REAL-TIME PIANO TRANSCRIPTION FOR MUSIC PERFORMANCE

Ko Muramatsu
Eastman School of Music
kmuramat@u.rochester.edu

ABSTRACT

Automatic music transcription of the piano has been addressed by several researchers and has proven improvement in accuracy. Previous researchers achieved state-of-the-art results by employing Transformer architectures and treating the transcription task as a sequence-to-sequence prediction of events, mainly onsets and offsets of each of the 88 keys. However, the preceding study preliminarily builds upon offline predicting implementations, thus it is not practical to use in real-time music performance. The present work discusses a potential streaming model for the piano AMT task, following the preceding study. Critical observation of the trade-off between the accuracy of the result and the latency of the processing is necessary for the discussion, and a desired product of the task must be pre-considered to determine the priority. While the goals can be and should be different for individuals, I aimed to achieve a stable and fast enough model to use in real-time performance with above-minimal bearable preciseness.

I. INTRODUCTION

For many musicians, including myself, Artificial Intelligence is something unknown technology in a black box. Typical musicians' reactions toward AI would often be associated with antagonism, fear, doubt, or complete indifference. These negative assumptions on AI generally come from the belief that "art can only be made by humans." Whether or not it is true, early propaganda on AI was effective and had a strong enough impact on people to believe that "AI thinks themselves and makes decisions in lieu of human,' which not arguably, has exacerbated the feeling of artists as it aesthetically endangers the belief of the human autonomy on a piece of art.

Admittedly even though the propagandized statement makes some sense at least in a rhetorical way, learning more and more about neural networks convinced me that AI does not think themselves, but they are TRAINED to reproduce the desired results of (a) human. Thus, knowing how a neural network works and how it is trained is important, not only for engineers but also for the people who have sensitive minds wishing to distinguish "who made what." Artists also should not neglect AI's huge impact on contemporary society and the possibility of neural networks as well, either if they would like to succeed in a market that has been heavily commercialized by the power of AI or if they would be willing to challenge themselves to work with the developing technology for creative purpose. Beatles took full advantage of the high-end electronics placed in Abby Road Studios back then; Beethoven understood the function and character of the modern piano, particularly apparent in the low register, and so on and so forth. As the modern piano was one of the products of the Industrial Revolution and the early electroacoustic gear had an inseparable relation to radio and military technology for telecommunication, the medium of art itself can somewhat reflect the culture and influence of contemporary society. This leads to the primary motivation of the project: To use the AI model as a creative tool and incorporate it as an artistic challenge.

Attempting state-of-the-art technology with an innovative theory can legitimately be a clear motivation, especially for engineers and mathematicians I suppose, but having the creative motivation and goals was undoubtedly helpful during the working process on the project. Essentially, the transcription model is an intermedium between input audio and output labels, so knowing "what the outputs should look like for the music I want" allowed me to have a clear direction and prioritization of the project. For example, it was clear to me that the pedal prediction of the model should not be a binary prediction because the amount of pedaling is very important for rich expression in piano music. Also, the importance of onsets should apparently be more than that of offsets in the model. It is because even though the information about offset timing (i.e., duration of a note) can be an interesting parameter for expressiveness associated with articulation, it would be less explicitly represented by those instruments with short release such as piano and guitar. In addition, I preferred to randomize the velocity and durations of notes in the performance using the model, thus the precise offset prediction is less prioritized in my

approach. Above all, shorter processing time and stability of the model architecture during run are essential for this project to be a reliable tool for real-time performance.

Given the conditions and considerations, I acknowledge that I followed the streaming piano events detection model proposed by Weixing Wei [3], with two modifications. First, the pedal prediction is made as continuously changing values as already mentioned. Second, the receptive field and convolutional layers in the encoder are reduced by half to mitigate the latency when running the model. In the realization stage, I used MaxMSP to process the detected events through OSC messages between Python and MaxMSP, having MIDI instruments semi-automatically perform based on the detected pitches and additional pitches based on the predictions. Onsets of the input audio are observed in MaxMSP and trigger output notes with slightly randomized and generalized velocity, taken from the loudness of the onsets.

II. METHOD

The proposed method uses a CNN encoder and two Transformer decoders, one for onset detection and the other for offset and pedal detection. Transcription takes input audio data and outputs MIDI labels.

2.1 Pre-Processing

Input audio is pre-processed before being fed into the CNN in the encoder. The pre-processing takes a few steps: resampling the audio into 16,000 Hz, converting audio into a single channel, normalizing, and applying constant-Q transform (CQT) to obtain a spectrogram. For CQT, hop size is set to 320 (20 ms) and minimum frequency to 27.5, the lowest note in the piano, and 48 bins for each octave. CQT provides promising spectrum information over FFT as the geometrical spacing between each of the bins is equally proportional to log-scaled frequency, corresponding to the pitch system of music and human auditory perception.

2.2 Encoder

The CNN layer is comprised of eight 2-D convolutional filters with additional dilated convolutional layers. The layering structure is borrowed from HPP-net proposed by Wei [4], in which the dilated convolutional layers are called Harmonic Dilated Convolution (HD-Conv), and the dilation ratio is calculated based on the overtone structure of piano notes. Wei suggests that the use of HD-Conv in HPP-net reduces a significant amount of memory processing while successfully capturing meaningful harmonic information. After the convolutional filters, positional encoding is applied.

In the present project, I started to model the encoder to process within a receptive field (M) of 40, which I believe is not enough number for the design of the convolutional layers but supposedly suffices the requirement for the task. However, the training and evaluation encountered a problem with time resolution later in the experiment, and the latency is critical for the aim of the project. Therefore, I reduced the M to 20 and the number of layers to 6. While this modification resulted in a faster conversion in training and resolved most of the latency issues, the structure of the encoder needs fine-tuning, as the demo running of the trained model produced recognizable octave misrecognition.

2-3. Decoder

Two Transformer decoders output onsets, offsets, and pedal events, in a sequence-by-sequence manner.

2.3.1 Onset Decoder

The onset decoder takes the encoder-decoder attention mechanism on the hidden dim = 256 from the encoded representation and the previous Onset Tokens **Yt**. Positional encoding is applied to **Yt** before the cross-attention. Linear layer (output dim = 90, including BOS and EOS tokens in addition to 88 tokens representing piano keys) and sigmoid activation are applied to obtain the probability of activated keys. After thresholding the probability, tokens representing the Onset Events **y** are obtained. The token prediction is recursively processed until either 1) a special token EOS is obtained or 2) the number of recursions reaches a certain number. Currently Activated Onsets **A** and **Yt** are updated with the obtained tokens.

Max number of recursions is originally set to 64, but it is later reduced to 32 in the experimentation, considering that the plausible number of onsets within the time frame likely does not exceed 32 in most of the cases in piano performance. Also, **y** inclined to be the same in most of the recursions, which makes the early-stage training stagnate. Removing the duplicating **y** from **Yt** helped the issue and stabilized the training. Learning parameters for the threshold over probability dynamically adjust for different pitch ranges in order to reconcile the low confidence in the extreme register, however, it was not as successful as I assumed.

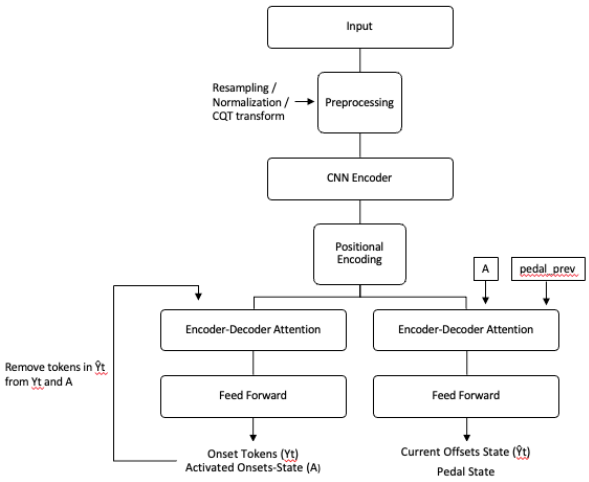

Figure 1. CNN Encoder and Two Transformer Decoders

2.3.2 Offset Decoder

The offset decoder has a similar structure to the onset decoder; The encoded representation of input is decoded with the Currently Activated Onsets **A** and pedal prediction from the previous frame, instead of **Yt**. The Offset Tokens **Ŷt** is predicted all at once in the offset decoder, unlike the onset decoder. Combining offset and pedal detection intuitively makes sense because pedaling of the piano sustains the sound, blurring the offsets of notes.

2.3.3 Interdependency between Onset and Offset Decoders

Offset Tokens in **Ŷt** in the previous frame are removed from **Yt** and **A**. Offset Tokens are predicted from **A**, thus they are interdependent with one another. Even though the model takes into account the predictions from only one frame back, the interdependency enhances the accuracy of the model's prediction. I assume this logic can be suitable for musical instruments with short attack and release, such as the piano and mallet-percussion instruments.

Later in the experimentation, I added an additional token removal operation based on the pedal value from the previous frame, which helped to make the sequence of tokens in **Yt** more concise, improving both efficiency and accuracy.

III. EXPERIMENT

3.1 Dataset

The dataset used for training is MAESTRO, which contains about 200 hours of paired audio and MIDI labels, including onsets/offsets temporal information for each key and sustain pedal values (values in CC: 64). The distribution of the dataset is 962, 137, and 177 for training, evaluation, and testing, respectively.

3.2 Training

Ground truth MIDI labels are prepared to align with the time resolution of the model. The time resolution is calculated from the hop size set in the CQT (20 ms) and receptive field (20): The ground truth labels are downsampled into 400 ms frames with max pooling. The training utilizes BCEWithLogitsLoss for the binary class predictions of onsets and offsets, while MSELoss is used for pedal value prediction. Weights on the three losses are critical for the convergence. Pedal prediction struggles with attaining stability, by which offset detection stagnates. However, the pedal prediction is not as important as the other two predictions, thus I started from a higher weight on the pedal in the early stage of training and gradually decreased it to provide more room for onset and offset losses to improve.

Also, the early experimentation suffers from predicting too many False states and eventually predicting no events. In piano music repertoire, False states dominate True states, and this is especially problematic in the extreme pitch ranges as those high/low pitches are often used less frequently compared to the middle range. During the experimentation, I found that occasionally there are no onsets or offsets at all in short segments of the audio chunks. While there can be several ways to address this data imbalanced issue, I set pos_weight in the loss function of the onsets and offsets. It significantly mitigated the issue of overconfidence on False states.

3.3 Results

Figure 2 shows the exemplary prediction from the model, showing the best result (M = 20, overlap = 0.0). Detected pitches are generally in correct places. The pedal prediction is loose for the different time resolutions between ground truth and the model, but hard clamping the values between 0.0 and 0.5 into 0.0 would help better realization, since the values below 0.5 globally mean no sustain pedal when converted into MIDI values.

Two post-processing techniques are applied before obtaining the final result demonstrated here. First, predicted onsets and offsets with low confidence are removed from the predictions. Table 1 shows the evaluation score of the result, and the recall score is dominant over the precision, most likely due to too much weighting of the True states after setting pos_weight. Even though a higher recall score is

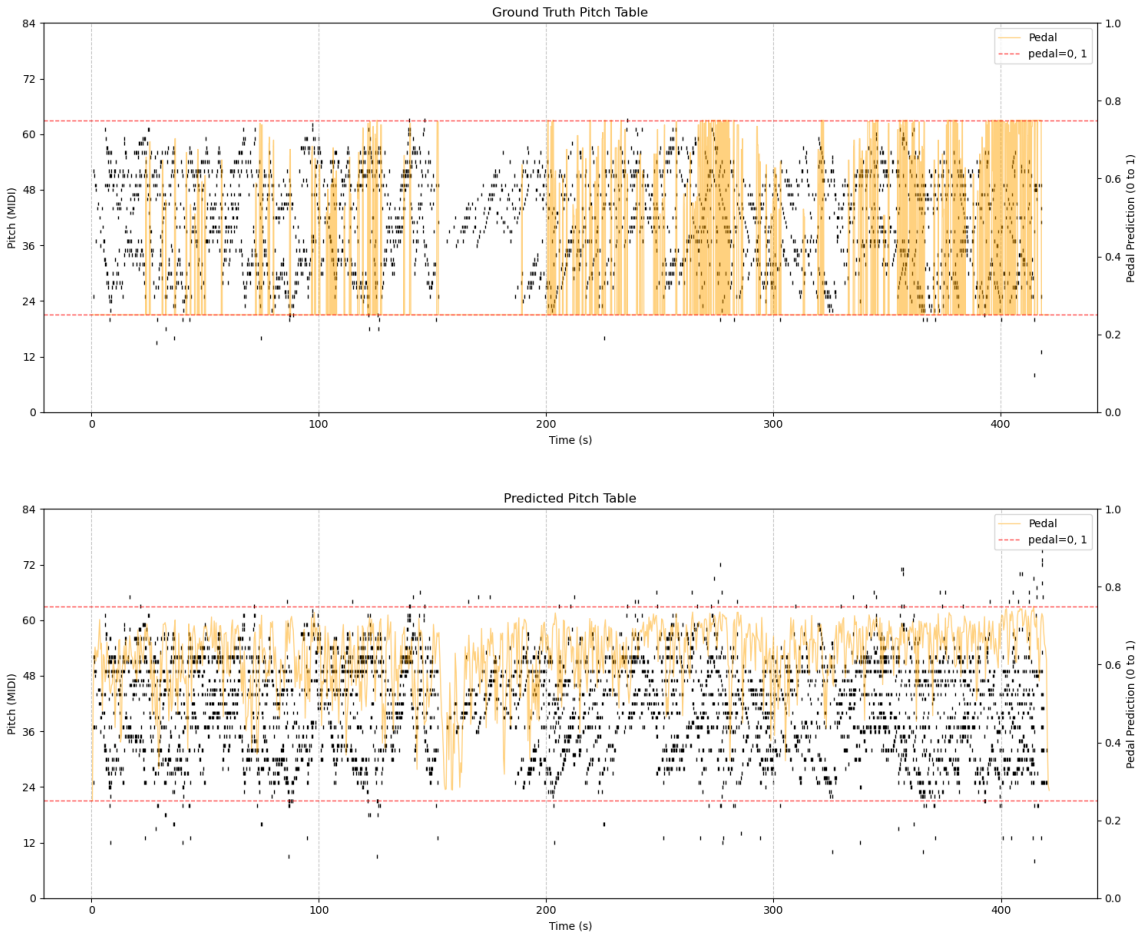

Figure 2. Comparison betwen ground truth and predicted pitch tables

helpful for live performance and processing in the realization stage as those "mistakenly" detected pitches can contribute to the musical richness depending on context, the imbalance between the two scores is too much. The first post-processing improves the balance between the scores, clearly removing the noisy prediction in extreme pitch register and redundant repetitive events detection. Second, offsets are compensated after the prediction if the detected onsets do not have corresponding offsets. The compensation is made in either of the following two cases: 1) 0.8 or more seconds after the onset is activated, pedal-off state is detected: and 2) 2.0 seconds after the onset is activated, the onset is still active. By this compensation approach, the result avoids representing notes with unrealistically long durations. These two post-processing techniques need careful fine-tuning and improvement, and, they have a huge impact on the output sound in the realization stage as well. To note, it evaluates the onset prediction and does not take offset prediction into consideration. It is because I have not implemented a way to use the offset information in the realization stage, thus the evaluation score with offset detection does not provide a meaningful context in my usage of the model.

IV. REALIZATION

https://youtu.be/pYZuahUBb_8
(Demonstration video of the realization of the model in musical context, December 2024 in Eastman School of Music, © Ko Muramatsu 2024. All rights reserved.)

The detected events are sent to MaxMSP through OSC messages using a UDP connection, and the software is in charge of adding musical context to the predictions (e.g., add/remove the pitches from the predictions, control velocity, change timbre). Ideally, the change in the handling of different settings in MaxMSP and transitions from one scene to another should be smooth. To note, the output of the notes is triggered by onset slices detected in MaxMSP (not from the transcription model), thus the timing of the note output is not simultaneous with the timing of the events detection. This setting allows more organic and

Model	w/o post-processing			w/ post-processing		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
PPTStreamingModel (proposed)	46.4	88.5	60.5	70.7	79.6	75.5
Streaming Sq2Seq (state-of-the-art model)	N/A	N/A	N/A	98.30	94.83	96.52

Table 1. Evaluation Scores

flexible control over the musical events, meanwhile, the time adjustment between the prediction and onset slices is an issue to be addressed.

In the demonstration, I ran the model with a half overlap and segment size=18,432 under a sampling rate of 48k Hz. The segment size is determined as the number that is a multiple of 1024, a kernel size of the CQT, and that is equal to or larger than 18,240, the number of samples that is required for the receptive field fed into the encoder. In this setting, the hop size of the prediction is 0.192 seconds, and the model's processing time is approximately 0.015 seconds. Even though this is reasonable latency for real-time performance in theory, the demonstration showed a recognizable amount of delay between the live performance and the realization of the detected pitches. The delay is likely caused by 1) the timing misalignment between events detection and onset slices, 2) latency in UDP communication, and/or 3) potentially wrong configuration somewhere in my current setting.

V. CONCLUSION AND FUTURE IMPROVEMENTS

The trade-off balance between accuracy and latency of this presented work is practical and proved to be useful for real-time music performance, even though the evaluation score is not high enough to be considered as a counterpart to the state-of-the-art result in the latest research. Nevertheless, it is apparent that the presented model has room for improvement both in accuracy and latency. First, the size of the receptive field and the encoder structure with dilated convolutional layers can be optimized. Second, my decoder structures may well contain redundancy and there must be a better implementation for both computational efficiency and accuracy. Third, fine-tuning thresholds and weights during the training was pretty much a trial-and-error process in my experimentation, which is not the best practice for concluding the final outcome of the model. The accuracy of offset detection is particularly unsatisfactory in the current trained model, clearly proven by the imbalance between precision and recall scores. Lastly, the timing alignment issue between model predictions and processing within MaxMSP can be reconciled with finer adjustments.

Still, the project presents the potential for creative music performance, in which the live performer and the semi-automated MIDI instruments (considered as a virtual performer) build a closely interactive relationship. Possible future explorations for the expressive music performance would be in 1) use of Markov Models and/or HMMs to "predict" upcoming pitches based on the notes played by the performer, 2) experiment with offsets that provide more content on articulations and durations, in addition to velocity, 3) timbre variation during the performance, 4) control over the delay time between the two performers, 5) use the pitch information as parameter for audio effect, such as spectral reverb on certain pitch-class notes and formants-fixed phase vocoder, and more.

The use of a neural network model as a creative tool opens up the exciting potential for any kind of music and artists, and I hope this project will provide optimistic insights for future study, both in music and engineering domains.

VI. REFERENCES

[1] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W.-H. Liao, and Y. Mitsufuji, "Automatic Piano Transcription with Hierarchical Frequency-Time Transformer," *arXiv:2307.04305*, 2023.

[2] Y. Yan and Z. Duan, "Scoring Time Intervals using Non-Hierarchical Transformer For Automatic Piano Transcription," *arXiv: 2404.09466*, 2024.

[3] W. Wei, J. Zhao, Y. Wu, and K. Yoshii, "Streaming Piano Transcription Based on Consistent Onset and Offset Decoding with Sustain Pedal Detection," in *ISMIR 2024*, November 2024. [Online], *Ismir.net*, https://ismir2024program.ismir.net/poster_18.html#paper [Accessed Nov, 2024].

[4] W. Wei, P. Li, Y. Yu, and W. Li, "HPPNet: Modeling the Harmonic Structure and Pitch Invariance in Piano Transcription," *arXiv: 2208.14339*, 2022.