# Real-time Automatic Piano Music Transcription System

**Dasaem Jeong**
SK Telecom
dasaem.jeong@sktbrain.com

## ABSTRACT

Research in automatic music transcription (AMT) showed significant improvement thanks to advances in deep learning. However, most of the research was designed for an offline scenario, where the input audio recording is provided from beginning to end. In this paper, we present an online piano AMT system with visualization using a web browser and MIDI export which works on CPU in real-time. We employed a model with auto-regressive LSTM and multi-note-state, which is adapted for an online scenario without losing its accuracy.

## 1. INTRODUCTION

An automatic music transcription (AMT) is a task that transcribes note information such as pitch, onset, and offset from an audio recording. Although the task of AMT is not limited to a specific instrument, most of the research [1–4], focused on piano music, because of its timbral characteristics and large public datasets such as MAPS [5] and MAESTRO [6].

With the recent exploitation of deep learning, the accuracy of piano AMT systems showed significant advances in terms of its accuracy [7–9]. The improved performance of the AMT model also shows the reliable result on real-world audio recordings, thus enables web application that transcribes piano audio recording to MIDI file [1] .

However, our goal is to implement a piano AMT system that works in an online scenario with real-time computation and visualization. Unlike the offline scenario, the real-time application can demonstrate the AMT system with more interactions with the user. In this paper, we introduce our implementation of the real-time piano AMT system, which was presented in SKT Tech Gallery in Pangyo, Korea. The code of our system is available in GitHub repository [2] .

[1] https://piano-scribe.glitch.me/
[2] https://github.com/jdasam/online_amt



## 2. AUTO-REGRESSIVE MODEL

Recent research [9] proposed a deep-learning-based AMT model with an auto-regressive (AR) RNN. The AR connection feeds the transcribed result of the previous time frame into the activation of the following time frame. Unlike models from the previous research [7] that predict the result of the entire frame at once in the final step, the AR model predicts the transcription output in frame-by-frame by its nature. Therefore, the AR model can be easily adapted for the online scenario without losing accuracy. Mathematically, the computation process is identical regardless of whether it is in an online or offline scenario.

Another difference is the type of output prediction of the model. The previous model [7] predicts onsets and frames of note in two different modules, thus demands a post-processing for note-level reconstruction. On the other hand, the AR model [9] directly estimates the note states like off, onset, sustain, offset, and re-onset by each pitch for every frame. Therefore, it does not need any post-processing for note-level reconstruction.

Our online AMT system employs the same neural network architecture proposed in [9]. The model was trained with MAESTRO V2.0.0 [6], which composed of 200 hours of audio and corresponding MIDI files of piano performances. We have tested total 162 combinations of hyper-parameters for the training, such as different learning rate, weight decay, number of CNN channels, and number of hidden nodes in LSTM and selected the model with the highest note-with-offsets F1 score.

## 3. IMPLEMENTATION FOR REAL-TIME

One of the most important issue to be solved in online AMT system is that the entire computation has to be handled in real-time with low-latency without degrading model's accuracy. Our goal was to convert the model proposed in [9] to real-time online system works on CPU.

The hop size of mel-spectrogram is identical with the length of each time step in the language module of the model. The language module consists of two layers of auto-regressive uni-directional LSTM. The hop size is also equivalent to the time resolution of the transcription. The previous models [7, 9] was trained with hop size of 512 samples in 16kHz sample rate, which is 32 ms. Increasing the hop size reduces the computation cost because the number of time frames for given audio length decreases. However, we wanted to preserve the time resolution and therefore used the same hop size.

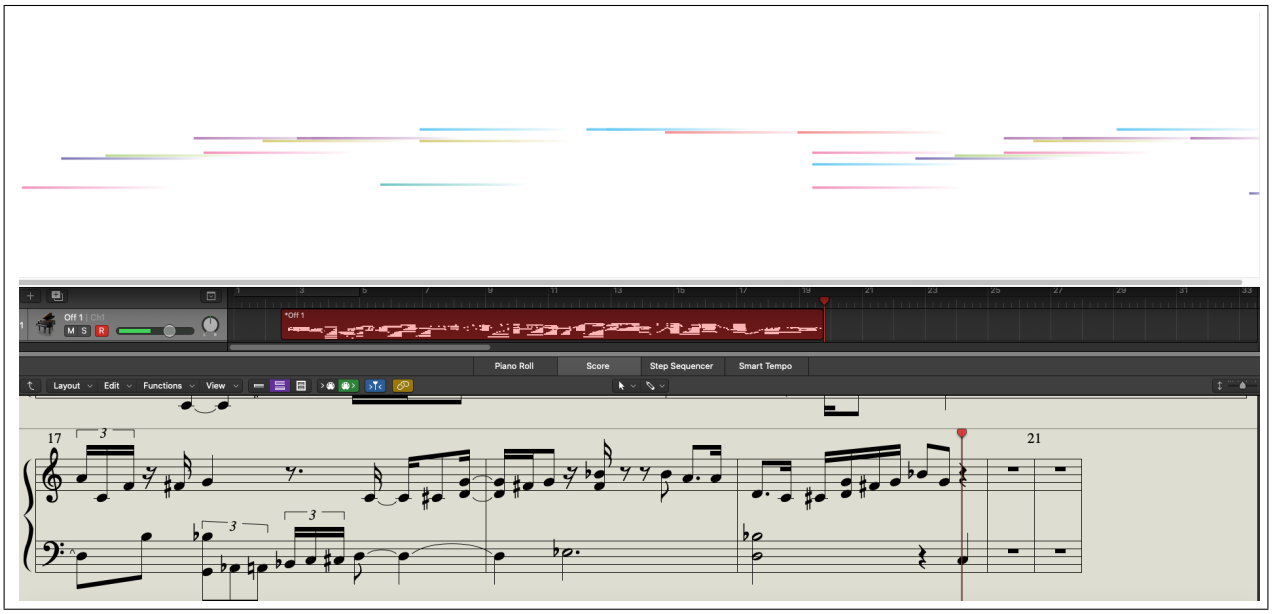


**Figure 1**. Screenshot of visual representation of our system. The upper half is a piano roll rendered on web browser and the lower half is a music score rendered on Logic Pro X.

To fully exploit the model's capacity, the online system has to feed enough length of audio samples to the model. The AMT model we used takes a mel-spectrogram with hop window size of 2048 and hop size of 512 in 16kHz sample rate. The acoustic module of [9] consists of three layers of convolutional neural networks (CNN) with kernel size of three frames in time axis. Therefore, a field of view of the final layer is seven frames in the time axis of mel-spectrogram, which corresponds to 5120 samples, or 320 ms of audio in 16kHz sample rate.

To optimize the computation process, our system uses audio buffer of 5120 samples that is updated for every 512 samples. The audio buffer is converted into mel-spectrogram with 221 bins and 7 frames. When the new 512 samples are updated, only the last frame of mel-spectrogram is updated while the others are shifted for one frame. We also update the output of each CNN layers for only the last frame to reduce the computation. Our system is implemented with `PyAudio` for handling micropohne input, `PyTorch` for neural network models and `librosa` for converting audio samples to mel-spectrogram.

The whole computation process for every updated 512 samples should be handled within 32 ms. We found that our system handles the update audio buffer less than 12 ms in average with MacBook Pro 13-inch 2018 with 2.3 GHz quad-core Intel Core i5.

## 4. VISUALIZATION AND MIDI EXPORT

A piano roll, which is a floating 2D plot of note information in time and pitch axes, is the most typical visualization scheme used for presenting the AMT result. Since python has disadvantages in rendering fluent animation, we used JavaScript and web browser for the visualization. While the microphone input and the AMT model are handled in Python, the JavaScript module calls the result of AMT and renders the result as a piano roll on the browser, as presented in Figure 1. We applied gradation on the note so that a re-onset class, which is an onset following a sustain frame without an offset, is visibly distinguishable. The color of each note is decided by the pitch class. The mapping of color and pitch class is designed by considering circle of fifth, so that close pitch classes are represented with similar color.

In order to show that the transcribed result is in form of symbolic music, although it lacks metric information, we employed Logic Pro X for the additional visualization. Logic Pro X renders the input MIDI signal as a music score, although the metric information is converted based on the fixed metronome beat. The Python module sends MIDI signals to internal MIDI ports with `RtMidi` and Logic Pro X renders the MIDI signals as a music score. The MIDI export can be used not only for the visualization, but also as a recording or a real-time MIDI converter.

## 5. CONCLUSION

With our implementation, we demonstrated that the recent AMT model can also be adapted for a real-time online scenario. We hope that our system can be utilized for various purpose, such as monitoring the AMT result in more interactive way or designing a novel listening experience with the piano music.

## 6. ACKNOWLEDGMENTS

I appreciate to Changhyun Kim and Jaehoon Choi in T-Brain X for organizing the demonstration of this work, and Taegyun Kwon in KAIST for sharing his insights on the AR AMT model. The model was trained with the meta learner developed by SK Telecom Vision AI Labs.

## 7. REFERENCES


[1] C. Raphael, "Automatic transcription of piano music," in *Proc. of the 3rd International Conference on Music Information Retrieval (ISMIR)*, 2002.

[2] S. Abdallah and M. Plumbley, "Polyphonic music transcription by non-negative sparse coding of power spectra," in *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*, 2004, pp. 318–325.

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

[4] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *Proc. of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, August 2016, pp. 475–481.

[5] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, September 2010.

[6] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proc. of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[7] 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. of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, October 2018, pp. 50–57.

[8] J. Kim and J. Bello, "Adversarial learning for improved onsets and frames music transcription," in *Proc. of 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 670–677.

[9] T. Kwon, D. Jeong, and J. Nam, "Polyphonic piano transcription using autoregressive multi-state note model," in *Proc. of the 21st Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2020.