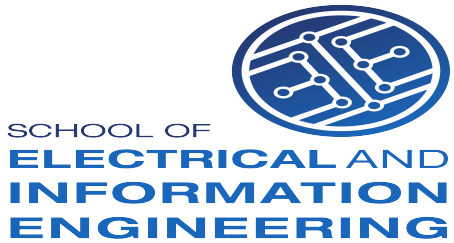
Automated Music Transcriber for two Musical Instruments

Sbonelo Mdluli and Moshekwa Malatji
Supervisor: Prof. Olutayo O Oyerinde



07 October 2020

Overview

Introduction

Automatic Music Transcription(AMT) is defined as the design of computational algorithms to convert acoustic music signals into of music notation, This is a process that is of concern to digital signal processing and artificial intelligence methodologies.
Typical AMT several sub-tasks and applications include (multi-)pitch estimation, onset and offset detection, instrument classification, music practice using computer accompaniment [1].

Figure 1: link to github repo

Automatic Music Transcription Framework

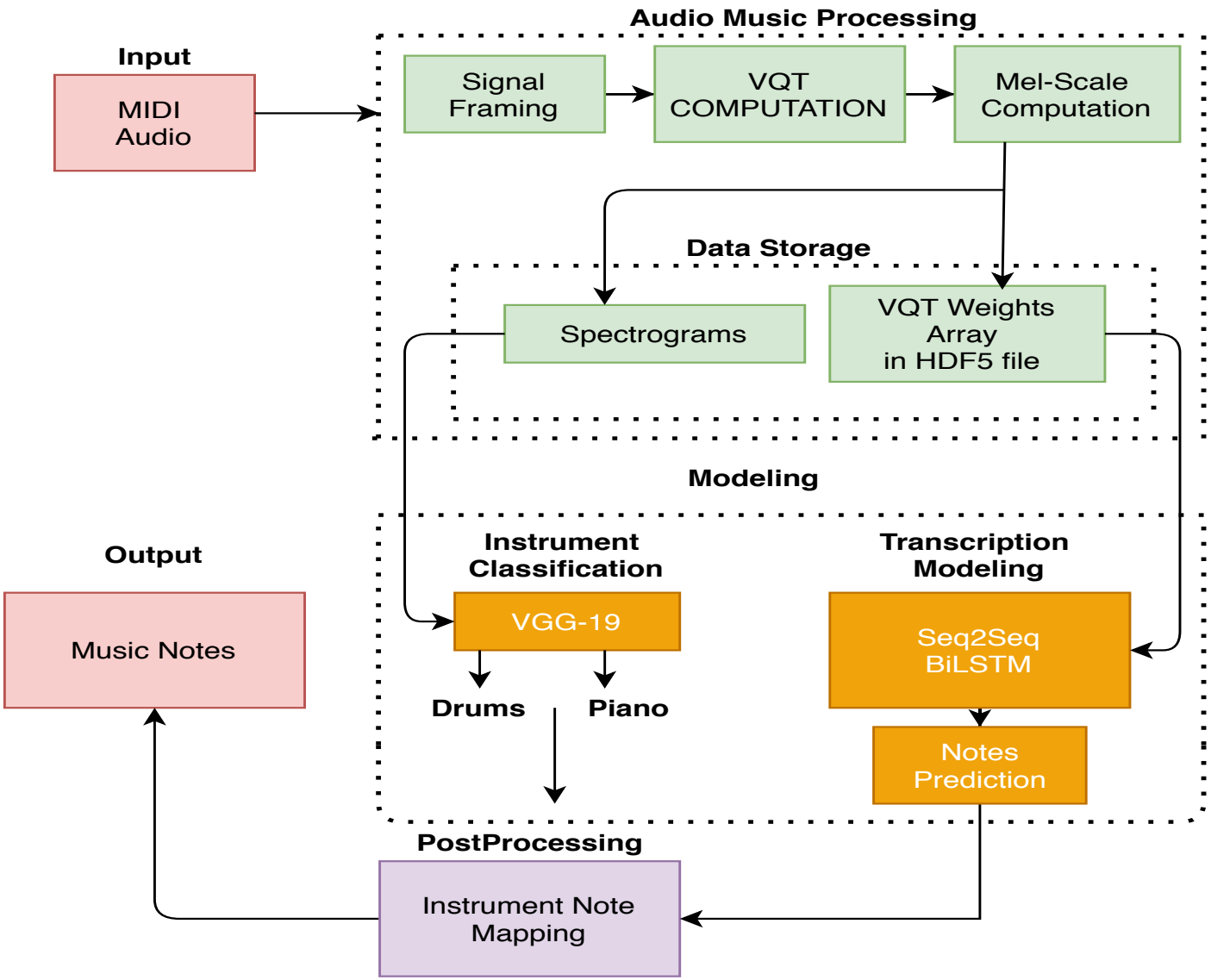


Figure 2: Transcription model high level overview

Variable Q-Transform

For AMT the aim of the digital signal processing of audio signals is to provide a spectral representation of the signals in frequency domain in the form of spectrograms. Moreover, we compute the Variable Q-Transform of the signal with the following procedure:
1. Split the signal into 625 windows with 7 frames per window
2. Compute the VQT of each window and apply Hann window function defined by Eqn 2

The k th spectral component of VQT for a signal $x[n]$ can be obtained using the 1.

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k,n]x[n]e^{\frac{-j2\pi Qn}{N[k]}} \tag{1}$$

$$W[k,n] = sin^2(\frac{\pi n}{N[k]}) \tag{2}$$

Mel-Spectrogram

Transformation of the linear VQT frequency scale results into a Mel-Scale which is non-linear transformation of the frequency scale.

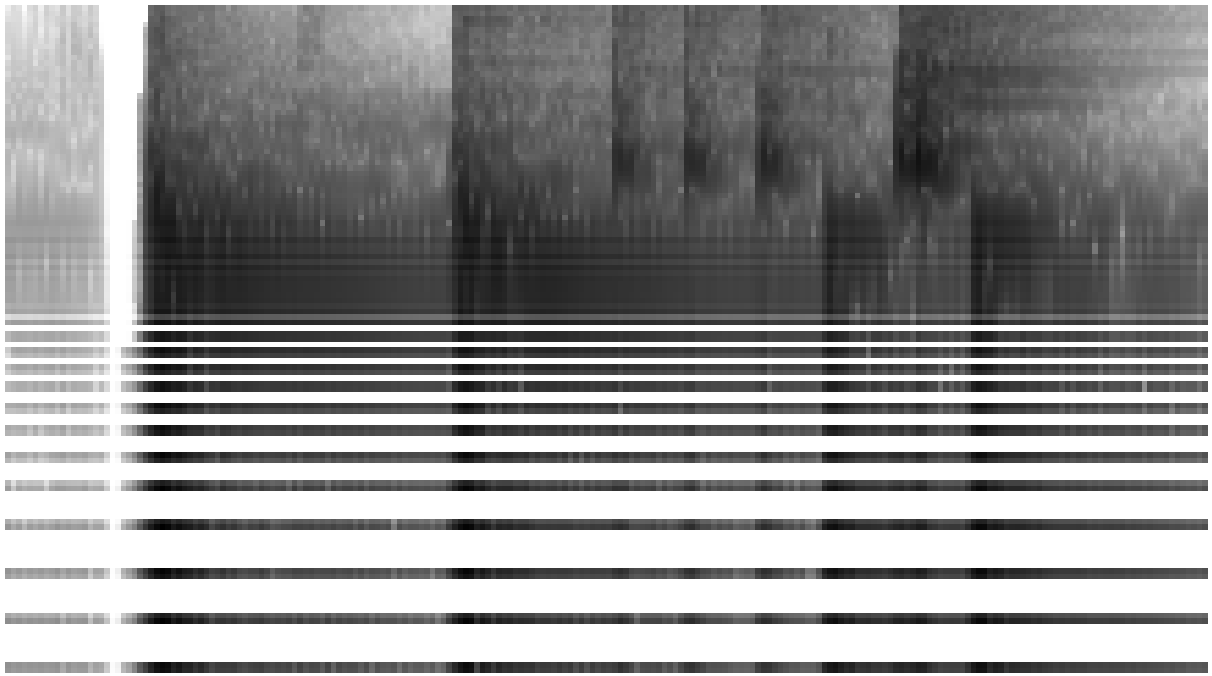


Figure 3: Mel-Scaled Spectrogram

Instrument classification

Get instrument type based on spectrogram. VGG16 = 94% VGG19 = 97%. VGG19 has 19 layers trained on imagenet data set [2]. The model is trained using a batch size of 10 and 10 epochs using stochastic gradient descent with lr=0.001 and momentum=0.9.

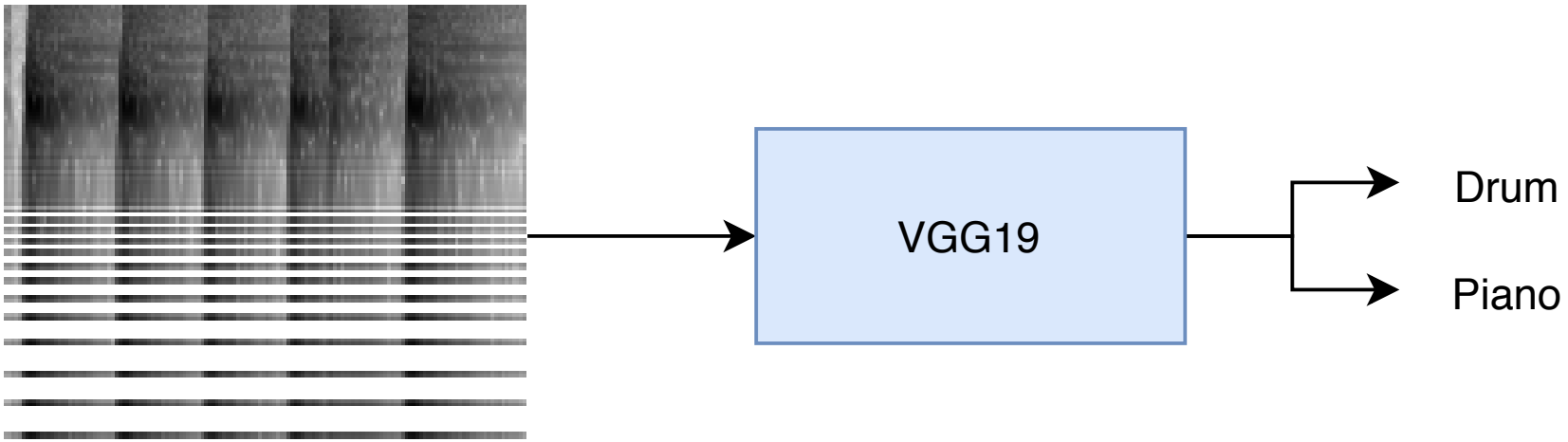


Figure 4: Instrument classification high level

Cont..

Binary cross entropy loss function. Where y_n is either 0 (drum) or 1 (piano) depending on the instrument class.

$$L = \frac{1}{N}\sum_{n=0}^{N} y_n log(\hat{y_n}) + (1 - y_n) log(1 - \hat{y_n}) \tag{3}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

$1 > \hat{y_n} \geq 0.5$ indicates a piano and $0 \leq \hat{y_n} < 0.5$ for drums.

Note prediction

One hot encoded vector represent active notes for a given time frame. We use vectors of length 88, with 1 being active note and 0 inactive note. The notes for a piano $= \{0, 1, 2, ..., 87\}$ and drum $= \{35, 36, 37, ..., 80\}$, drum $\subset$ piano as such 88 indices are sufficient for both instruments.

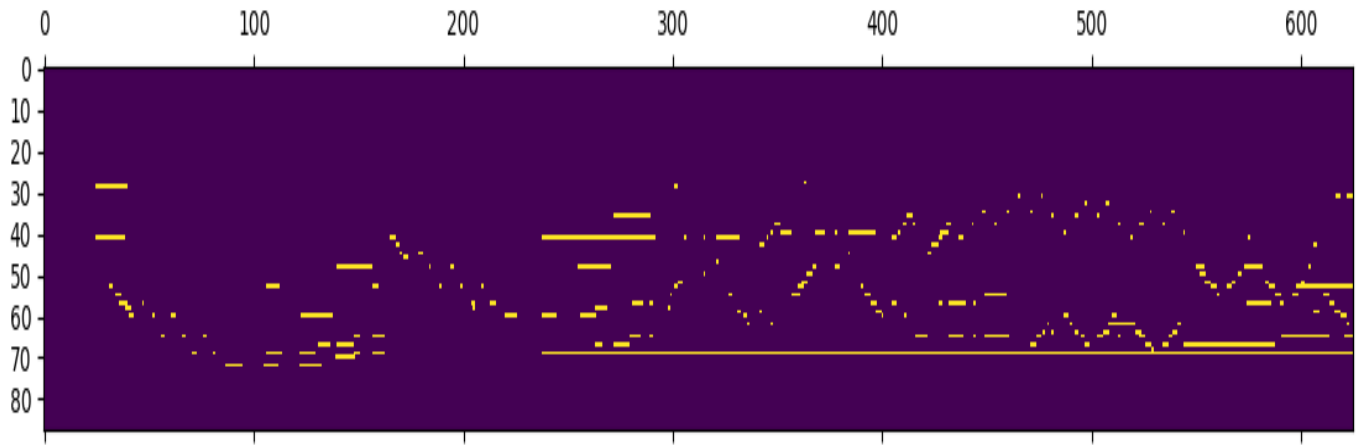


Figure 5: One hot encoded vector for a piano for 20 seconds

Transcription

We use a Seq2seq model using a BiLSTM to model a many-to-many relationship between frequencies (X) in a given window and present musical notes (Y) in that same window. Given an input sequence $X = (x_0, x_1..., x_i)$ we want to predict output sequence $Y = (x_0, x_1..., x_j)$ [3].

$$P_\theta(Y|X) = \prod_{j=1}^{J+1} P_\theta(y_j|Y_{<j}, X) \tag{5}$$

We use a model with 200 encoder cells and 100 decoder cells, with 88 predictions in a given time step using a binary cross entropy loss function. The elements of the one hot encoding vectors are defined as $0.5 < \hat{y_n}, \hat{y_n} = 0$ and $\hat{y_n} > 0.5, \hat{y_n} = 1$.

Post processing

Indices of the hot encoded vector represent the MIDI note numbers. We define $f : Y \rightarrow N$ that maps MIDI note number to musical notes. The mapping function is selected based on the instrument class, determined during the classification stage. The model achieves an overall f1 score of 51%.

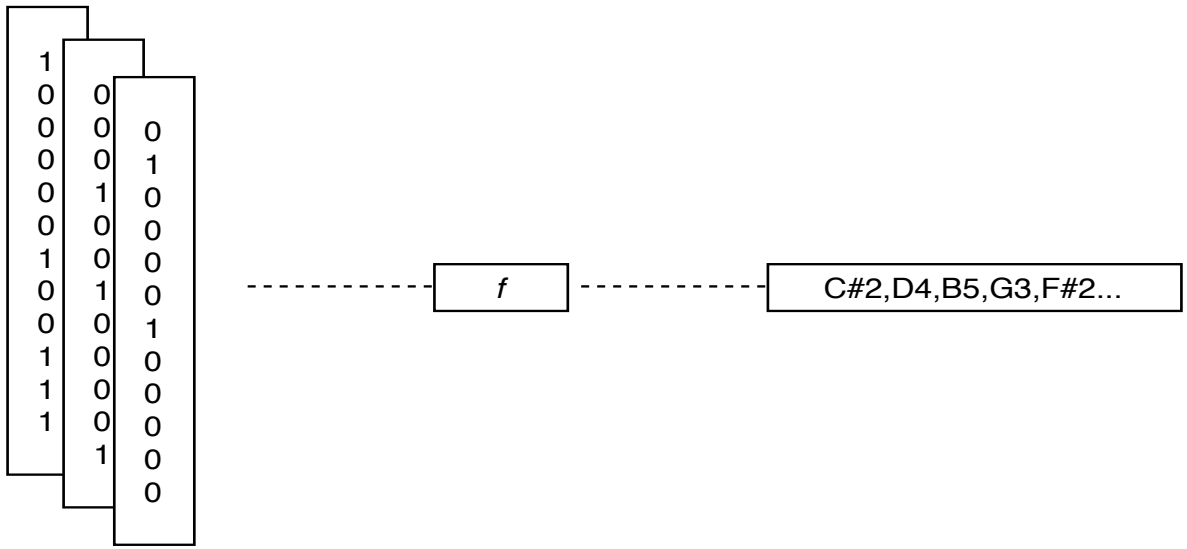


Figure 6: Mapping from hot encoded vectors to musical notes

User Interface

The application platform is developed using PyQt5 and supports Windows and Linux OS.

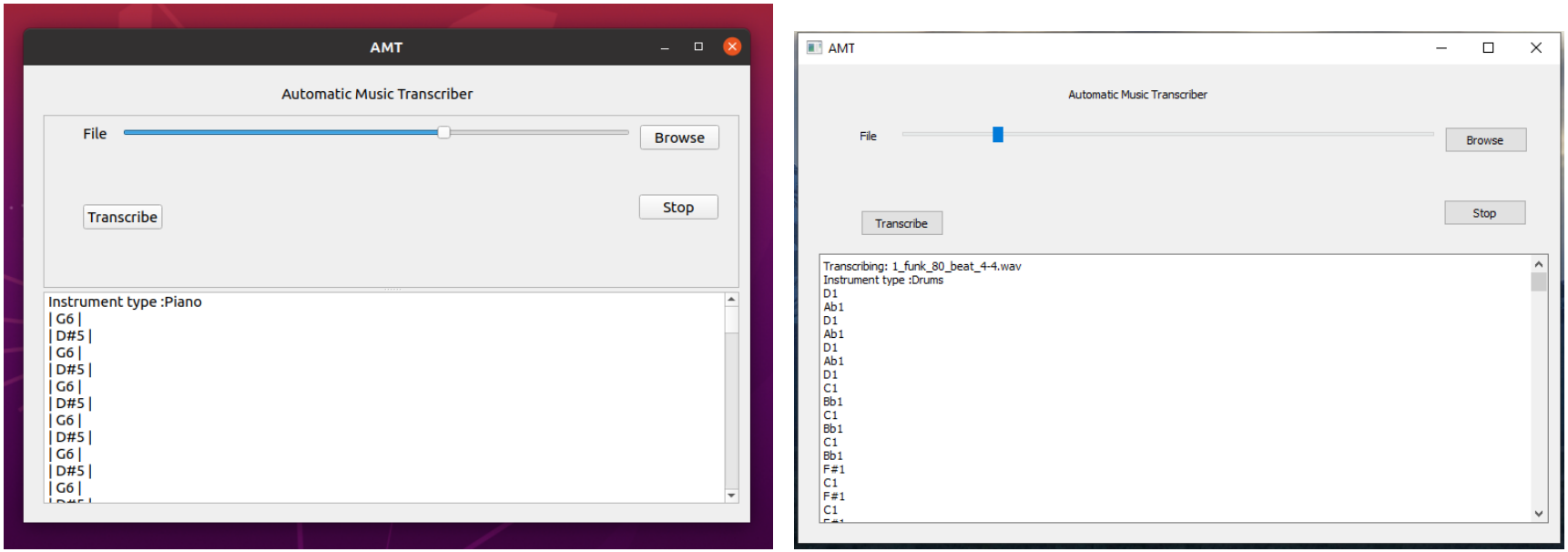
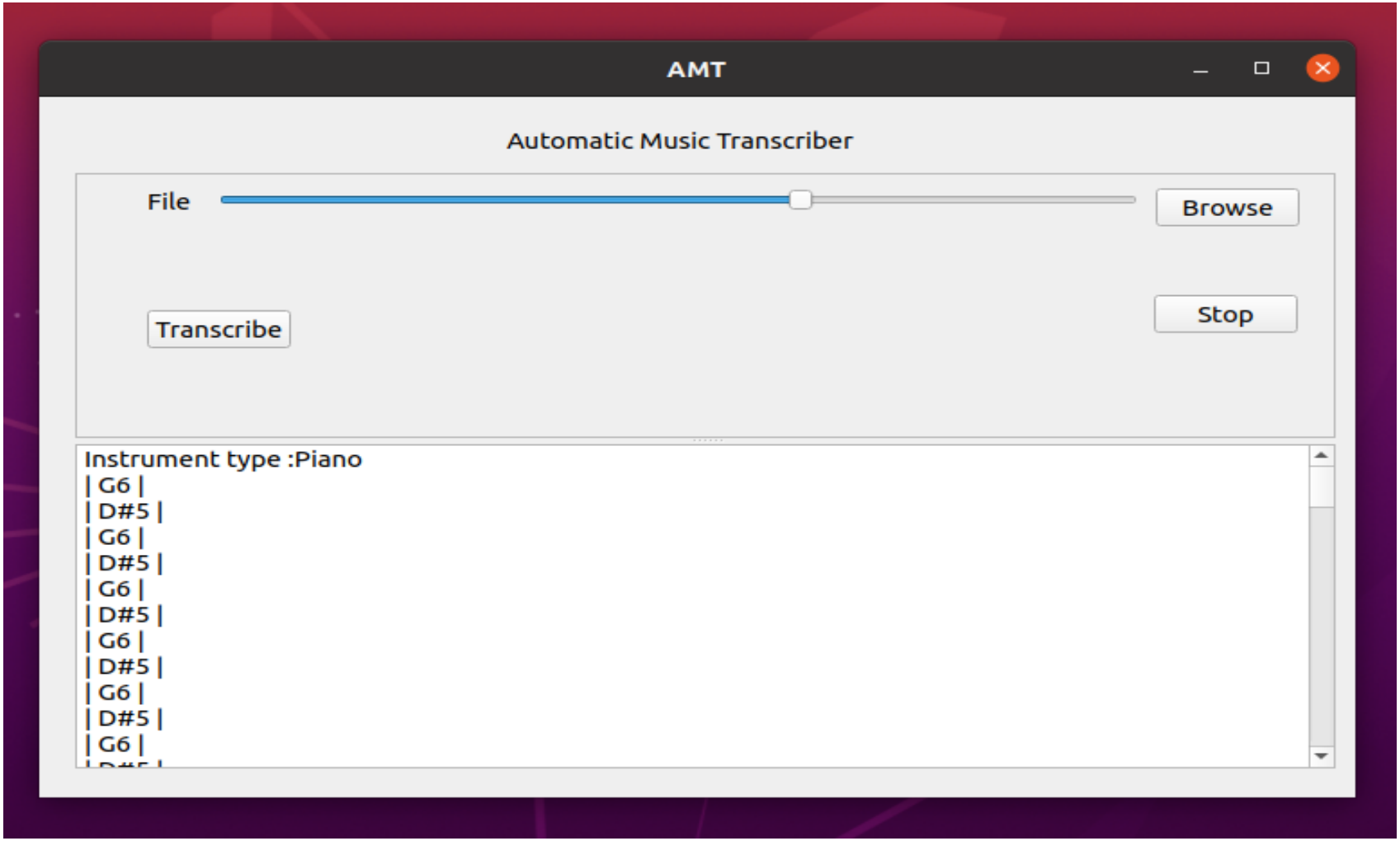


Figure 7: UI in both Ubuntu and Windows os

Prototype Demonstration



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, Jan. 2019. DOI: 10.1109/MSP.2018.2869928.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.