MLodists

TONES 2 NOTES

Automatic Music Transcription

PRESENTED BY

- Atishay (210050026)
- Avadhoot (210050027)
- Megh (210050055)
- Harshit (210050062)
- Veeresh (210050163)

CONTENTS

OVERVIEW

- Automatic Music Transcription refers to task of transcribing audio recordings into symbolic representations (musical notes/MIDI)
- MIDI stands for Musical Instrument Digital Interface
- It is a standard to transmit and store music, originally designed for digital music synthesizers
- Given a music file, the deep learning model transcribes it in MIDI file
- Transcription requires appropriate analysis of audio and processing
- Four models are tested and their results are compared in this project
- The architecture of models and processing techniques are based on the following paper - High-resolution Piano Transcription with Pedals by Regressing Onset and Offset Times by Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, Yuxuan Wang

```
'note_on channel=0 note=74 velocity=50 time=4169',
'note_off channel=0 note=74 velocity=50 time=52022'
'note_on channel=0 note=71 velocity=51 time=0',
'note_on channel=0 note=52 velocity=31 time=20713',
'note_on channel=0 note=55 velocity=36 time=3689',
'note_off channel=0 note=62 velocity=40 time=3588',
'note_on channel=0 note=62 velocity=37 time=0',
'note_on channel=0 note=67 velocity=39 time=3487'
```

DATASETS

We found these main datasets available for piano transcription

MAPS

provides recordings with CD quality (16-bit, 44-kHz sampled stereo audio) and the related aligned MIDI files as ground truth. The overall size of the database is about 40GB, i.e. about 65 hours of audio recordings.

MusicNet

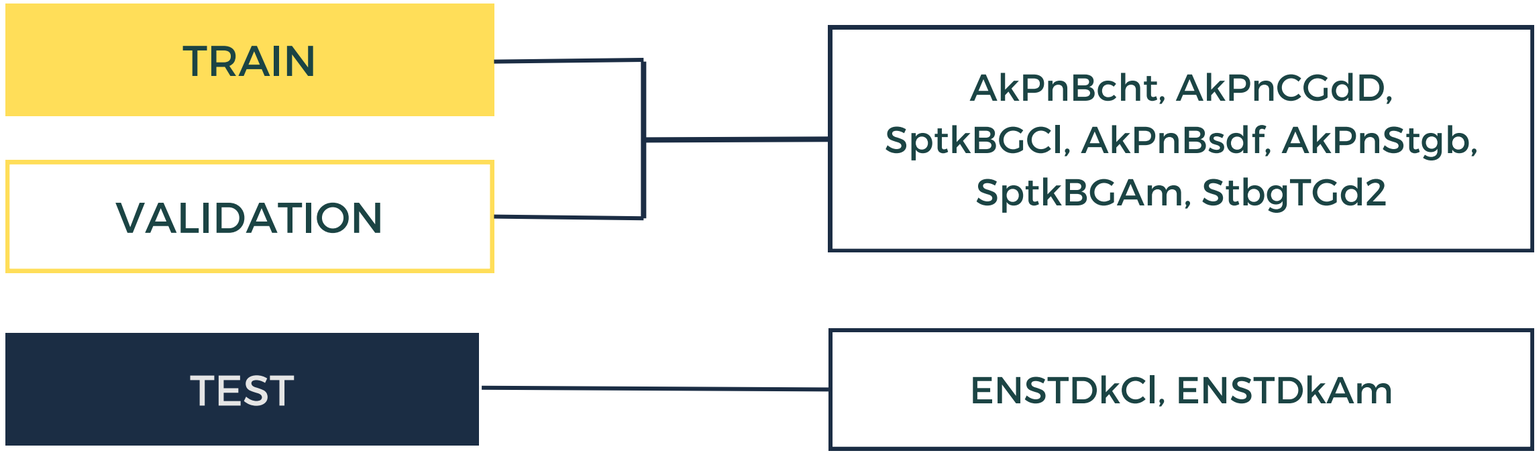
MusicNet is a collection of 330 freely-licensed classical music recordings, together with over 1 million annotated labels indicating the precise time of each note in every recording, the instrument that plays each note, and the note's position in the metrical structure of the composition.

MAESTRO

The dataset contains about 200 hours of paired audio and MIDI recordings from ten years of International Piano-e-Competition .Audio and MIDI files are aligned with 3 ms accuracy.
https://magenta.tensorflow.org/datasets/maestro

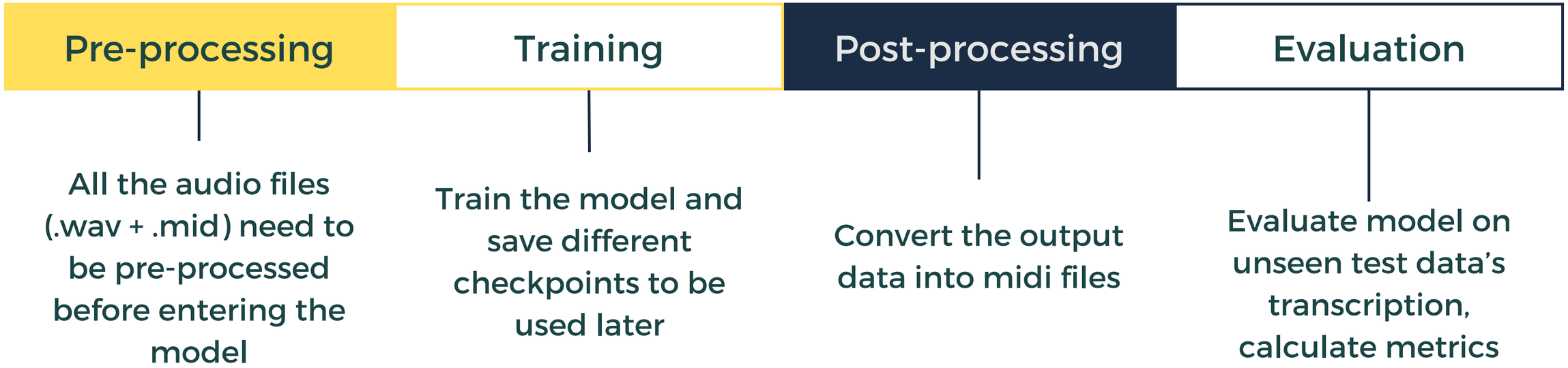
DATASET USED - MAPS

- We decided to use the MAPS dataset because it contains a diverse set of piano recording with different pianos, performers and recording conditions.
- Used the ‘MUS’ set for each folder, consisting of recordings made in music university studio settings.
- The MAPS dataset was split as -



- After splitting, packed the read midi files and audio for each song into .h5 files
- Attributes as - File name, Split, Duration, Audio Waveform, MIDI event dictionary

PIPELINE


Pre-processing
Training
Post-processing
Evaluation
All the audio files (.wav + .mid) need to be pre-processed before entering the model
Train the model and save different checkpoints to be used later
Convert the output data into midi files
Evaluate model on unseen test data's transcription, calculate metrics

PRE-PROCESSING

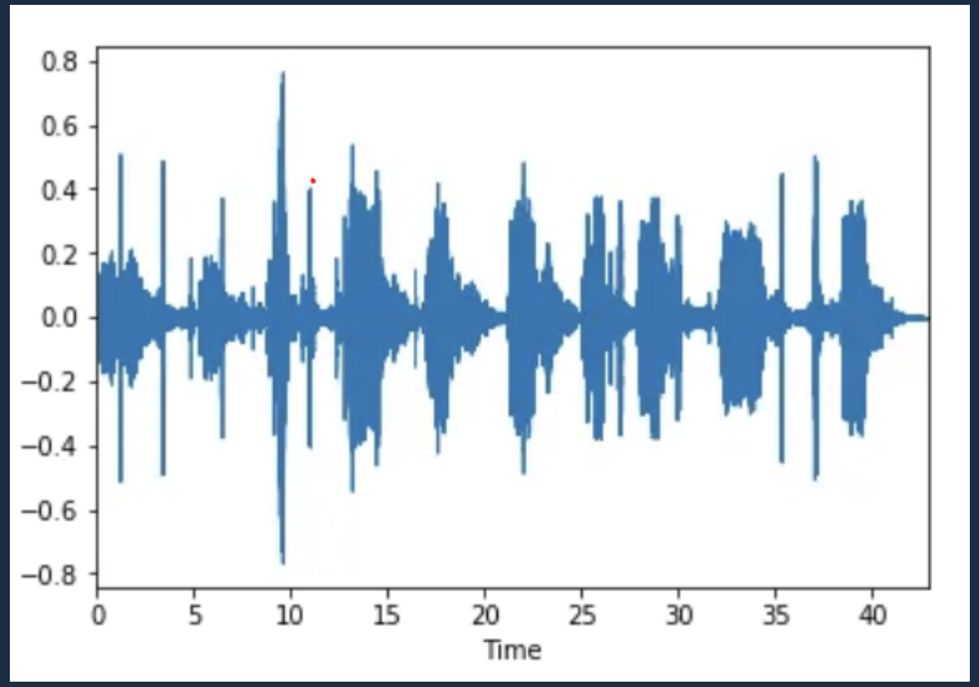
Summary

- Used librosa.core.load() to convert .wav audio input into a storable array (waveform).
- Used mido's MidiFile class to extract and store events (note onsets and offsets) and their time, from the target .mid audio file.
- Stored the waveform, midi events, and metadata for every input as a HDF5 binary file.

- As part of pre-processing, audio recordings are converted to monophonic, and resampled to 16kHz.
- Audio recordings (waveform) split into 10-second clips, to be fed as inputs to the model.
- Short time Fourier Transform applied to extract Spectrogram, and then processed to output a log mel spectrogram of size (1001, 229).
- Feature extraction of midi events, to obtain target dictionary of onsets, offsets, velocities and frames for every 10-second duration.

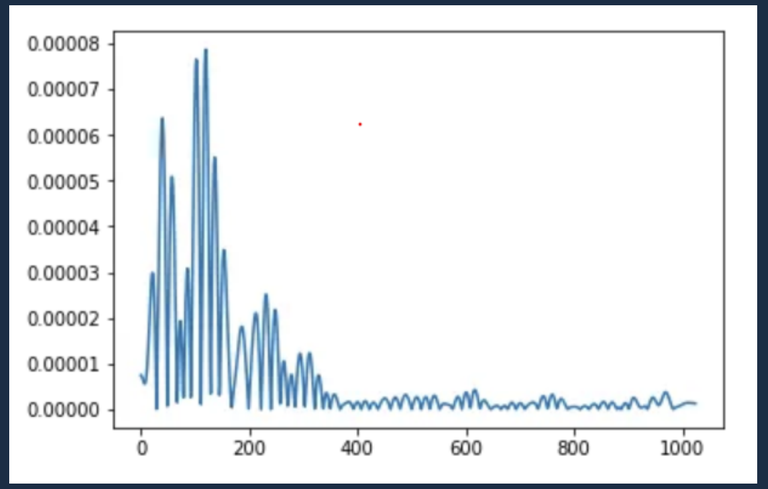
UNDERSTANDING SPECTROGRAMS

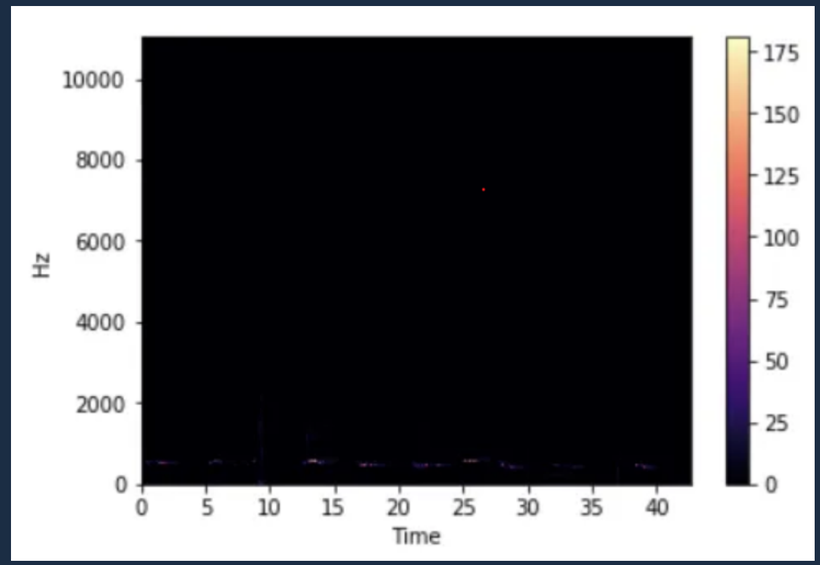
- When we talk about sound, we generally talk about a sequence of vibrations in varying pressure strengths, so to visualize sound kinda means to visualize airwaves



A simple 2D representation of an Audio

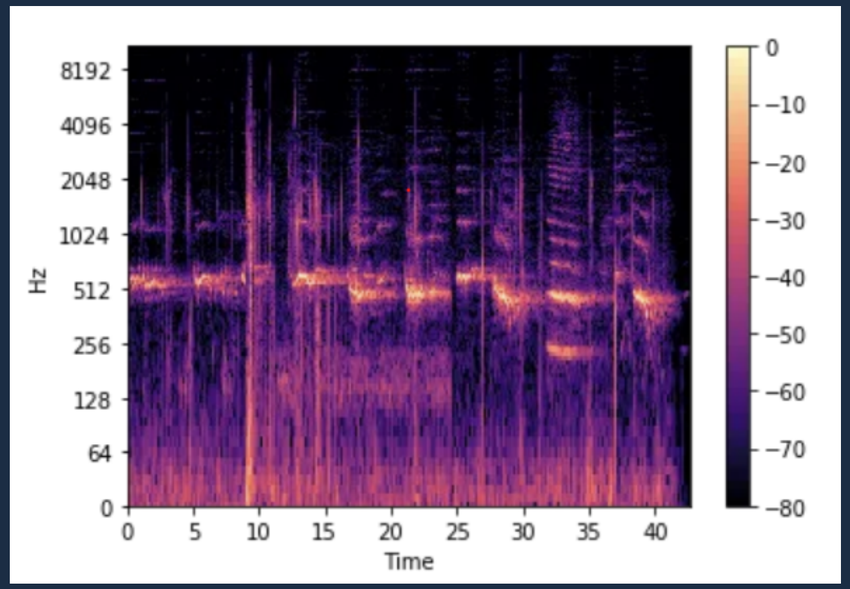
- One mathematical transformation of sound is the Fourier Transform, which is a function that gets a signal in the time domain as input, and outputs its decomposition into frequencies.





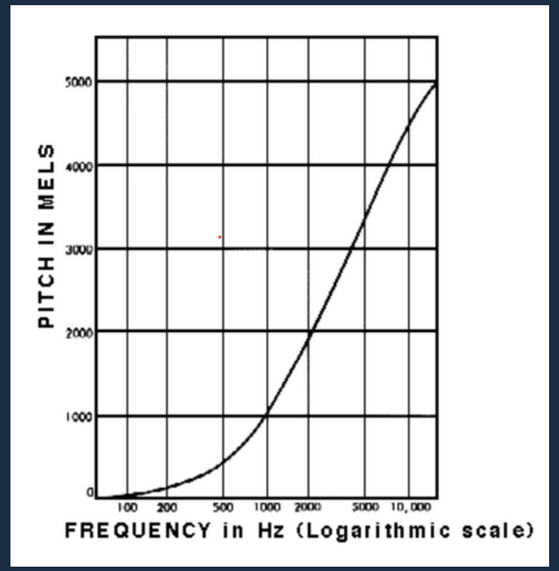
Fourier Transform Representation

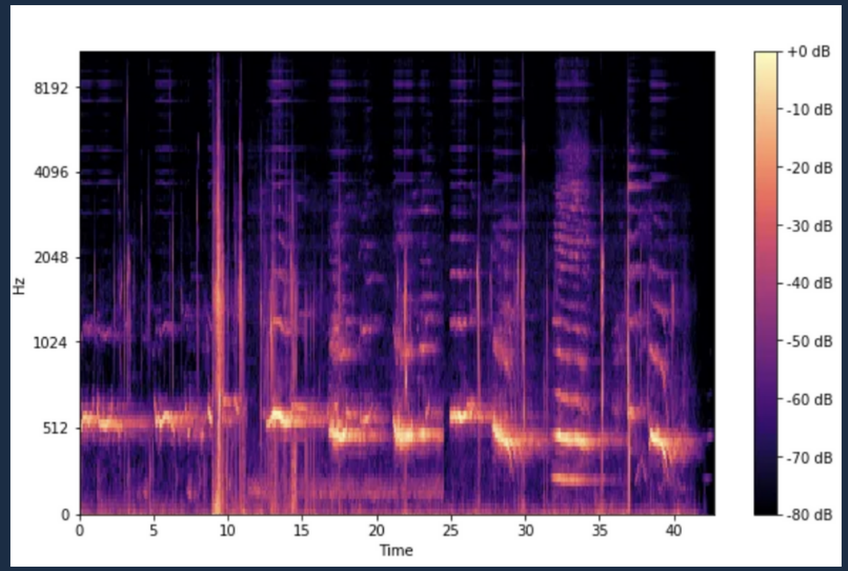
- Wow can’t see much there can we? It’s because most sounds humans hear are concentrated in very small frequency and amplitude ranges
- To make the range more perceivable and understandable by humans, transform both the y-axis (frequency) to log scale, and the “color” axis (amplitude) to Decibels (which is kinda the log scale of amplitudes).



Spectrogram

- Studies have shown that humans do not perceive frequencies on a linear scale. We are better at detecting differences in lower frequencies than higher frequencies.
- The Mel Scale, mathematically speaking, is the result of some non-linear transformation of the frequency scale.
- This Mel Scale is constructed such that sounds of equal distance from each other on the Mel Scale, also “sound” to humans as they are equal in distance from one another.
- A ‘mel spectrogram’ is a spectrogram where the frequencies are converted to the mel scale.





Log Mel Spectrogram

WHY MEL - SPECTROGRAM

Summary

- We used the MEL spectogram as it was used in the paper
- It provides a representation of audio data that's more aligned with human perception, as follows:
  - **Frequency Compression:** Mel spectrograms use the Mel scale, which is a perceptual scale of pitches that reflects how humans hear different frequencies
  - **Better Representation:** They capture important acoustic cues that are crucial for speech and sound recognition
  - Dimensionality Reduction: Mel spectrograms often have fewer dimensions compared to traditional spectrograms
  - Feature Extraction: Mel spectrograms serve as a feature extraction method, providing a way to represent complex audio signals in a format that machine learning algorithms can better interpret and learn from
- Overall, Mel spectrograms are a bridge between raw audio data and meaningful features that can be used effectively in machine learning models for tasks like speech recognition, music genre classification, sound event detection, and more

HIGH-RESOLUTION PIANO TRANSCRIPTION

- In most cases, input songs are represented in the form of frames (short durations of audio).
- A note onset (attack) or a note offset (delay) may span over multiple frames (though peak somewhere at one frame).
- Standard ways of denoting an offset or offset don’t account for this span across frames.
- Ways that do, don’t account for the fact that a frame itself extends over a span of time.
- Therefore, the true peak of an onset or offset is not at a frame, but at some point of time within a frame.
- To account for these subtleties, the following method has been proposed:
  - “A Regressed array representing a span of attack or delay based on the distance (time) from the estimated true onset or offset peak time”

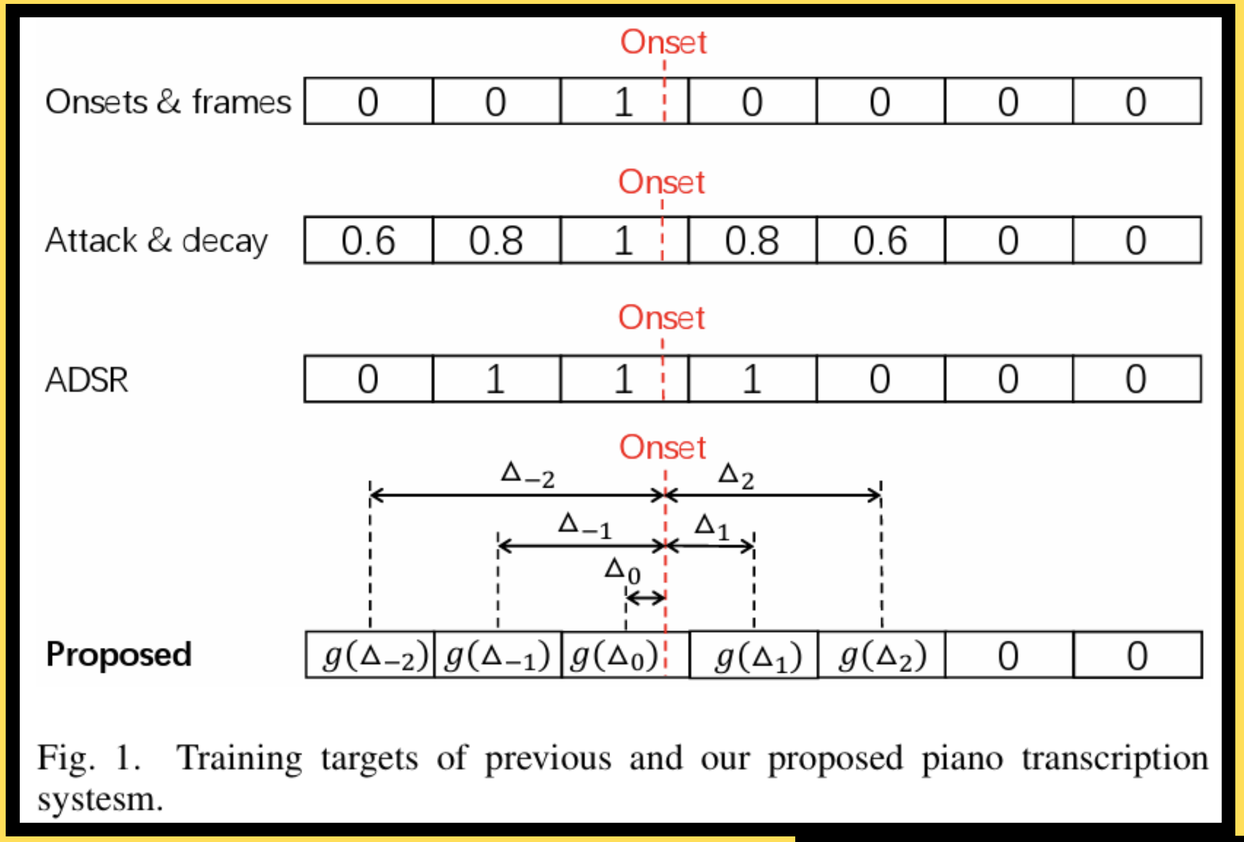


Fig. 1. Training targets of previous and our proposed piano transcription systesm.

REGRESSED ATTACK

$$\begin{cases} g(\Delta_i) = 1 - \frac{|\Delta_i|}{J\Delta}, |i| \leq J \\ g(\Delta_i) = 0; |i| > J, \end{cases}$$

Calculating Exact Onset & Offset Time

Estimating the
actual peak,
when the values at
different frames
are available,
using
triangular geometry

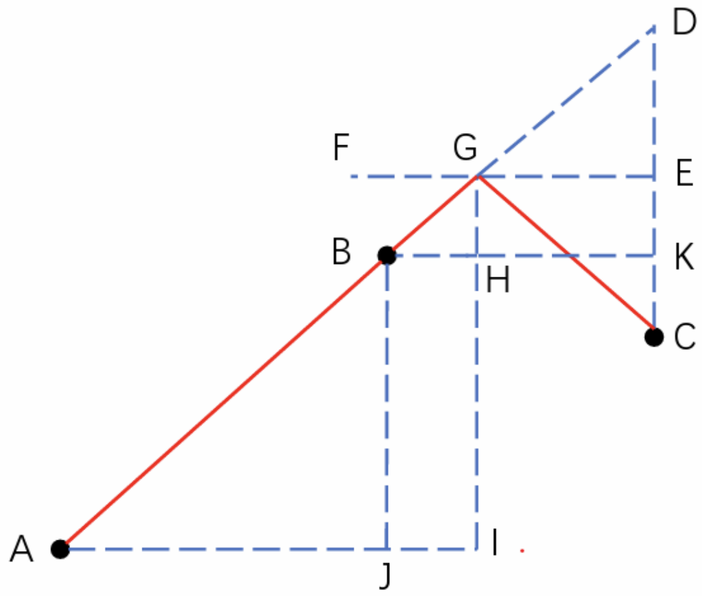
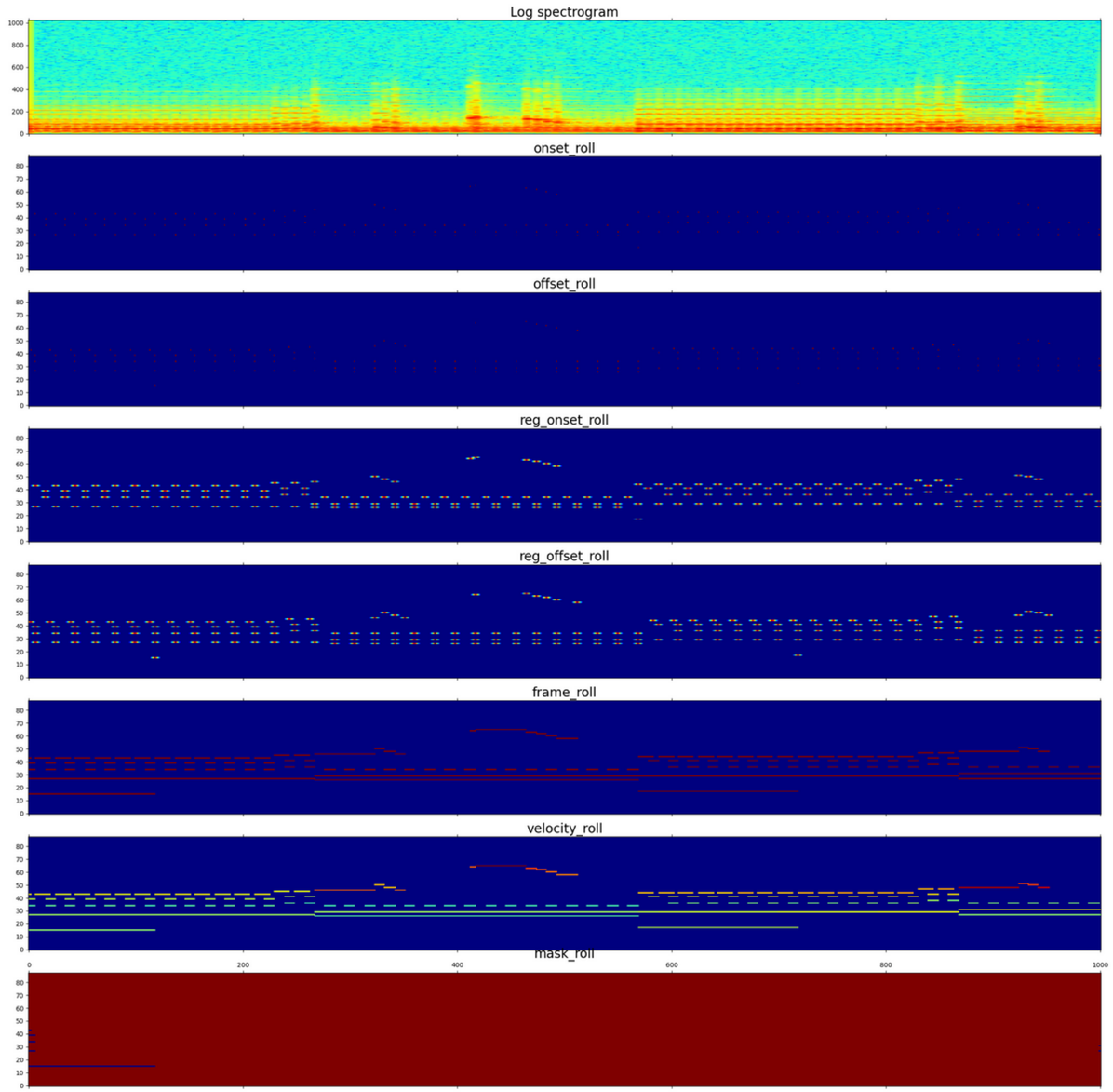


Fig. 4. Demonstration of calculating the precise onset or offset time of a note. The points A, B and C are predicted outputs of three frames. The point G is the calculated precise onset or offset time.

PROCESSED FEATURES

An example visualization of how data after pre-processing looks like:

Log Spectrogram is our input (X)
The other graphs (onset, offset, frame, velocity) are out target. Mask is just a mask used to compute losses



LOSS FUNCTION

The loss function used is a binary cross entropy loss supported by following equations:

Frame Loss:

$$l_{\text{fr}} = \sum_{t=1}^{T} \sum_{k=1}^{K} l_{\text{bce}}(I_{\text{fr}}(t,k), P_{\text{fr}}(t,k)),$$

(Similarly for Regressed onsets and offsets)

Velocity Loss:

$$l_{\text{vel}} = \sum_{t=1}^{T} \sum_{k=1}^{K} I_{\text{on}}(t,k) \cdot l_{\text{bce}}(I_{\text{vel}}(t,k), P_{\text{vel}}(t,k)).$$

Here, T is the number of frames (1001), K is number of piano notes (88).

I is the target frame value while P is the predicted frame value.

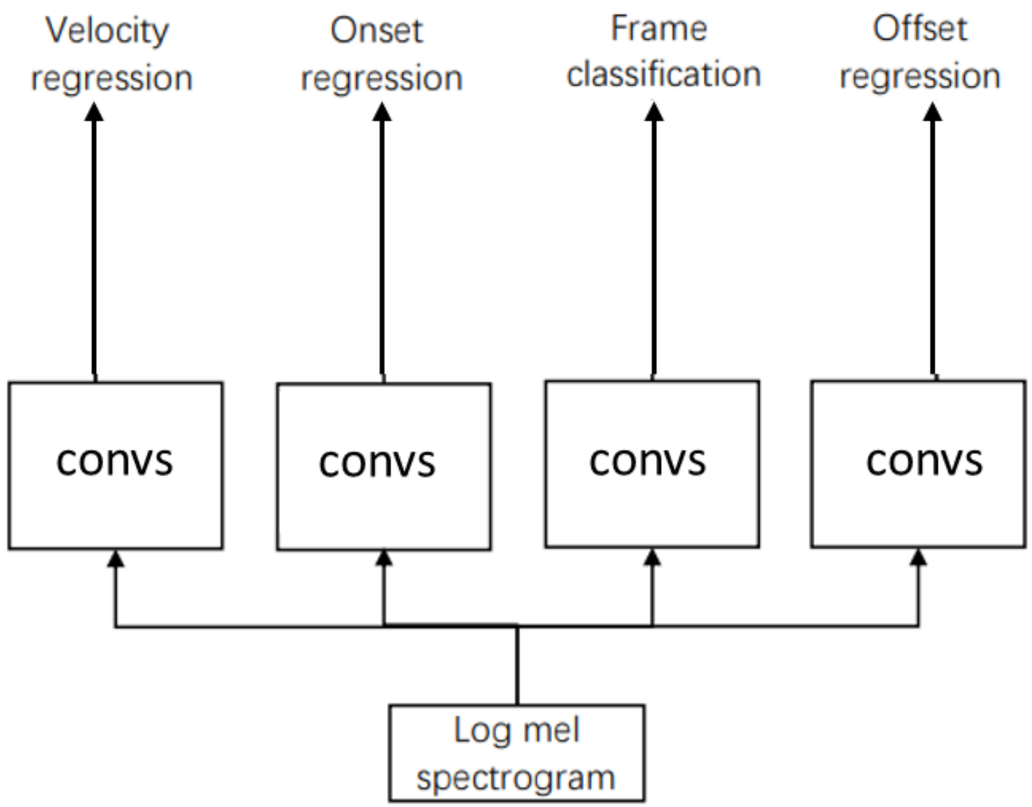
Similar to frame loss, onset regression loss and offset regression loss are calculated.

The velocity loss is masked using onsets because onsets carry rich information about velocities while the decay part carry less information.

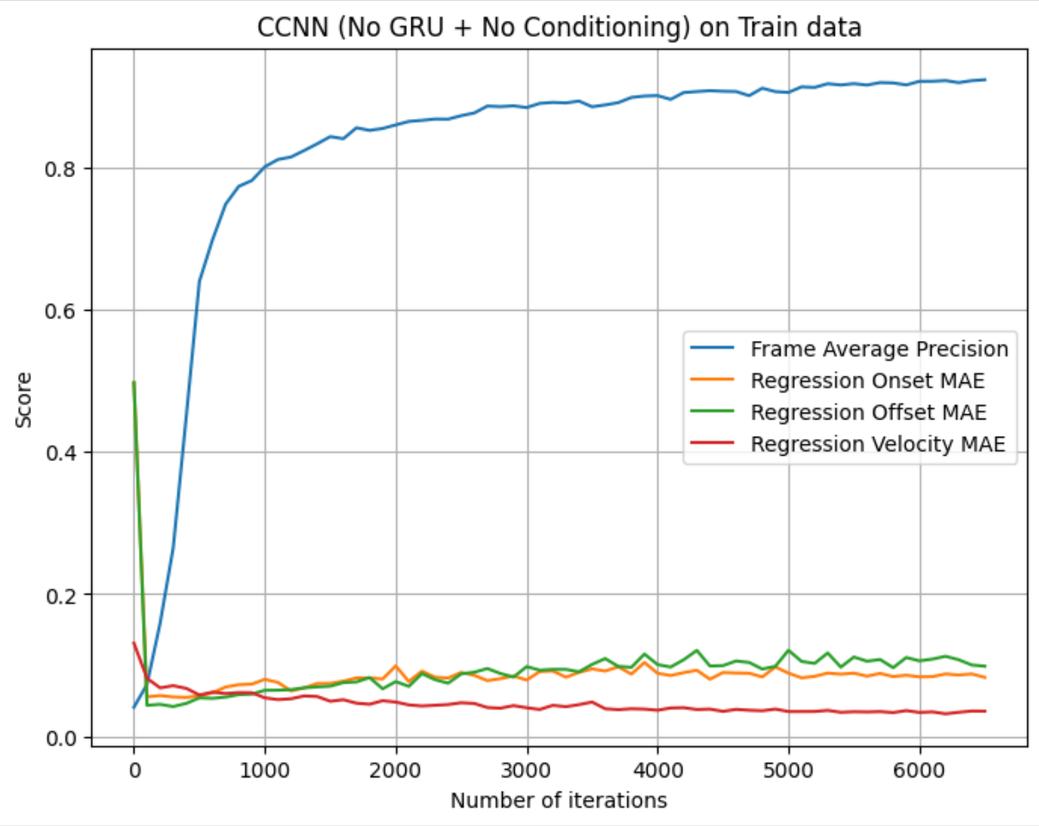
TOTAL LOSS FUNCTION = (frame loss) + (onset loss) + (offset loss) + (velocity loss)

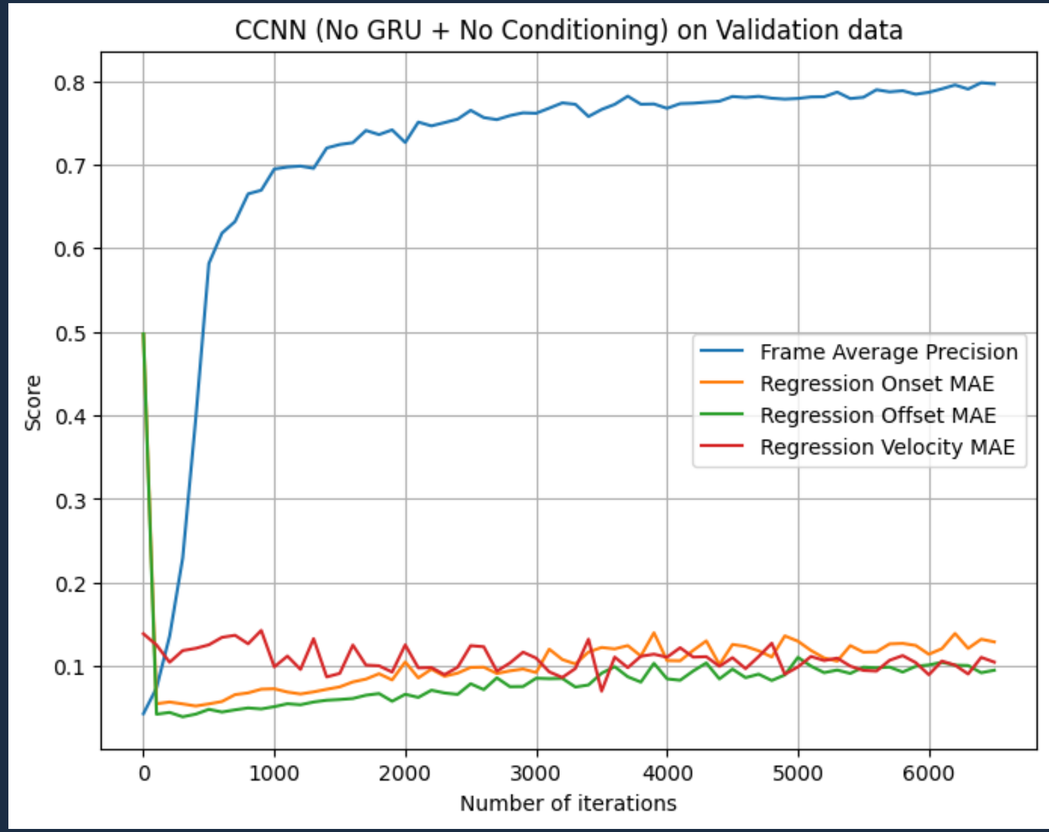
MODEL 1

The input waveform is passed through a log mel spectrogram. Then four concurrent CNN models without any Gated recurrent unit are used. They are run on the input for velocity, onset regression, offset regression, and frame classification respectively.



RESULTS

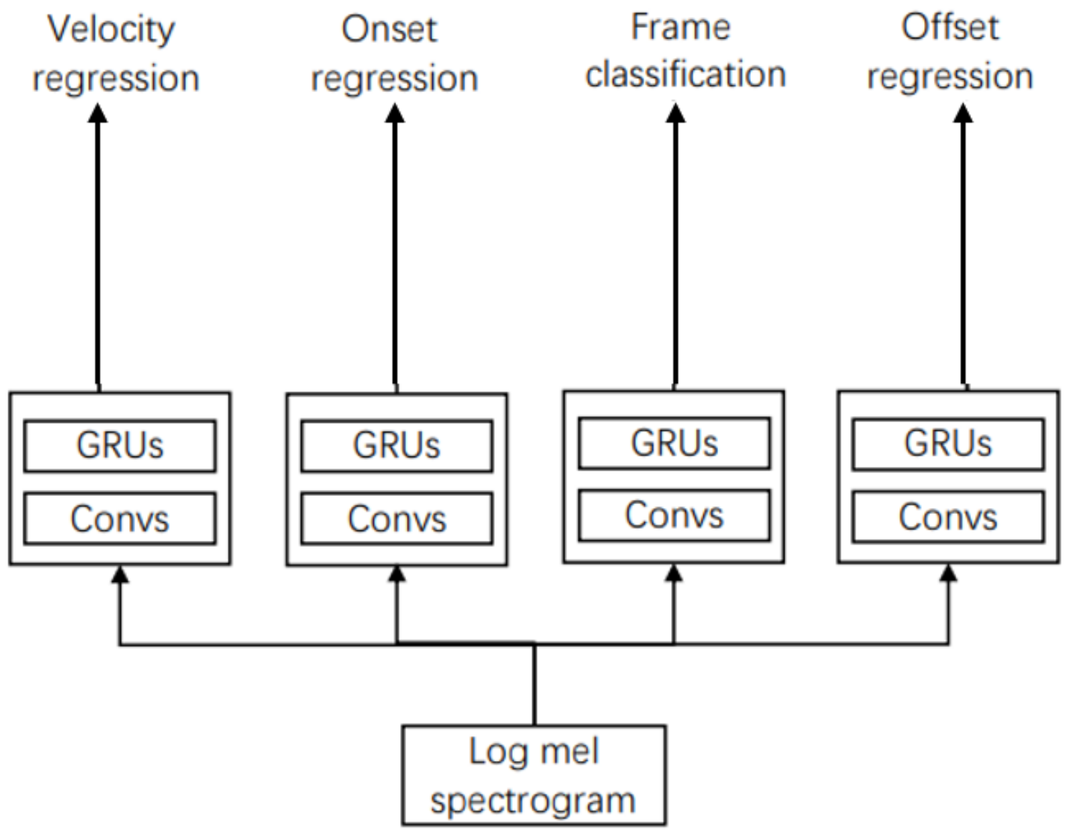

CCNN (No GRU + No Conditioning) on Train data
Score
Number of iterations
Frame Average Precision
Regression Onset MAE
Regression Offset MAE
Regression Velocity MAE
0.0
0.2
0.4
0.6
0.8
0
1000
2000
3000
4000
5000
6000


CCNN (No GRU + No Conditioning) on Validation data
Score
Number of iterations
Frame Average Precision
Regression Onset MAE
Regression Offset MAE
Regression Velocity MAE
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1000
2000
3000
4000
5000
6000

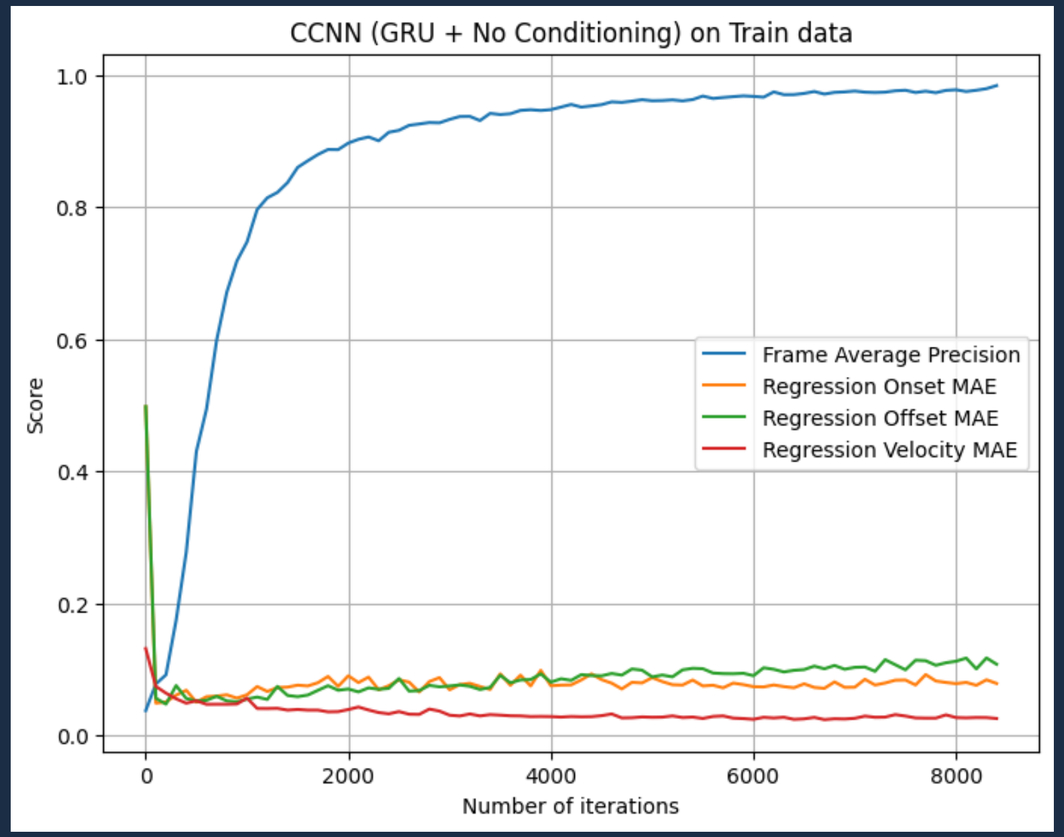
Acoustic model

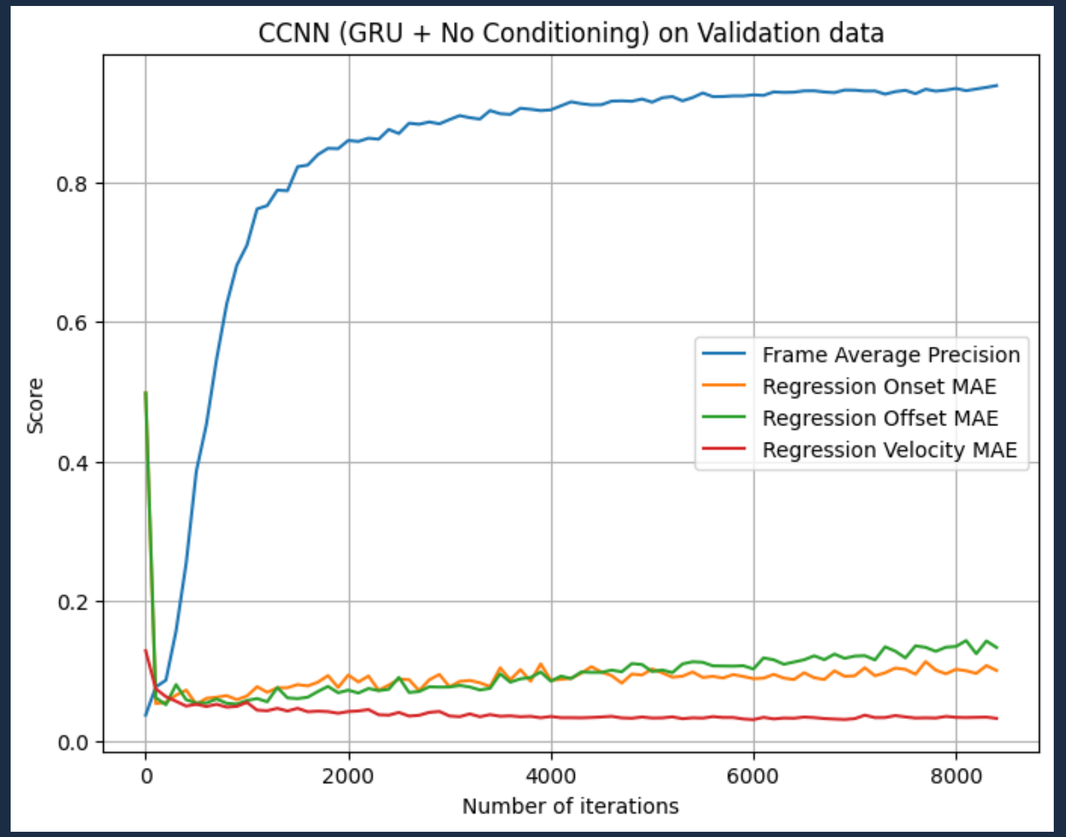
MODEL 2

To improve model 1, after each CNN we are adding a gated recurrent unit. It helps in capturing temporal context. In music temporal context is important to get note duration, rhythmic patterns, note onset offset information more precisely.



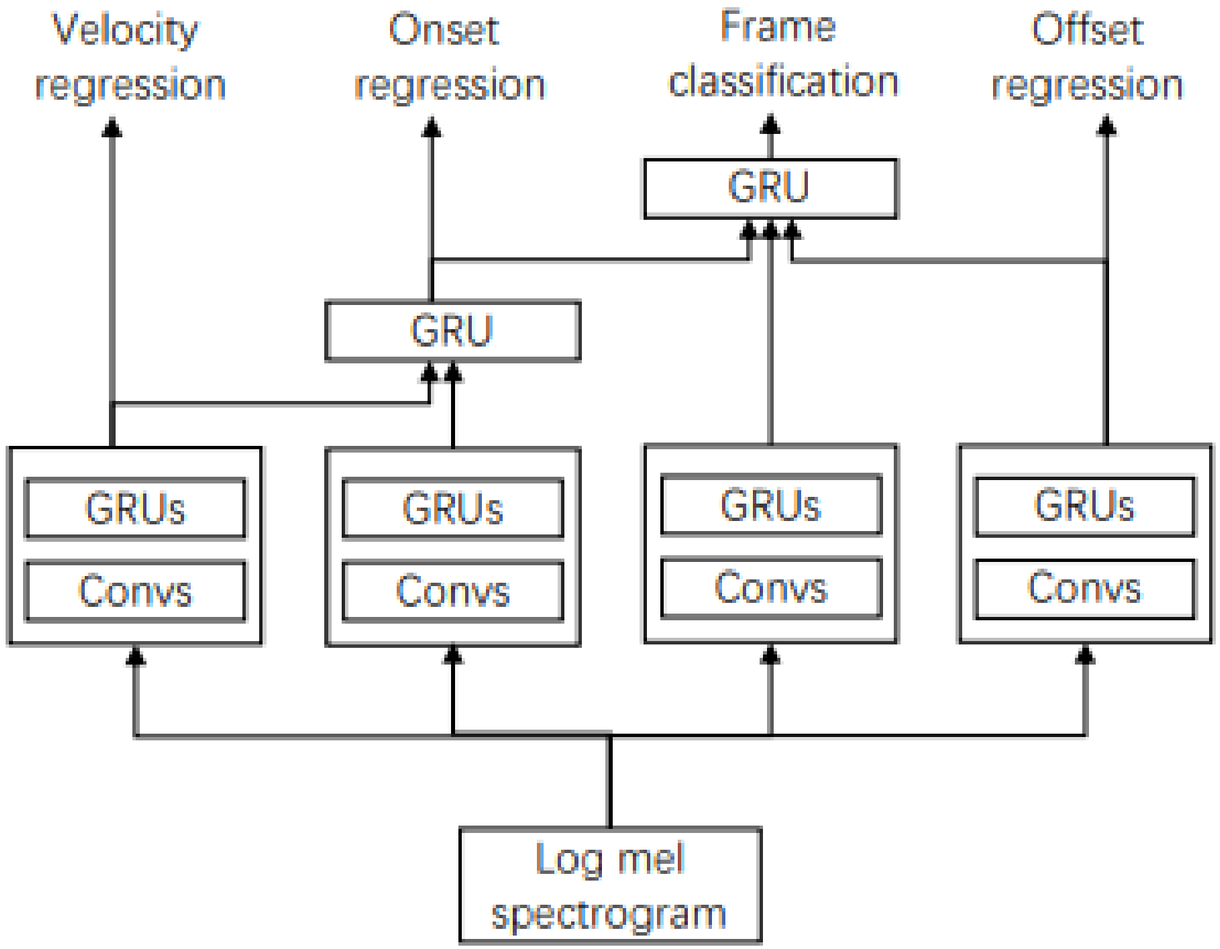
RESULTS


CCNN (GRU + No Conditioning) on Train data
Score
Number of iterations
Frame Average Precision
Regression Onset MAE
Regression Offset MAE
Regression Velocity MAE
0.0
0.2
0.4
0.6
0.8
1.0
0
2000
4000
6000
8000

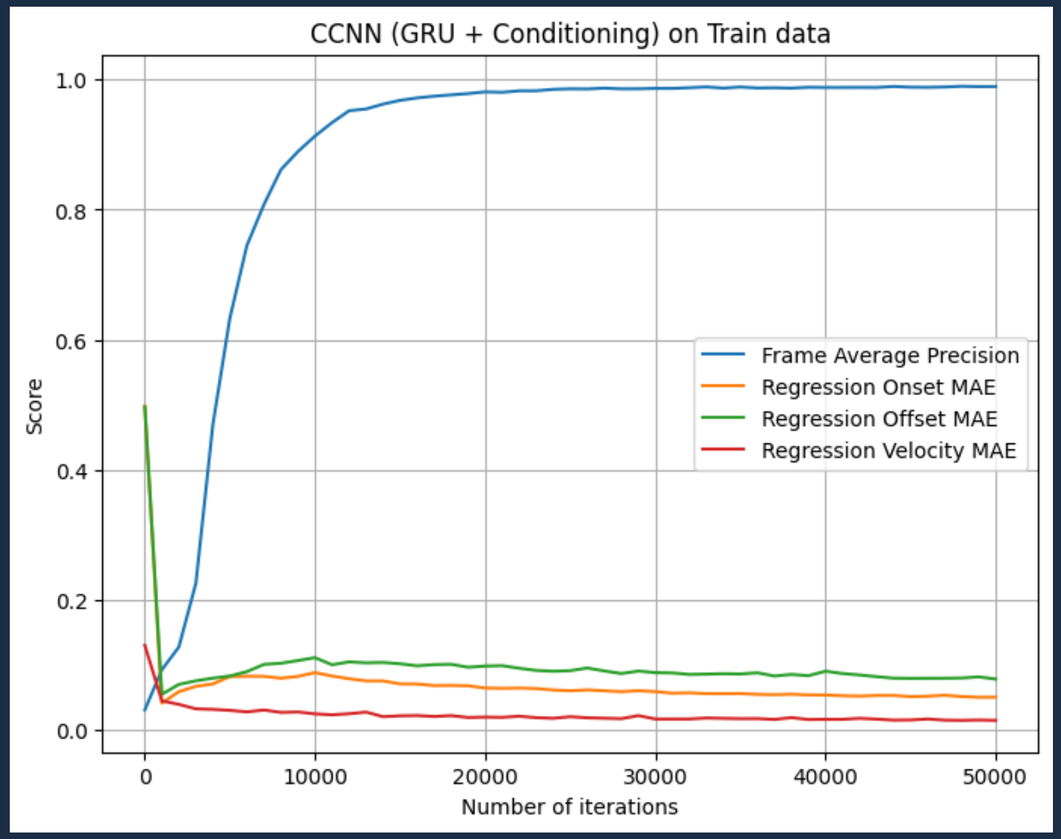

CCNN (GRU + No Conditioning) on Validation data
Score
Number of iterations
Frame Average Precision
Regression Onset MAE
Regression Offset MAE
Regression Velocity MAE
0.0
0.2
0.4
0.6
0.8
0
2000
4000
6000
8000

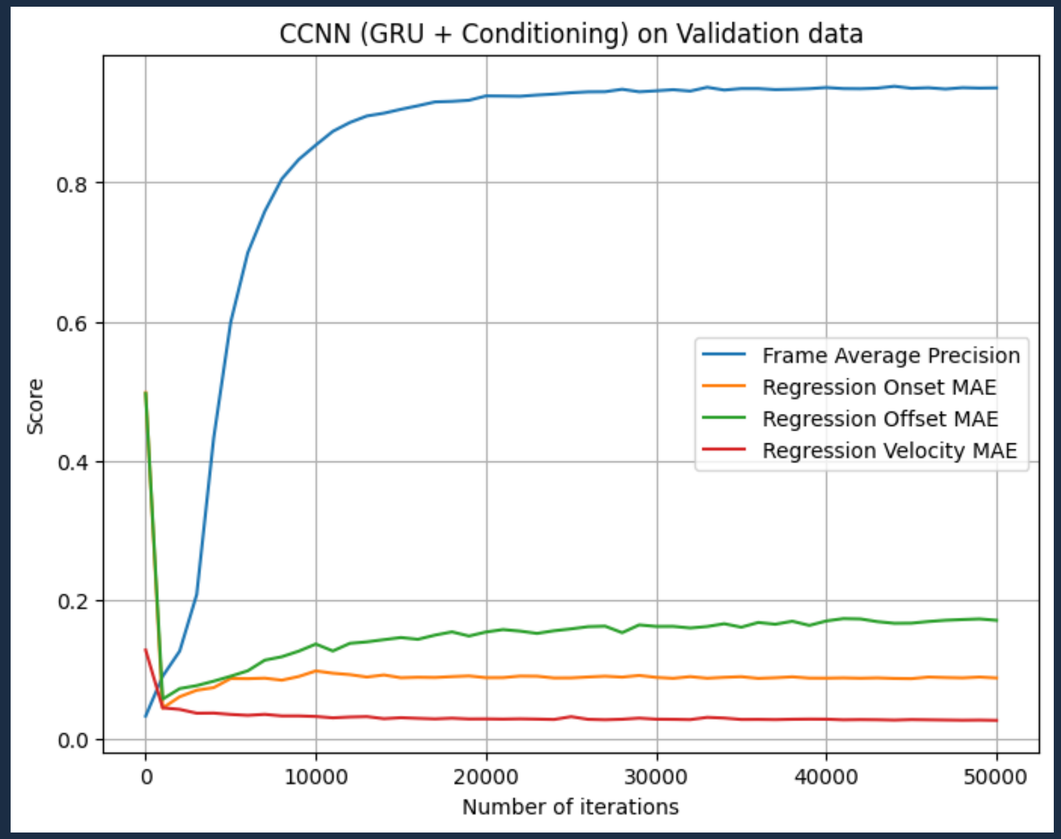
OVERALL MODEL “CCNN”

According to paper, velocity regression output is used to condition the onset regression output. This is because the detection of onsets and velocities can affect each other. Similarly, according to paper, onset regression, offset regression and frame classification to get final frame classification predictions.

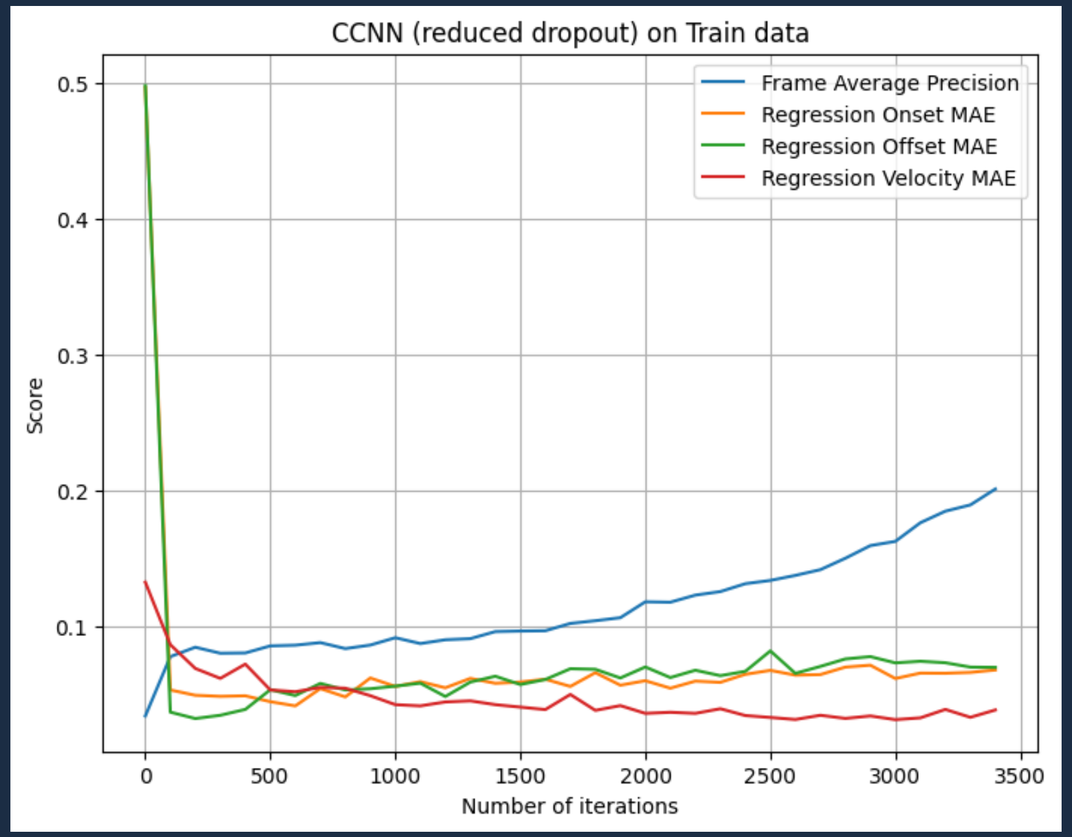


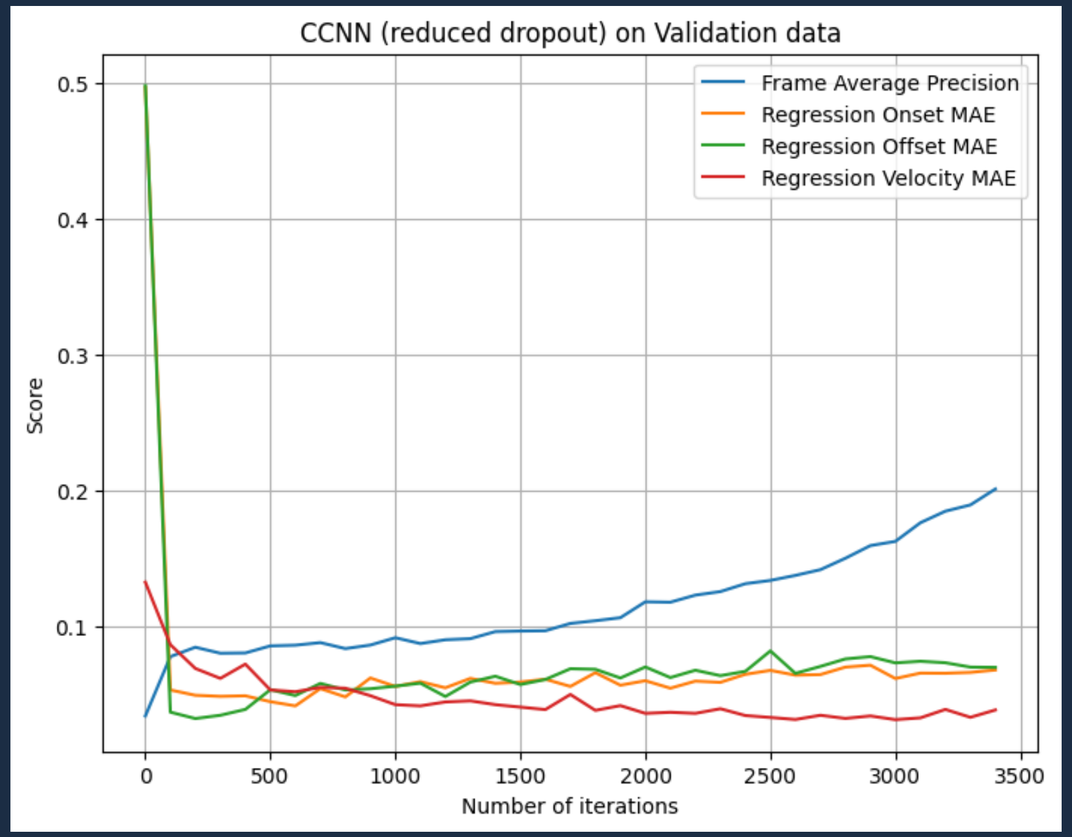
RESULTS


CCNN (GRU + Conditioning) on Train data
Score
Number of iterations
Frame Average Precision
Regression Onset MAE
Regression Offset MAE
Regression Velocity MAE
0.0
0.2
0.4
0.6
0.8
1.0
0
10000
20000
30000
40000
50000


CCNN (GRU + Conditioning) on Validation data
Score
Number of iterations
Frame Average Precision
Regression Onset MAE
Regression Offset MAE
Regression Velocity MAE
0.0
0.2
0.4
0.6
0.8
0
10000
20000
30000
40000
50000

RESULTS


CCNN (reduced dropout) on Train data
Frame Average Precision
Regression Onset MAE
Regression Offset MAE
Regression Velocity MAE
Score
Number of iterations
0.5
0.4
0.3
0.2
0.1
0
500
1000
1500
2000
2500
3000
3500


CCNN (reduced dropout) on Validation data
Frame Average Precision
Regression Onset MAE
Regression Offset MAE
Regression Velocity MAE
Score
Number of iterations
0.5
0.4
0.3
0.2
0.1
0
500
1000
1500
2000
2500
3000
3500

INFERENCE

- In inference, we input the log mel spectrogram of an audio recording to our trained piano transcription to calculate the frame-wise prediction, onset regression, offset regression and velocity regression outputs. Then, we propose an algorithm to process those outputs to high-resolution note events, where the note events can be represented by quadruples of (note, note_on/off, channel, velocity).
- First, we detect the onset regression outputs with local maximum values.
- If a local maximum is larger than an onset threshold, then we say there exist an onset or offset near the frame.
- Next, we analytically calculate the precise onset or offset times of the piano note, using the past, and future frames to constitute a triplet of frames

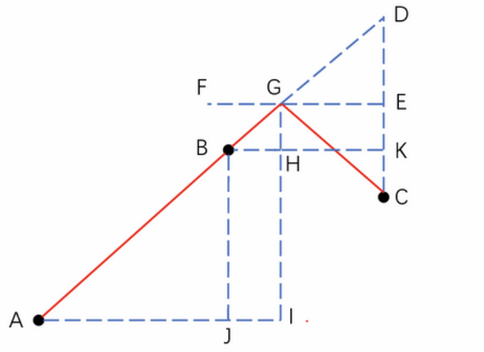


Fig. 4. Demonstration of calculating the precise onset or offset time of a note. The points A, B and C are predicted outputs of three frames. The point G is the calculated precise onset or offset time.

INFERENCE

- By this means, we are able to calculate the precise onsets and offsets times of piano notes.
- For each detected onset, an offset is detected if the offset regression is over an offset threshold, or the frame prediction is lower than a frame threshold.
- The velocities of onsets are obtained by scaling the velocity regression back to a range of [0, 128)

TRAINING

Use it to transcribe a given file

Need apt post-processing after it

Store checkpoints of trained model

Returns output dictionary of frame, offset, onset, velocity

calculate average precision of frames during validation

MODEL

Calculate LOSS and Backprop

x

y

store into h5py files

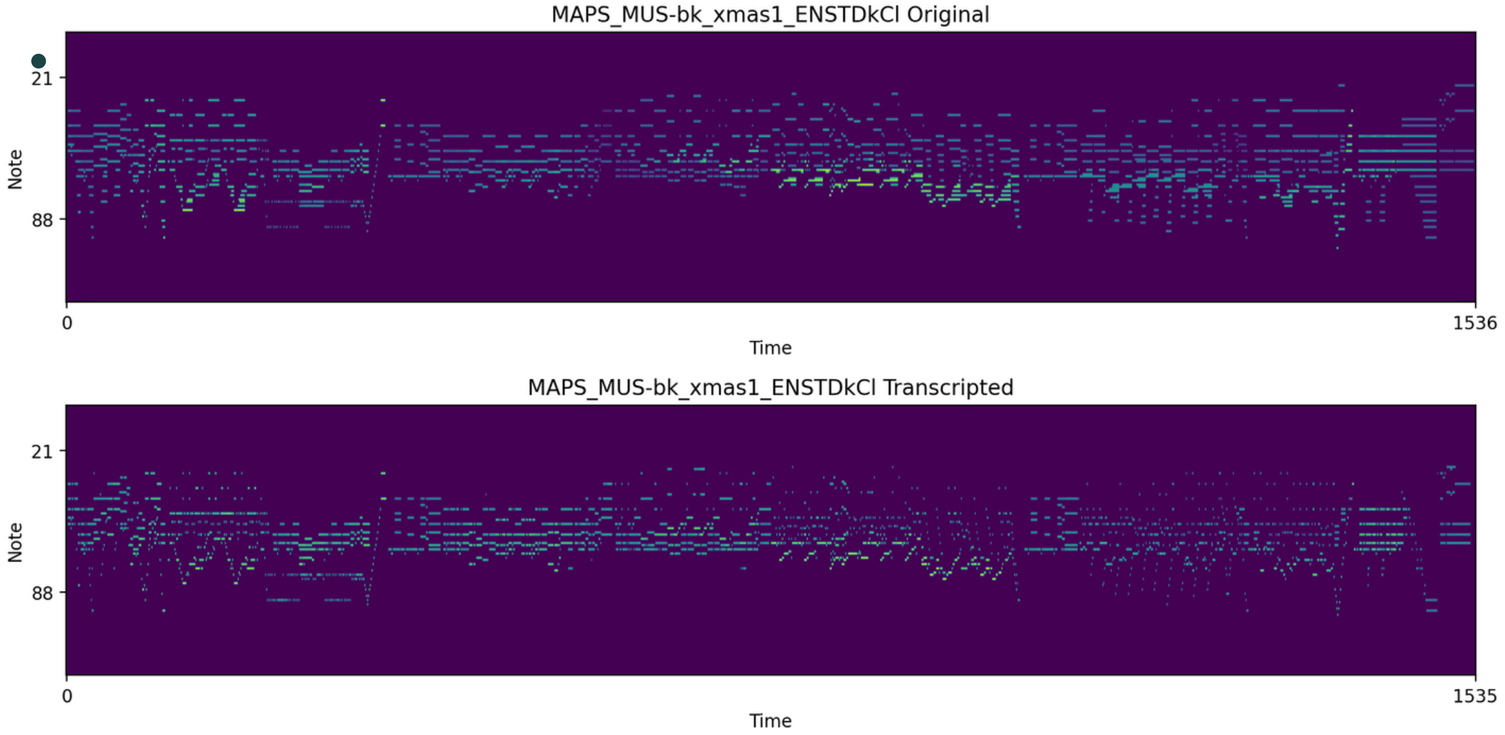
Create segments each of 10 seconds for each audio

Create dictionary of frame waveform, onset, offset, velocity, and provide waveform to model

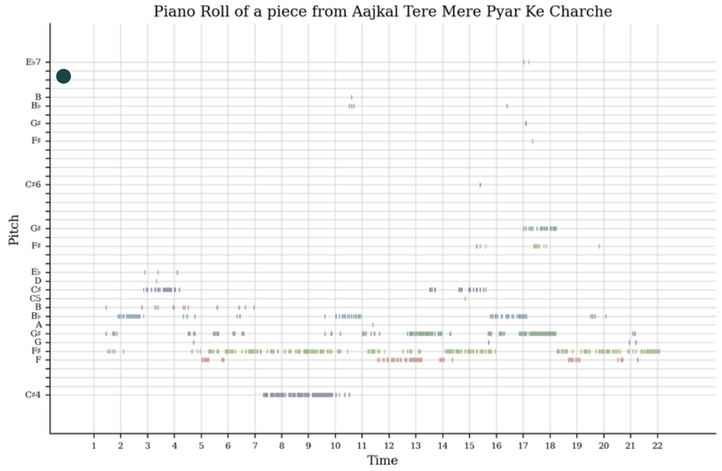
Audio Waveform

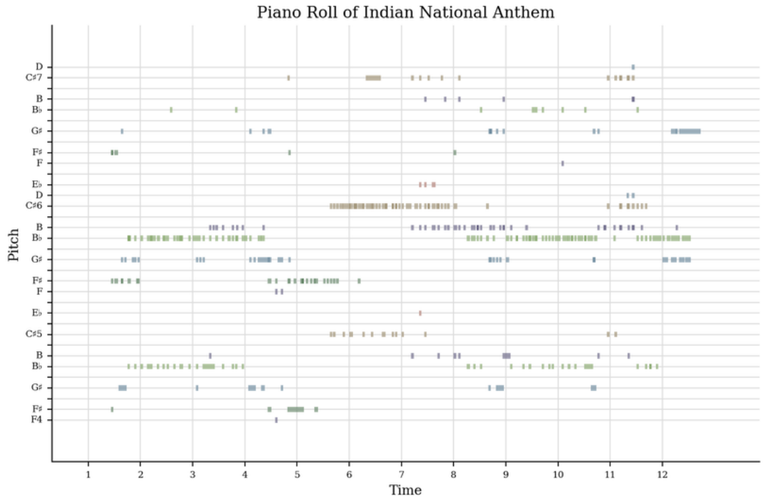
Corresponding MIDI file

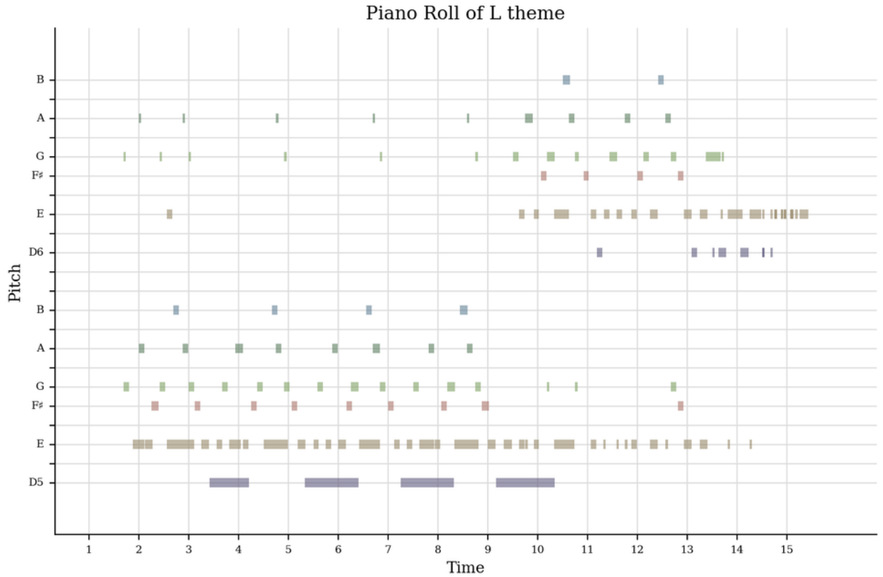
SAMPLE TRANSCRIPTION



TRANSCRIPTION ON OTHER AUDIOS

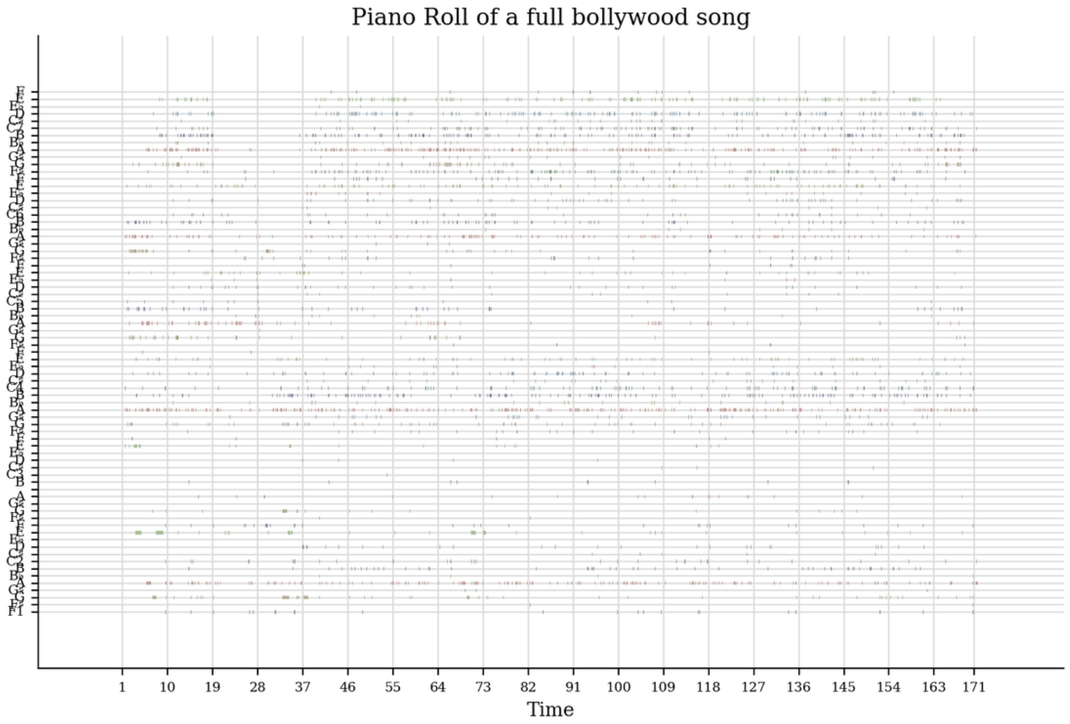






[Click for Audio/MIDI files]

LIMITATIONS



- The model performed well enough for transcripting monophonic audios with single instruments
- If the instrument is piano (or something from which it can interpret the spectrogram), then transcription is better, otherwise noise gets introduced
- In a real setting song, it can transcript a basic cover, but a lot of noise is there due to polyphonic audios present, as well as vocals

WORK DISTRIBUTION

Atishay : Collecting data and paper, reading data + splits, dataloaders, feature extraction

Avadhoot : Models + Post-processing

Megh : Models + Post-processing

Harshit : Loss function

Veeresh : Pre-processing of data

Overall everyone contributed to different parts of code + model implementations, processing data, evaluating and collecting results

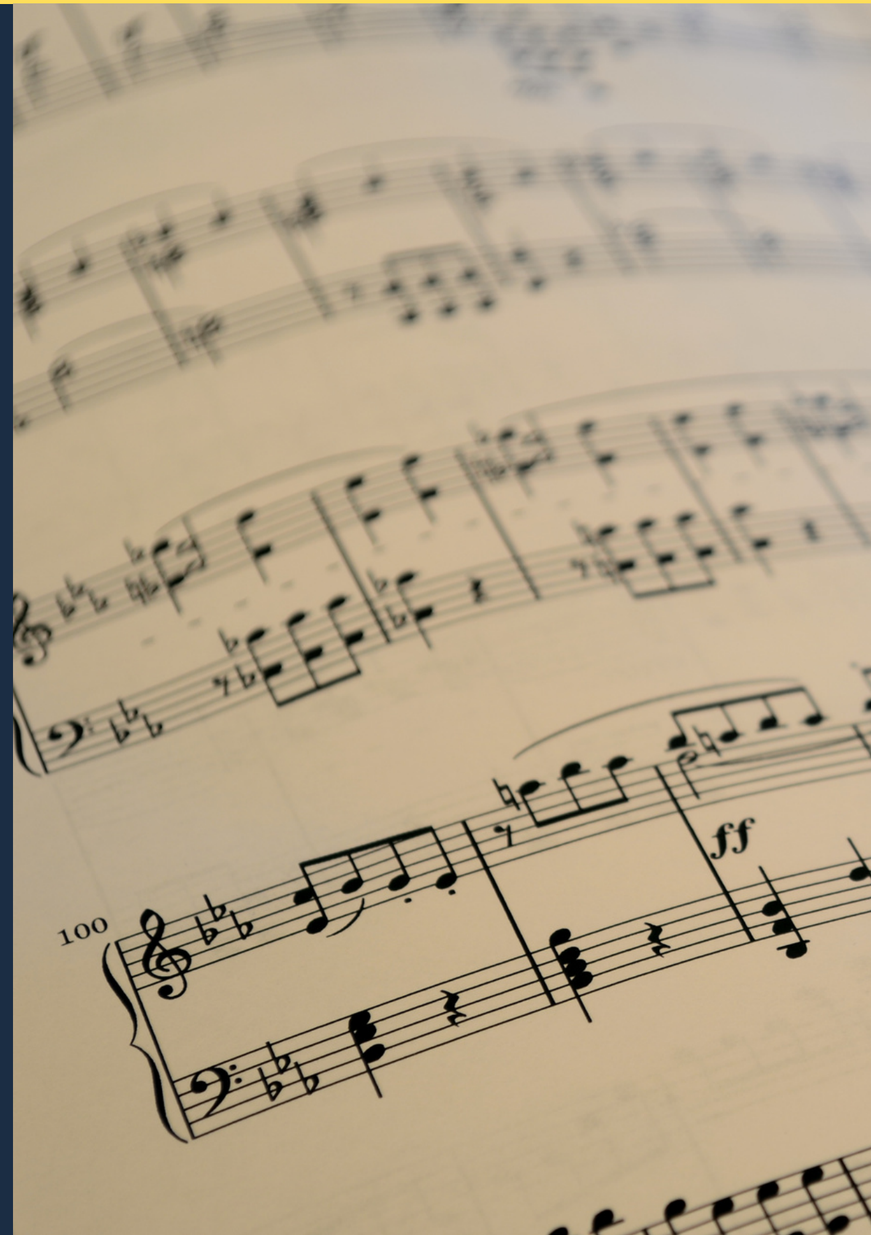
PROJECT AHEAD

- We can train the model on a different dataset such as MusicNet to see its results when it is trained with different instruments, that is, for Polyphonic Transcription
- This would require some more amount of preprocessing which could enable model to understand instruments better
- Our model has been trained maximum for 50000 iterations, we can increase that or use early stopping when necessary
- Augmentation can be used for using MAPS or MusicNet datasets, to accomodate for noise getting introduced

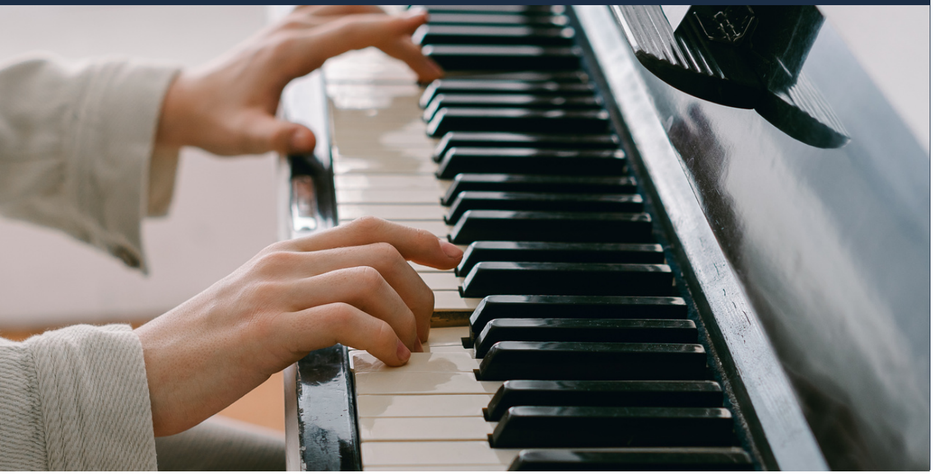
---

- The original paper was also about Pedal transcription, and combined both ideas, this can be implemented to improve performance

---



REFERENCES



1

High-resolution Piano Transcription with Pedals by Regressing Onset and Offset Times

Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, Yuxuan Wang

***Abstract*—Automatic music transcription (AMT) is the task of transcribing audio recordings into symbolic representations. Recently, neural network-based methods have been applied to AMT, and have achieved state-of-the-art results. However, many previous systems only detect the onset and offset of notes frame-wise, so the transcription resolution is limited to the frame hop size. There is a lack of research on using different strategies to encode onset and offset targets for training. In addition, previous AMT systems are sensitive to the misaligned onset and offset labels of audio recordings. Furthermore, there are limited researches on sustain pedal transcription on large-scale datasets. In this article, we propose a high-resolution AMT system trained by regressing precise onset and offset times of piano notes. At inference, we propose an algorithm to analytically calculate the**

discriminative models, such as support vector machines to predict the presence or absence of notes in audio frames [11]. To address the multiple pitch estimation problem, a probabilistic spectral smoothness principle was proposed in [12] for piano transcription. A combination of frequency domain and time domain method was proposed for piano transcription in [13], where authors assumed that signals are a linearly weighted sum of waveforms in a database of individual piano notes. Non-negative matrix factorizations (NMFs) and non-negative sparse codings [14] have been proposed to decompose spectrogram into polyphonic notes [15], where signals are decomposed into the multiplication of

Paper on HIgh Resolution Piano Transcription by Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, Yuxuan Wang (Main reference)

https://inria.hal.science/inria-00544155/en - for MAPS dataset and its related informations

ENABLING FACTORIZED PIANO MUSIC MODELING AND GENERATION WITH THE MAESTRO DATASET paper - for reviewing comparison between datasets

https://ieeexplore.ieee.org/abstract/document/8682605 - referred for reviewing other processing techniques and applying them on MAPS/MusicNet

Tech Stack - Pytorch, torchlibrosa, h5py, librosa, audioread, soundfile, mido (audio processing), Matplotlib, numpy, scikitlearn

Arigato