Retrieval Conf.*, Jul. 2020, pp. 454–461.

[13] A. Fernandez, "Onsets and Velocities: Affordable Real-Time Piano Transcription Using Convolutional Neural Networks," in *31st European Signal Processing Conf.*, Sep. 2023, pp. 151–155.

[14] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation," in *2022 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, May 2022, pp. 781–785.

[15] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A Study on Data Augmentation of Reverberant Speech for Robust Speech Recognition," in *2017 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Mar. 2017, pp. 5220–5224.

[16] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-Robust Acoustic Scene Classification via Impulse Response Augmentation," in *31st European Signal Processing Conf.*, Sep. 2023, pp. 176–180.

[17] D. Edwards, S. Dixon, E. Benetos, A. Maezawa, and Y. Kusaka, "A Data-Driven Analysis of Robust Automatic Piano Transcription," *IEEE Signal Process. Lett.*, vol. 31, pp. 681–685, 2024.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Apr. 2015, pp. 5206–5210.

[19] J. Traer and J. H. McDermott, "Statistics of natural reverberation enable perceptual separation of sound and space," *Proc. Natl. Acad. Sci. USA*, vol. 113, no. 48, pp. E7856–E7865, Nov. 2016.

[20] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A Transparent Implementation of Common MIR Metrics," in *Proc. 15th Int. Society for Music Information Retrieval Conf.*, Oct. 2014, pp. 367–372.

[21] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," in *5th Int. Conf. on Learning Representations*, Apr. 2017.

[22] (2021) ONNX Runtime. [Online]. Available: https://onnxruntime.ai/