The Impact of Audio Input Representations on Neural Network based Music Transcription

Kin Wai Cheuk
Information Systems,
Technology, and Design
Singapore University
of Technology and Design

Institute of
High Performance Computing
Agency for Science,
Technology and Research
Singapore
kinwai_cheuk@mymail.sutd.edu.sg

Kat Agres
Yong Siew Toh Conservatory of Music
National University of Singapore
Institute of
High Performance Computing
Agency for Science,
Technology and Research
Singapore
muskra@nus.edu.sg

Dorien Herremans
Information Systems,
Technology, and Design
Singapore University
of Technology and Design

Institute of
High Performance Computing
Agency for Science,
Technology and Research
Singapore
dorien_herremans@sutd.edu.sg

***Abstract*—This paper thoroughly analyses the effect of different input representations on polyphonic multi-instrument music transcription. We use our own GPU based spectrogram extraction tool, nnAudio, to investigate the influence of using a linear-frequency spectrogram, log-frequency spectrogram, Mel spectrogram, and constant-Q transform (CQT). Our results show that a 8.33% increase in transcription accuracy and a 9.39% reduction in error can be obtained by choosing the appropriate input representation (log-frequency spectrogram with STFT window length 4,096 and 2,048 frequency bins in the spectrogram) without changing the neural network design (single layer fully connected). Our experiments also show that Mel spectrogram is a compact representation for which we can reduce the number of frequency bins to only 512 while still keeping a relatively high music transcription accuracy.**

***Index Terms*—Automatic Music Transcription, Spectrogram, Neural Network, Audio input representation**

I. Introduction

Polyphonic music transcription is extremely difficult, yet it is a fundamental step to other music information retrieval tasks [1, 2, 3, 4]. Most automatic music transcription (AMT) research focuses on developing sophisticated models for the transcription problem. Various models such as support vector machines (SVM) [5, 6], restricted Boltzmann machines (RBM) [4], long-short term memory neural networks [7], and convolutional neural networks (CNN) [8, 9] have been developed to tackle this task. For example, Wang et al. [10] integrate non-negative matrix factorization (NMF) with a CNN in order to improve transcription accuracy. Hawthorne et al. [11] split the AMT into three sub-tasks: onset detection, frame activation, and velocity estimation, which allows them to achieve state-of-the art transcription accuracy on piano music. Most literature focuses mainly on polyphonic piano transcription and their models are being trained on piano datasets such as MAPS [12]. The existing studies on music transcription often use different input representations such as log magnitude spectrogram [5, 6], constant-q transform (CQT) [7, 10], and Mel spectrogram [11]. Only few studies, however, offer a comparison of the effect of different input representations [8, 13, 14, 15]. One of the most comprehensive studies [14] compares the effect of using spectrograms with linearly and logarithmically spaced frequency bins versus CQT. Their results show that a single layer, fully connected network that uses logarithmic frequency bins with a logarithmic magnitude spectrogram performs best on polyphonic piano transcription with the MAPS dataset. There are, however, several missing aspects in their study. First, a comparison with one of the popular representations, Mel spectrogram, is missing. Second, the effect of the number of bins for different representations was not examined. Finally, they did not test whether the same approach may be applied for polyphonic multi-instrument transcription.

Our study provides a more comprehensive analysis of the effect that input representation has on polyphonic audio transcription accuracy for multiple instruments when using a single layer fully connected network. We make use of our own GPU based spectrogram toolkit in Python (nnAudio) [16, 17] to allow for on-the-fly spectrogram calculations. The main contributions of this research are to: 1) provide a comprehensive analysis of the effect of four popular frequency domain input representations on polyphonic music transcription; 2) provide an analysis of model performance when varying the number of bins for each of the input representations; and 3) study the model's performance when varying the input frame length. We first discuss our base model for transcription and input representations before moving on to the experimental setup and results. Finally, a conclusion wraps up the paper together with suggestions for future work. The source code is available at our github page[1].

[1] https://github.com/KinWaiCheuk/IJCNN2020_music_transcription.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

II. System description

A. Input Representations

Our experiments compare the influence of four different input representations: linear-frequency spectrogram (LinSpec), log-frequency spectrogram (LogSpec), CQT, and Mel spectrogram (MelSpec). In order to study the effect of different spectrogram parameters, we developed a GPU based spectrogram conversion toolkit using Pytorch, called nnAudio[2]. This allows us to generate different types of spectrograms on-the-fly while training the network. Fig. 3 shows the four input representations studied in this paper. The sampling rate of the audio clips follows the work of Thickstun et al. [8, 18], which are kept at $44,100$Hz, and the hop size for waveforms to spectrograms transformation is kept constant at 512 throughout our experiments.

For the sake of convenience, we will briefly mention the mathematics behind each spectrogram representation and the differences among them to save readers troubles from looking them up individually.

1) LinSpec: This input type is the typical spectrogram obtained with short-time Fourier transform (STFT) as shown in equation 1, where the k inside $X[k,\tau]$ is the frequency bins of the spectrogram and the τ inside $X[k,\tau]$ is the time-step of the spectrogram. N is the window size, n is the sample index for the input waveform, and h is the hop size. Each frequency bin k corresponds to a physical frequency f in the unit of Hertz through the relationship of $f = k\frac{s}{N}$, where s is the sampling rate. In this case, the spacing between each k is linear, and hence this spectrogram has a linear frequency along the y-axis.

$$X[k,\tau] = \sum_{n=0}^{N-1} x[n+h\tau]\cdot e^{-2\pi i k \frac{n}{N}}. \tag{1}$$

2) LogSpec: This spectrogram shares the same equation 1 as LinSpec, except the spacing between frequency bin k is logarithmic instead of linear. Since the musical notes have a logarithmic relationship with each other, one would expect that LogSpec works well for music.

3) MelSpec: Mel spectrograms use a Mel scale which is an attempt to imitate humans' perceptual hearing on musical pitches [19, 20, 21]. There are various versions of Mel scale, in this paper, we use the version implemented in the HTK Speech Recognition toolkit [22]. Then, we can use Mel filter banks to map frequency bins k from STFT to Mel bins in Mel spectrograms as shown in Fig. 2.

4) CQT: Constant-Q transformation (CQT) [23, 24, 25] is a modification of the STFT with the formula as shown in equation 2.

$$X^{cq}[k_{cq},\tau] = \sum_{n=0}^{N_{k_{cq}}-1} x[n+h\tau]\cdot e^{-2\pi i Q \frac{n}{N_{k_{cq}}}} \tag{2}$$

Different from STFT, CQT has a constant factor Q instead of the variable k in $e^{-2\pi i Q \frac{n}{N_{k_{cq}}}}$. Also, the window length $N_{k_{cq}}$ for CQT is now a variable instead of a constant. The constant factor Q is defined as $Q = (2^{\frac{1}{b}} - 1)^{-1}$ and the variable window length is defined as $N_{k_{cq}} = \text{ceil}\left(\frac{s}{f_{k_{cq}}}\right) Q$. Here b is the number of bins per octave, which is a parameter for CQT.

B. Model Architecture

We use the architecture proposed by Thickstun et al. [8] as our base model (see Fig. 1), with which we test the influence of different spectrogram representations. This architecture consists of only one linearly fully connected layer with 128 neurons, whereby the spectrograms are flattened before feeding them as input to the model. The neural network architecture is kept minimal in order to focus the effect of input representations on music transcription accuracy. The network is trained for 35 epochs, in batches of size 1,000. Each batch contains 100 spectrograms sampled from the dataset. Adam optimizer with learning rate 10^{-6} is used due to a faster convergence. The output $\hat{y}$ of the neural network is used to calculate the binary cross entropy with the pitch label obtained at the time step that corresponds to the the center of the spectrogram (red dotted line in Fig. 1).

III. Experimental Setup

A. Dataset and evaluation

We perform a number of experiments using the MusicNet [18] dataset. The dataset contains 330 classical music recordings with multi-pitch annotations aligned using dynamic time warping, and verified by trained musicians. Unlike previous datasets, which focus mostly on piano music [12], MusicNet contains various musical instruments such as violin, clarinet, flute, and even harpsichord. This makes the transcription task much harder and realistic. The training set consists of 320 recordings, and the remaining ten recordings are used for evaluation. We use the same split for training and evaluation as Thickstun et al. [8], and use precision, accuracy, and error as defined by **mir_eval.multipitch.metrics** [26], to ensure a fair comparison.

B. Experiment 1: Resolution of Spectrograms

In this experiment, we study the effect of *varying the number of spectrogram bins* from roughly 256 to 2,048 for each of the input representations. The bin spacings have been carefully chosen such that the first bin starts at 50Hz and the last bin ends at $6,000$Hz for all of the input representations. For LinSpec and LogSpec, the short-time Fourier transform (STFT) window size is fixed to 4,096, and the number of frequency bins is varied from 256 to 2,048. A smaller number of frequency bins means a larger frequency spacing between each bin. In other words, the spectrogram resolution is changed by the frequency bins.

For CQT, the first bin corresponds to the musical note A1 and the last bin corresponds to F♯8 (encompassing 84 notes

[2] https://github.com/KinWaiCheuk/nnAudio

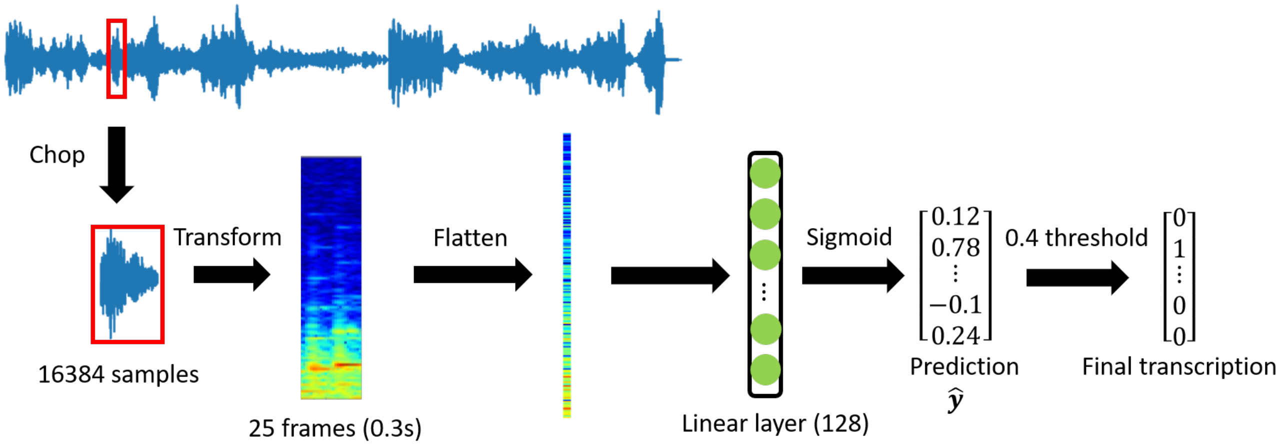

Fig. 1. The MusicNet model from Thickstun et al. [8] is used as our base architecture.

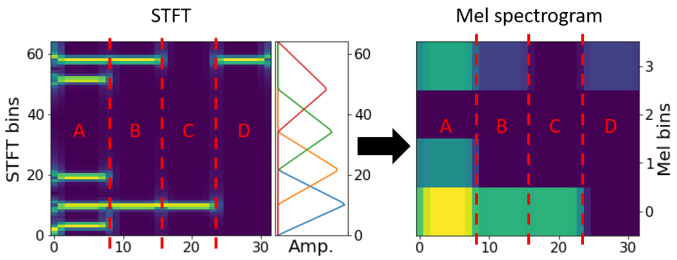

Fig. 2. Mel filter banks can be used to convert a STFT output to Mel spectrogram. Each Mel filter bank covers multiple STFT bins k and reduces them into one Mel bin.

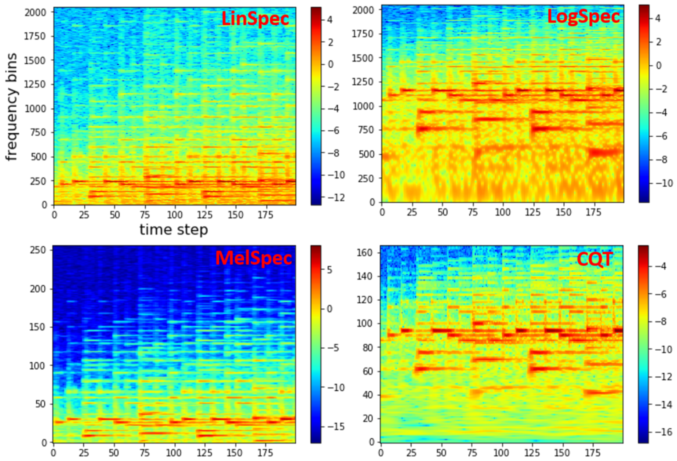

Fig. 3. Excerpt from Bach's WTK I, No. 5, Prelude (ID:2303) visualized by four input representations: linear-frequency spectrogram, log-frequency spectrogram, Mel spectrogram, and CQT (clockwise from top left).

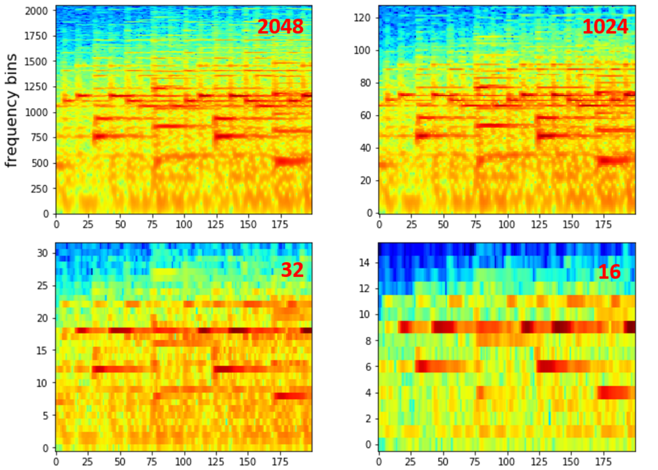

Fig. 4. LogSpec of song ID:2303 with varying bin size of (2,048, 1,024, 32, and 16).

from 55Hz to 5919Hz). During CQT calculation, the number of bins of the spectrogram can be tuned by adjusting the number of bins per octave. If the number of bins per octave is 12, then the total number of bins is 84; if 24 bins per octave are used, then the total number of bins is 168; and so on. Fig. 4 shows the influence of changing the bin size on LogSpec. It can be seen that the rightmost one has a lower resolution. Throughout the experiment, a fixed hop size of 512 is used. For the MelSpec, a fixed STFT window size of 4,096 is used, and the number of Mel filter banks is varied from 128 to 2,048.

C. *Experiment 2: STFT window size*

In the second experiment, we analyse the effect of the *STFT window size*, together with varying bin size on the transcription accuracy for the MelSpec, LogSpec, and LinSpec based models. The STFT window size directly affects the frequency resolution of the spectrogram. A longer STFT windows size ensures a high *frequency* resolution by sacrificing the *time* resolution [27]. Because the STFT window size is kept constant in Experiment 1 (4,096), the time resolution in the previous experiment is fixed. In this experiment, however, the time resolution varies so that we can examine the effect of a high time resolution on transcription accuracy. For LinSpec and LogSpec, the number of bins is set to half of the window size of the short-time Fourier transform (STFT), so as to remove redundant information due to symmetry. For instance, when a STFT window size of 4,096 is used, 2,048 frequency bins are used to create the spectrograms, and so on. For MelSpec, a different number of Mel filter banks are explored for each varying value of STFT window size. Because the STFT window size for CQT cannot be adjusted independently (see Section IV-B), it is not studied in this experiment.

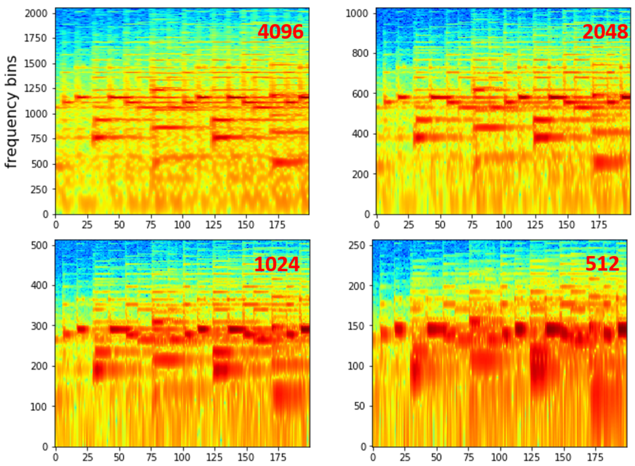

Fig. 5. LogSpec of song ID:2303 with varying STFT window sizes of (4,096, 2,048, 1,024, and 512)

D. Experiment 3: Length of input frame size

In the previous two experiments, the input audio length is fixed at 16,384 samples. In this experiment, however, we study the effect of varying the input audio length from 0.14 to 0.45 seconds. All the other spectrogram parameters such as STFT window size (4,096), Mel filter banks (512), and number of bins per octave (72) are held constant.

IV. Results and Discussion

Fig. 6 shows the results for the three experiments, which are discussed in the following subsections.

A. Experiment 1

Intuitively, one might think that a higher spectrogram resolution would result in higher transcription accuracy. Fig. 6(a), however, shows us that this is not necessarily true for all input representations, such as CQT. In the case of CQT, when the number of bins per octave is 12, the transcription accuracy is as low as 0.16. As the number of bins per octave is doubled to 24, the transcription accuracy increases dramatically to 0.36, and keeps improving slightly until the number of bins reaches 84. After that, the transcription accuracy for CQT deteriorates. When comparing CQT with other input representations, such as LogSpec or MelSpec, CQT-based models do not perform as well. This is consistent with the existing literature [14, 28]. LinSpec and LogSpec have a comparable transcription accuracy, except when the number of bins is less than 768. In that case, LogSpec performs slightly better than the LinSpec. Since the bin spacings for LogSpec are similar to relationships between musical notes, LogSpec can still maintain useful information when the number of bins is low. When there are enough bins, the difference in transcription accuracy between LogSpec and LinSpec is less obvious because the large number of bins evens out the advantage that LogSpec has for low frequency signals. As for models based on MelSpec, these perform best when the number of Mel filter banks is 1,024. If more Mel filter banks are used, some of these will be empty. This is why MelSpec with too many Mel bins performs slightly worse than both log-frequency and linear-frequency spectrograms. This result also reveals that Mel filter banks can be considered an effective compression algorithm for spectrograms. While reducing the number of frequency bins they still maintain a relatively high transcription accuracy. Overall, the best performing input representation is LogSpec with an STFT window size of 4,096 and 2,048 bins.

B. Experiment 2

Fig. 6(b) shows the influence of varying the STFT window size (or so-called n_fft in some of the audio processing libraries) for the MelSpec representation. The transcription accuracy for a MelSpec with a specific window size is always lower than its log-frequency spectrogram counterpart with the same window size. The results further suggest that the Mel filter banks act as a spectrogram compression to reduce the number of frequency bins from the STFT result with minimum loss in information. The improvement in transcription accuracy becomes minimal when the STFT window size reaches 8,192; therefore, our experiment was stopped at 4,096. (Note that a window size of 4,096 results in only 2,048 non-redundant frequency bins.) For CQT, the only tunable parameter is the number of bins per octave when the minimum and maximum frequency are fixed, the number of bins (b) per octave affects the quality factor $Q = (2^{\frac{1}{b}} - 1)^{-1}$, and hence the STFT window size $N_{k_{cq}} = \text{ceil}\left(\frac{s}{f_{k_{cq}}}\right) Q$, where s is the sampling rate and $f_{k_{cq}}$ the frequency for a specific bin. Given that the number of bins per octave is the only tunable parameter for CQT, the CQT result here is same as Experiment 1; The overall best performance is still LogSpec with STFT window size 4,096 and 2,048 bins, closely matched by LinSpec.

C. Experiment 3

The input audio length has only a minor effect on the music transcription accuracy. The length of the input audio affects the width (time-steps) of the spectrogram, and the precision of pitch labels. Fig. 1 shows that a long input frame results in a reduction of accuracy. This could be because the input frame might contain multiple note transitions. Since the pitch label corresponds to the note appearing in the middle of the frame, longer frames might confuse the neural network. Fig. 6(c) shows that decreasing the audio clip length from 16,384 slightly improves transcription accuracy. The transcription accuracy, however, starts to decay once the audio clip is too short. The optimal performance is reached with an input length between 10,000 and 16,384 for LogSpec.

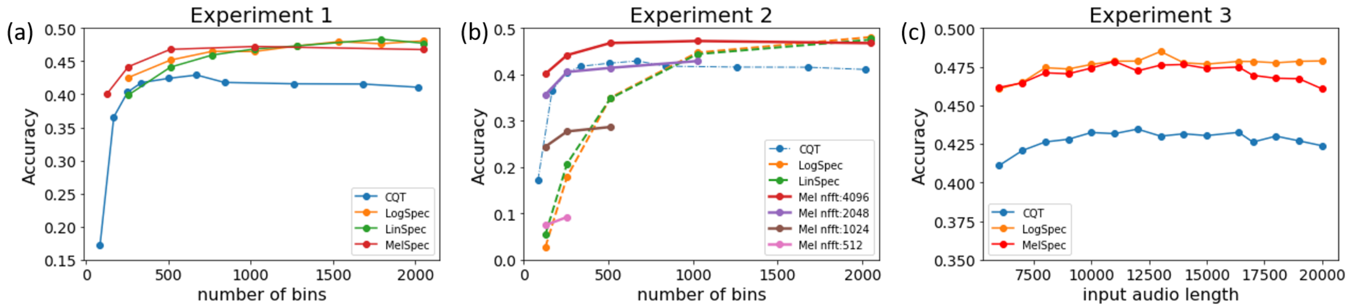

Fig. 6. (a), (b), (c) shows the experimental results for Experiment 1, 2, and 3 respectively, in terms of transcription accuracy.

D. Comparison with state-of-the-art

Given the above results, the best input representations are compared with state-of-the-art models trained on the MusicNet database in Table I. The comparison is made in terms of accuracy and error as defined in the **mir_eval** library. The overall best result is found by using LogSpec and LinSpec based models with n_fft 4,096, and 2,048 frequency bins. When comparing the results obtained from our simple but optimally-tuned networks to the state-of-the-art published results on the MusicNet dataset, only more complex CNN architectures [9] outperform the proposed system in terms of accuracy (other metrics are not given). In future work, we will explore the effects of input representations on more complex models and convolutional architectures.

Model	Precision	Accuracy	Error
MusicNet [8]	65.9	44.4	61.8
Melodyne [29]	58.8	41.0	76.0
Deep Complex CNN [30]	72.9	N.A.	N.A.
LinSpec nfft: 4,096, bins: 512	63.6	44.1	60.7
LogSpec nfft: 4,096, bins: 512	64.6	45.2	59.3
MelSpec nfft: 4,096, Mel bins: 512	65.6	46.8	58.5
LogSpec nfft: 4,096, bins: 2,048	**66.6**	**48.1**	**56.0**
CQT bins per ocatave: 72	61.0	42.9	66.0
MelSpec nfft: 4,096, Mel bins: 1,024	65.8	47.2	58.4

TABLE I

TRANSCRIPTION RESULTS ON THE TEST SET USING THE MIR_EVAL PACKAGE [26]. THE PRECISION, ACCURACY, AND ERROR REPORTED HERE ARE THE FRAME PRECISION, FRAME ACCURACY, AND FRAME CHROMA TOTAL ERROR AS IMPLEMENTED IN THE MIR_EVAL PACKAGE. BEST MODEL PERFORMANCE IS IN BOLD.

V. CONCLUSIONS

This study analyzes the importance of input representations on the accuracy of polyphonic, multi-instrument music transcription. We use our GPU based spectrogram toolkit, nnAudio [16], to calculate on-the-fly spectrogram representations around 100 times faster than traditional libraries. In our experiments, we found that, by fixing the neural network architecture and only varying the input representation, we can improve the transcription accuracy by 8.33% (from 44.4 to 48.1), and reduce the error by 9.39% (from 61.8 to 56.0). When the resolution of the spectrogram is high enough (e.g., a greater number of frequency bins), we found that a log-frequency spectrogram representation ensures a better transcription result, at the expense of computational complexity. If the computational complexity is a concern, MelSpec can compress the spectrogram into a more compact representation while maintaining a relatively high transcription accuracy. A higher level spectral representation, CQT, shows relatively poor transcription accuracy. One reason might be the STFT window size being coupled with the number of frequency bins. Modifying the CQT algorithm such that the number of frequency bins can be varied without changing the STFT window size is one of our future research directions. Neural networks may be able to learn better from a lower level input representation, such as a log-frequency spectrogram or linear-frequency spectrogram. Future research will also explore the influence of additional input representations, such as a learned representation obtained by training the transcription end-to-end with raw audio as the input.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their constructive reviews. This work is supported by a Singapore International Graduate Award (SINGA) provided by the Agency for Science, Technology and Research (A*STAR), under grant no. SING-2018-02-0204. Moreover, the equipment required to run the experiments is supported by SUTD-MIT IDC grant no. IDG31800103 and MOE Grant no. MOE2018-T2-2-161.

REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, “Automatic music transcription: An overview,” *IEEE Signal Processing Magazine*, vol. 36, pp. 20–30, 2019.

[2] A. Klapuri and M. Davy, *Introduction to Music Transcription*. Springer US, 2006.

[3] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, “Automatic music transcription: challenges and future directions,” *Journal of Intelligent Information Systems*, vol. 41, pp. 407–434, 2013.

[4] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription,” in *ICML*, 2012.

[5] G. E. Poliner and D. P. W. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–9, 2006.

[6] ——, "Improving generalization for classification-based polyphonic piano transcription," *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 86–89, 2007.

[7] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 121–124, 2012.

[8] J. Thickstun, Z. Harchaoui, D. P. Foster, and S. M. Kakade, "Invariances and data augmentation for supervised music transcription," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2241–2245, 2017.

[9] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *ArXiv*, vol. abs/1705.09792, 2017.

[10] Q. Wang, R. Zhou, and Y. Yan, "A two-stage approach to note-level transcription of a specific piano," 2017.

[11] 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 *ISMIR*, 2017.

[12] V. Emiya, N. Bertin, B. David, and R. Badeau, "Maps-a piano database for multipitch estimation and automatic transcription of music," 2010.

[13] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *ISMIR*, 2000.

[14] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *ISMIR*, 2016.

[15] B. Balamurali, K. E. Lin, S. Lui, J.-M. Chen, and D. Herremans, "Toward robust audio spoofing detection: A detailed comparison of traditional and learned features," *IEEE Access*, vol. 7, pp. 84 229–84 241, 2019.

[16] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1D convolution neural networks," *arXiv preprint arXiv:1912.12055*, 2019.

[17] D. H. Kin Wai Cheuk, Kat Agres, "nnaudio: A pytorch audio processing tool using 1D convolution neural networks," in *ISMIR–Late breaking demo*, 2019.

[18] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," *ArXiv*, vol. abs/1611.09827, 2016.

[19] S. S. Stevens, J. E. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, 1937.

[20] S. Umesh, L. Cohen, and D. Nelson, "Fitting the mel scale," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, vol. 1. IEEE, 1999, pp. 217–220.

[21] D. O'shaughnessy, *Speech communication: human and machine*. Universities press, 1987.

[22] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, "The htk book," *Cambridge university engineering department*, vol. 3, p. 175, 2002.

[23] J. Youngberg and S. Boll, "Constant-q signal analysis and synthesis," in *ICASSP'78. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3. IEEE, 1978, pp. 375–378.

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

[25] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant q transform," *The Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992.

[26] 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 *ISMIR*, 2014.

[27] P. S. Wright, "Short-time fourier transforms and wigner-ville distributions applied to the calibration of power frequency harmonic analyzers," *IEEE transactions on instrumentation and measurement*, vol. 48, no. 2, pp. 475–478, 1999.

[28] C.-i. Wang and G. Tzanetakis, "Singing style investigation by residual siamese convolutional neural networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 116–120.

[29] "Melodyne," http://www.celemony.com/en/melodyne/what-is-melodyne, accessed: 2019-10-08.

[30] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," *arXiv preprint arXiv:1705.09792*, 2017.