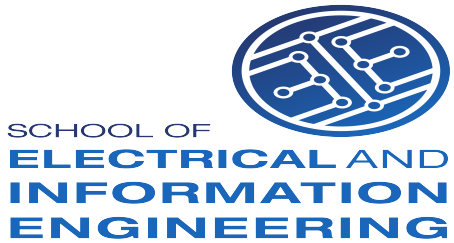
Automated Music Transcriber for two Musical Instruments

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



20 October 2020

Overview

Introduction

AMT Specifications

Dataset

Digital Signals Processing

Modeling

Results

User Interface

Project Demonstration

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

Specifications

Requirements:

- The AMT is required to transcribe two musical instruments viz. Piano and Drums.

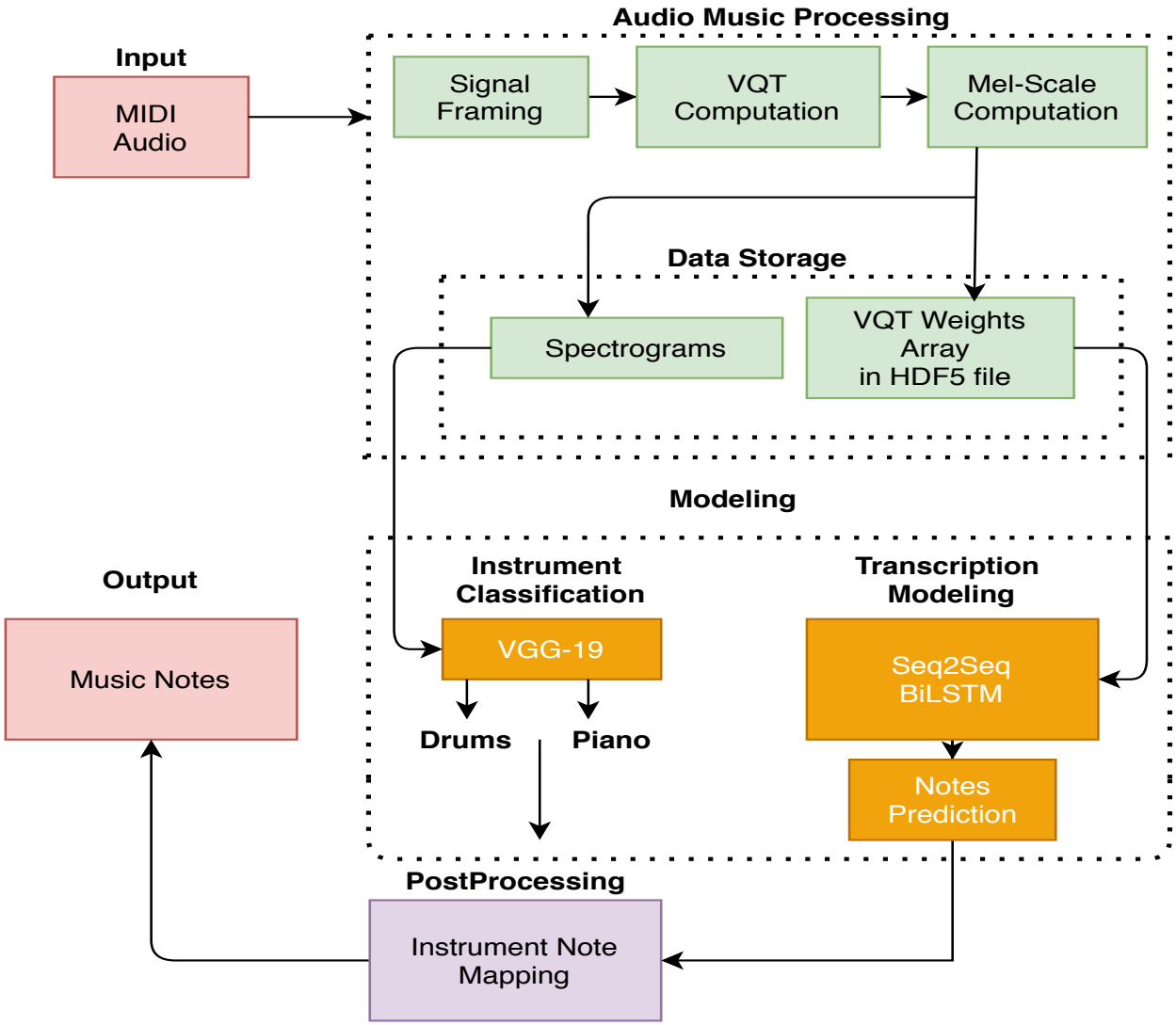
Assumptions:

- We assume that the signal source only contains monophonic music and the music to be transcribed contains no vocals over them only instruments.

Success Criteria:

- The deep learning model employed for the transcription process should have an accuracy of at-least 50%.
- A convenient Graphical User Interface(GUI) for the AMT software should be developed. The GUI should illustrate the instrument classified and the musical notation that is the output of the AMT.

Automatic Music Transcription Framework



Dataset

The proposed instruments to be transcribed are the piano and drums.

- The Drum dataset is composed of 1,100 MIDI files and over 22,000 measures of drumming including 13.6 hours of MIDI and audio human-performed drumming.
- The Piano dataset contains over 200 hours of paired audio and recorded MIDI data from performances.

Variable Q-Transform(VQT)

The VQT is defined as a filter-banks with centre frequencies(f_k) that the center frequencies of each filter-bank are geometrically spaced. Moreover, we compute the Variable Q-Transform of the signal with the following procedure:

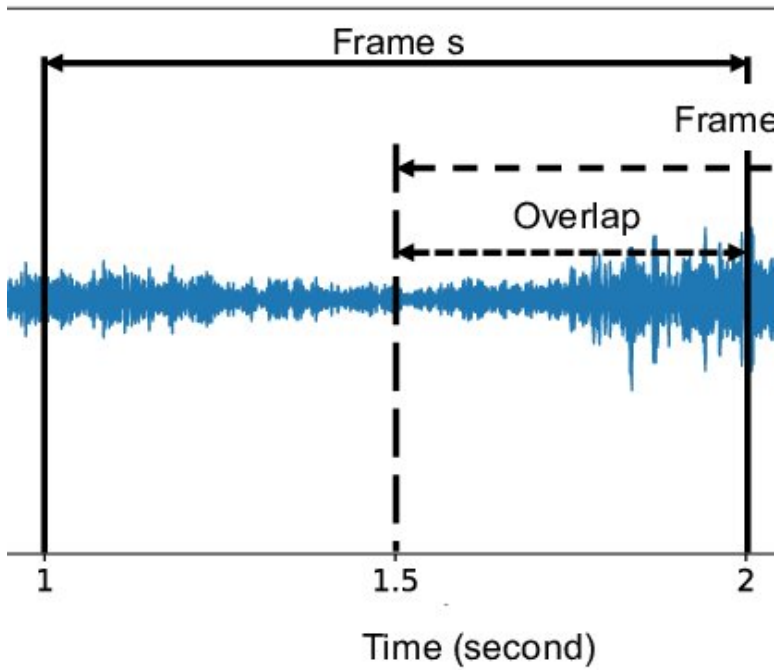


Figure 2: Signal Framing Procedure

Variable Q-Transform(VQT) (cont.)

1. Split the signal into 625 windows with 7 frames per window
2. The chosen window size is 7 and zero padding is applied to every frame by a length of 50% of the window size. Consequently, zero-padding ensures that the samples of all frames are equal. 3. There is an overlap between the windows to ensure that more data is calculated as notes can appear between two frames.

4. The VQT computation employs the Hann Window function at every instance, which is represented by hann below:

$$w[k] = 0.5(1 - cos(\frac{2\pi k}{M_k - 1})) \tag{1}$$

where the frame length, M_k, for every window is defined by Equation2, noting that f_s is the sampling frequency of the VQT.

$$M_k = Q\frac{f_s}{f_k} \tag{2}$$

Variable Q-Transform(VQT) (cont.)

- The VQT has an added parameter, γ, responsible for decreasing and maintaining constant Q-factor for low and high frequencies.

Mel-Spectrogram

The transformation into the Mel-frequency spectrum is defined by mel and it occurs with Mel-frequency bins, $n_{mels} = 128$, with equally spaced frequencies.

$$f_{mel} = 2595 log(1 + \frac{f_{hz}}{700}) \tag{3}$$

Transformation of the linear VQT frequency scale results into a Mel-Scale which is non-linear transformation of the frequency scale using Equation 3.

Input Features

Classification model:

- Spectrogram Image from the resulting VQT computation and Mel-Scale Frequency Transformation.

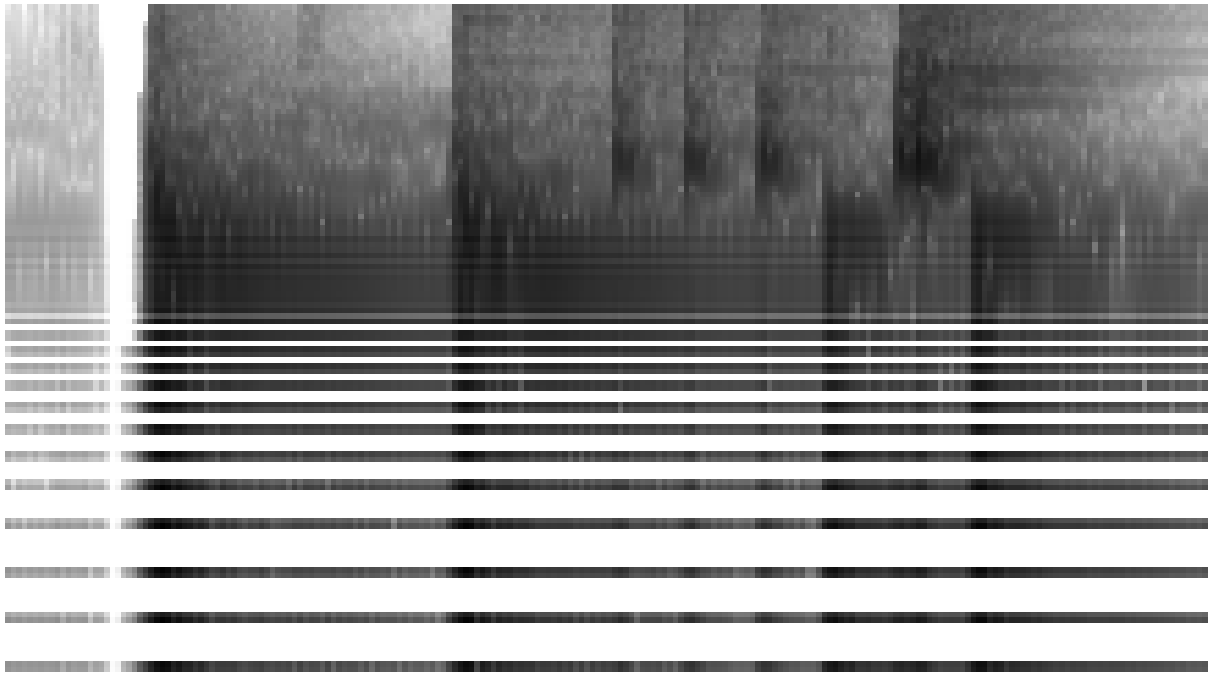


Figure 3: Input Spectrogram

Input Features (cont.)

Transcription model:

- The result from the VQT computation is complex matrix stored in a HDF5 file in preparation for modeling
- Dimensions (625,128,7): 625 windows, 7 frames per window for 128 Mel-Scale filter banks

Instrument classification

Aim: Get instrument type based on spectrogram. Unlike traditional images classification, music is not, symmetrical it is a one-way process which means modifying the spectrogram results in different audio properties. This simplifies the operation performed by the

- CNN since the model does not scale or rotate the spectrograms.
- VGG16 = 94% VGG19 = 97%
- VGG19 has 19 very deep layers pre-trained on image-net data set, and it performs well on small datasets [2].

The model is trained using a batch size of 10 and 10 epochs using Stochastic Gradient Descent with Learning rate=0.001 and Momentum=0.9. Furthermore, VGG19 has five-max pooling layers(size=22) pooling layers that perform2x2 max pooling with stride 2 without padding.

Instrument classification (cont.)

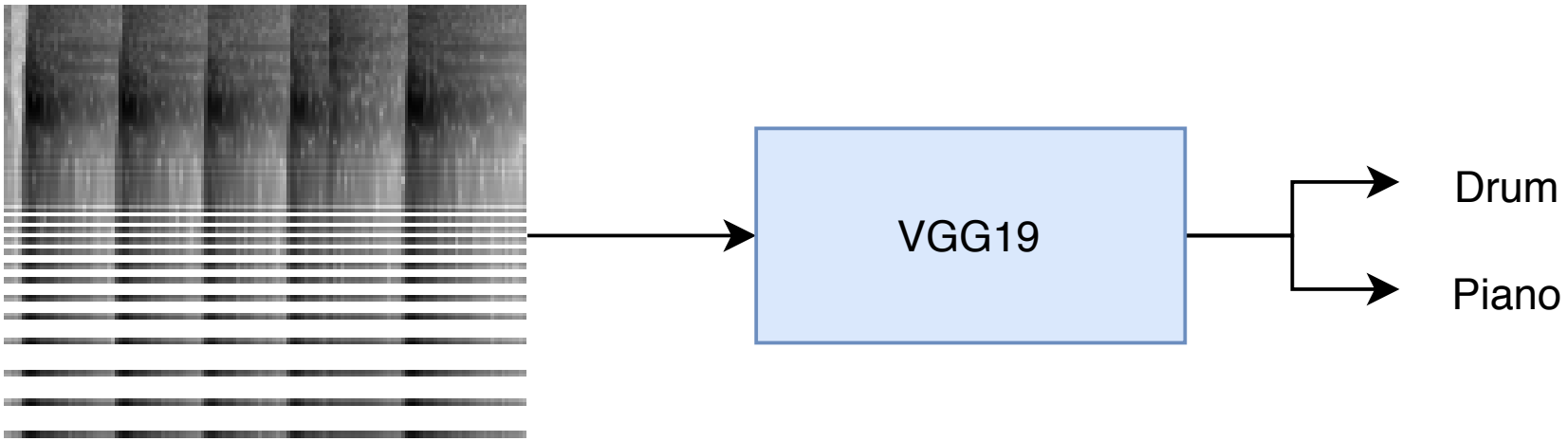


Figure 4: Instrument classification high level

Cont...

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

$$L = \frac{1}{N}\sum_{n=0}^{N} y_n log(\hat{y_n}) + (1 - y_n) log(1 - \hat{y_n}) \tag{4}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

$1 > \hat{y_n} \geq 0.5$ 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.

Each frame takes $20/625 = 0,032$ seconds, it follows that the note duration is $0,032 \times n$ where n is the number of consecutive frames the note spans.

encoded vectors in time

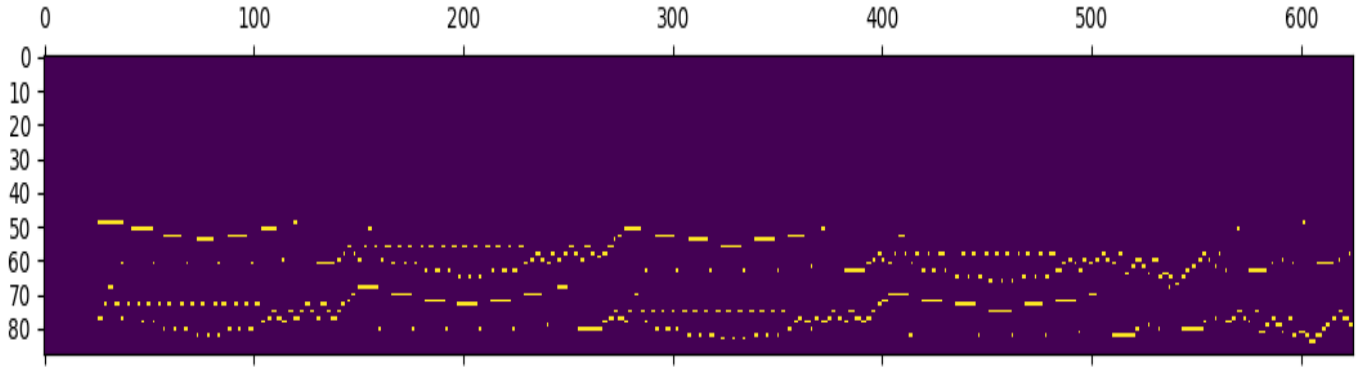


Figure 5: One hot encoded vector for a piano for 20 seconds

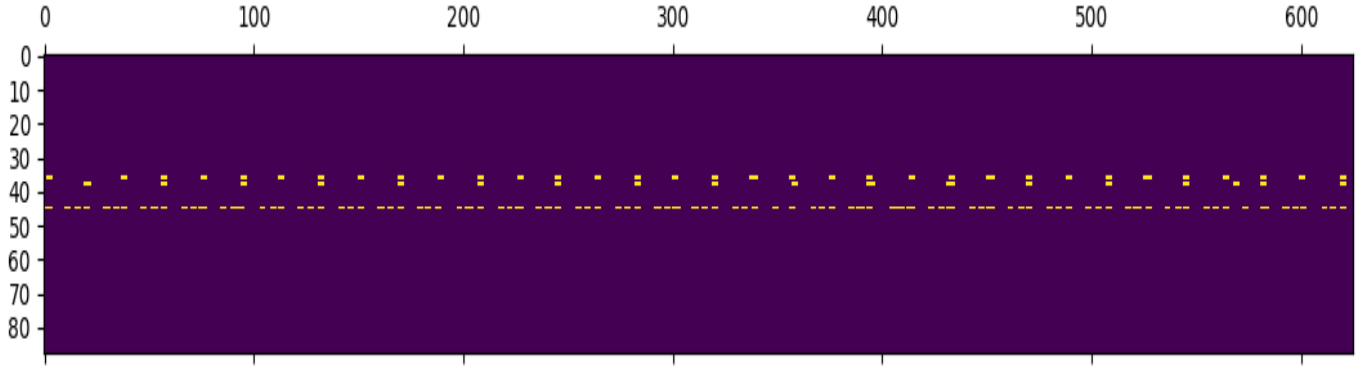


Figure 6: One hot encoded vector for a drums for 20 seconds

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 $Y = (y_0, y_1..., y_j)$ [3].

$$P_\theta(Y|X) = \prod_{j=1}^{J+1} P_\theta(y_j|Y_{<j}, X) \tag{6}$$

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 54%.

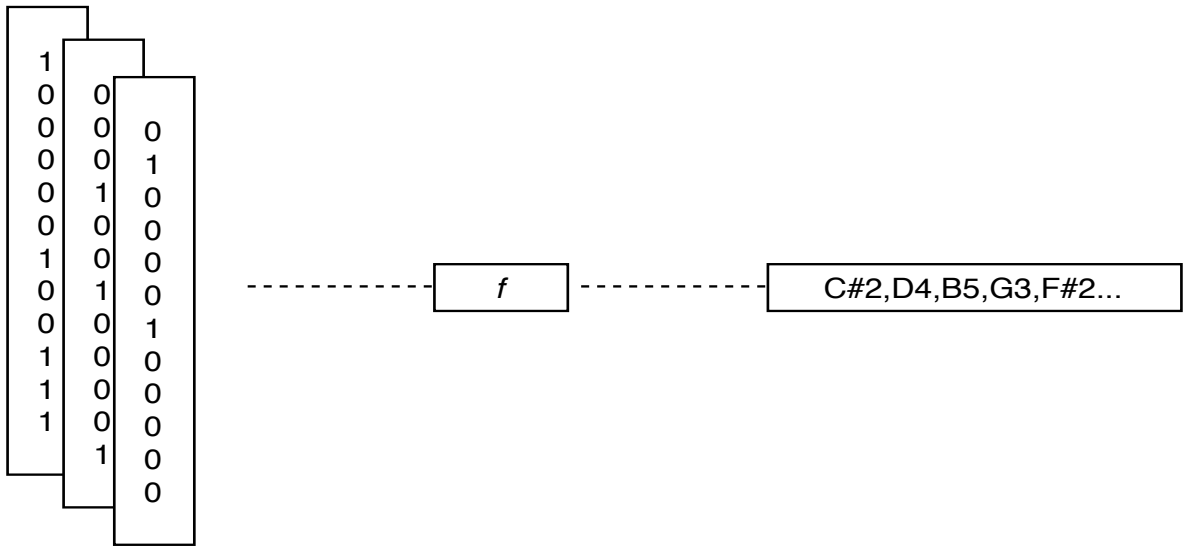


Figure 7: Mapping from hot encoded vectors to musical notes

Results

Performance metric is f1 score(Accuracy):

- VGG19 = 97%

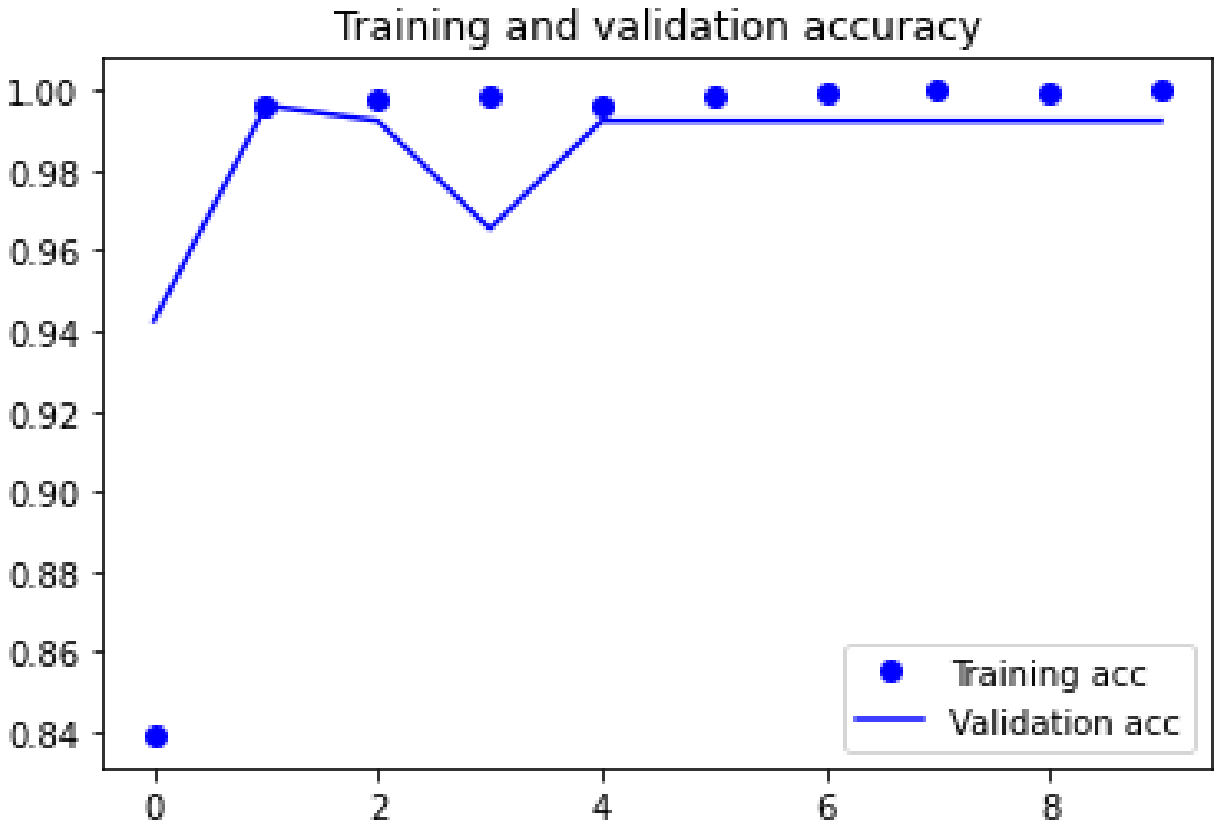


Figure 8: VGG19 Performance

Results (cont.)

- Sequence to Sequence using BiLSTM F1 Score: 54%

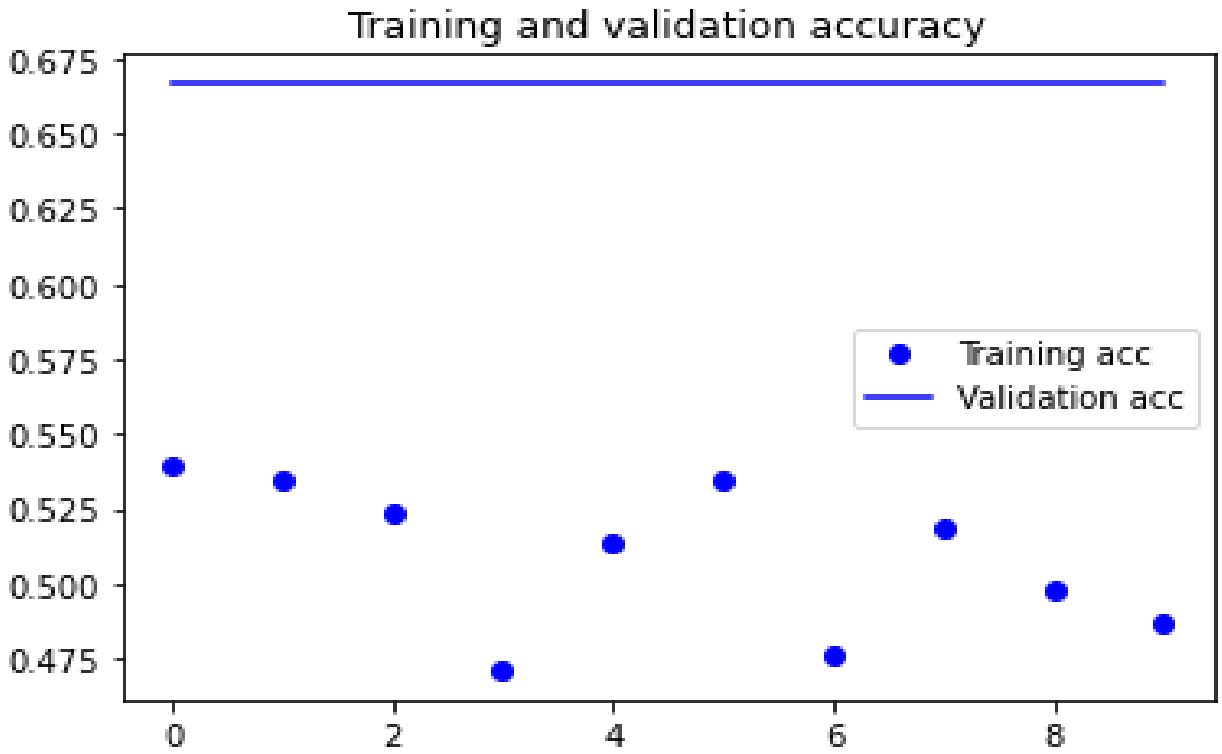


Figure 9: Sequence-To-Sequence Model Performance

Interpretation of results

- Good-Fitting Profile
- Within the Model's remit to perform well with small data sets

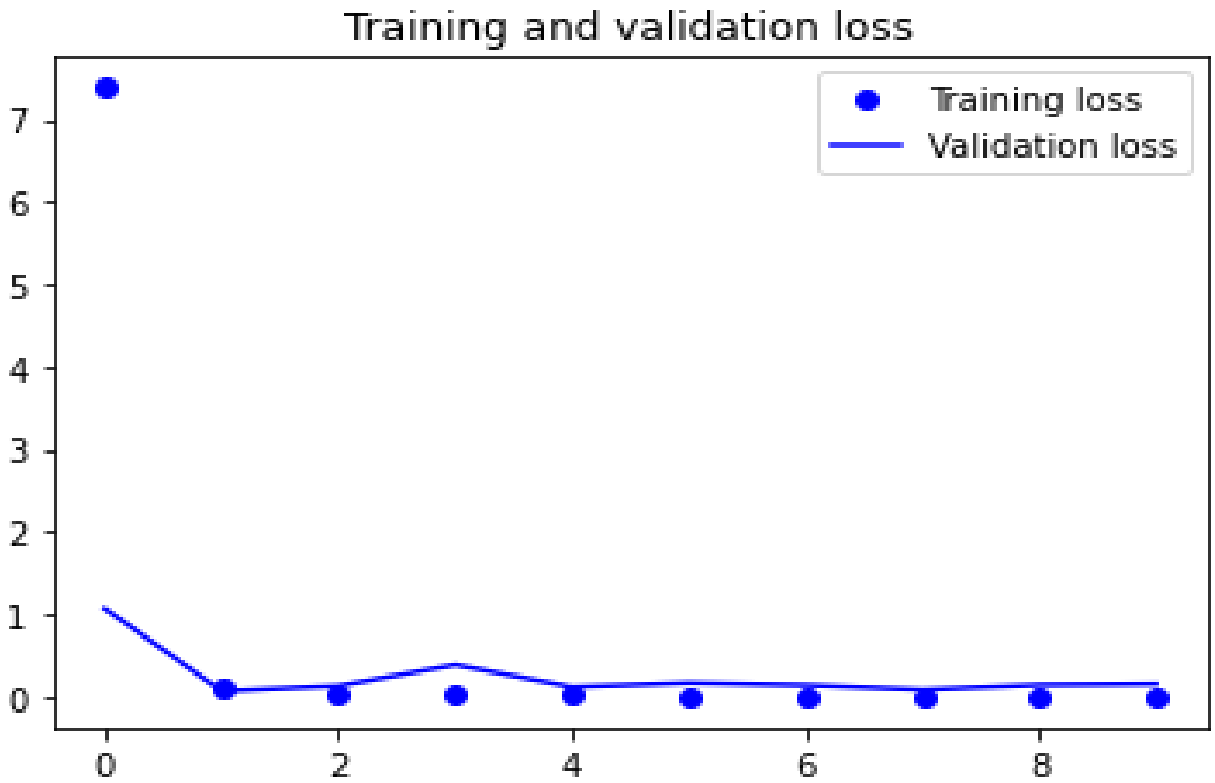


Figure 10: VGG 19 Training and Validation Loss

- Under-fitting
- Causation: Lack of abundance in Data
- Increase depth of the model

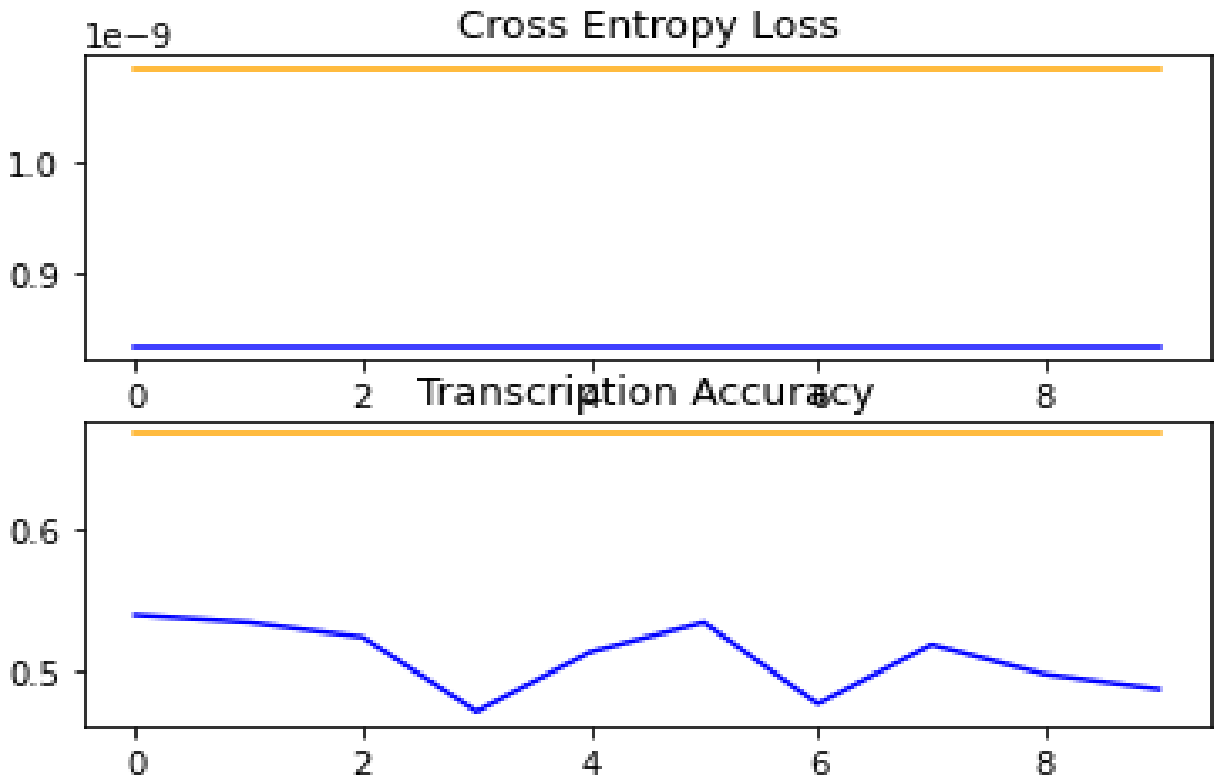


Figure 11: Sequence To Sequence Model Loss profile

User Interface

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

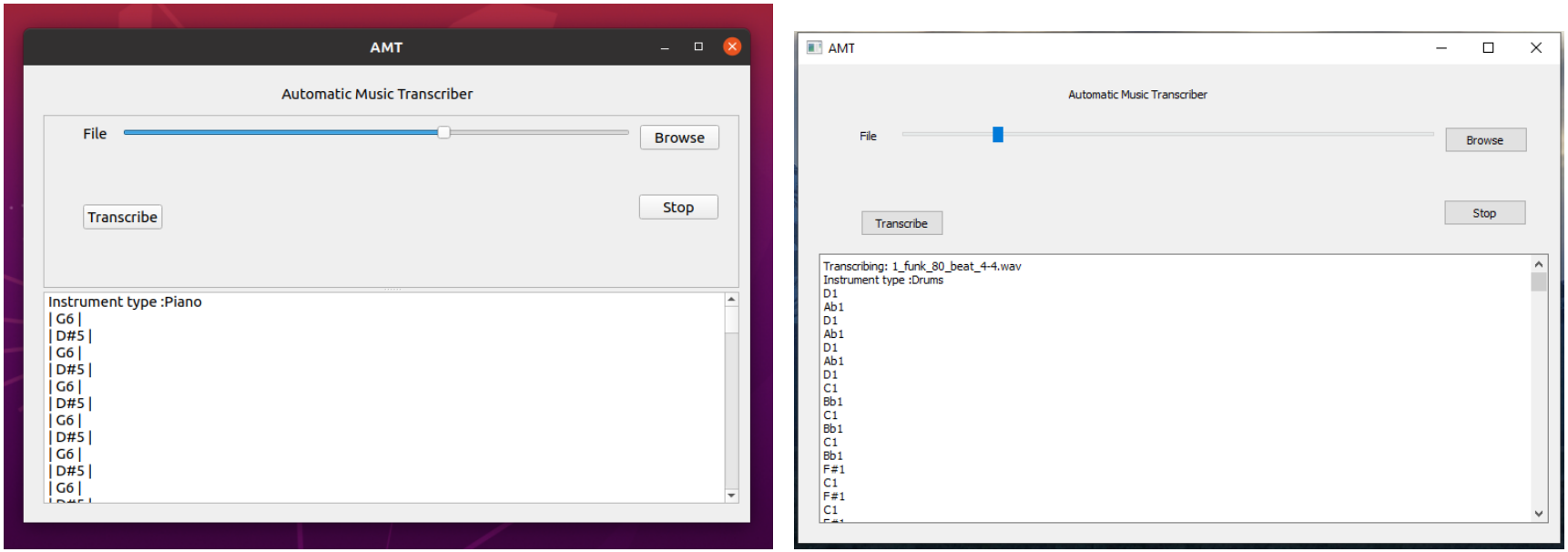
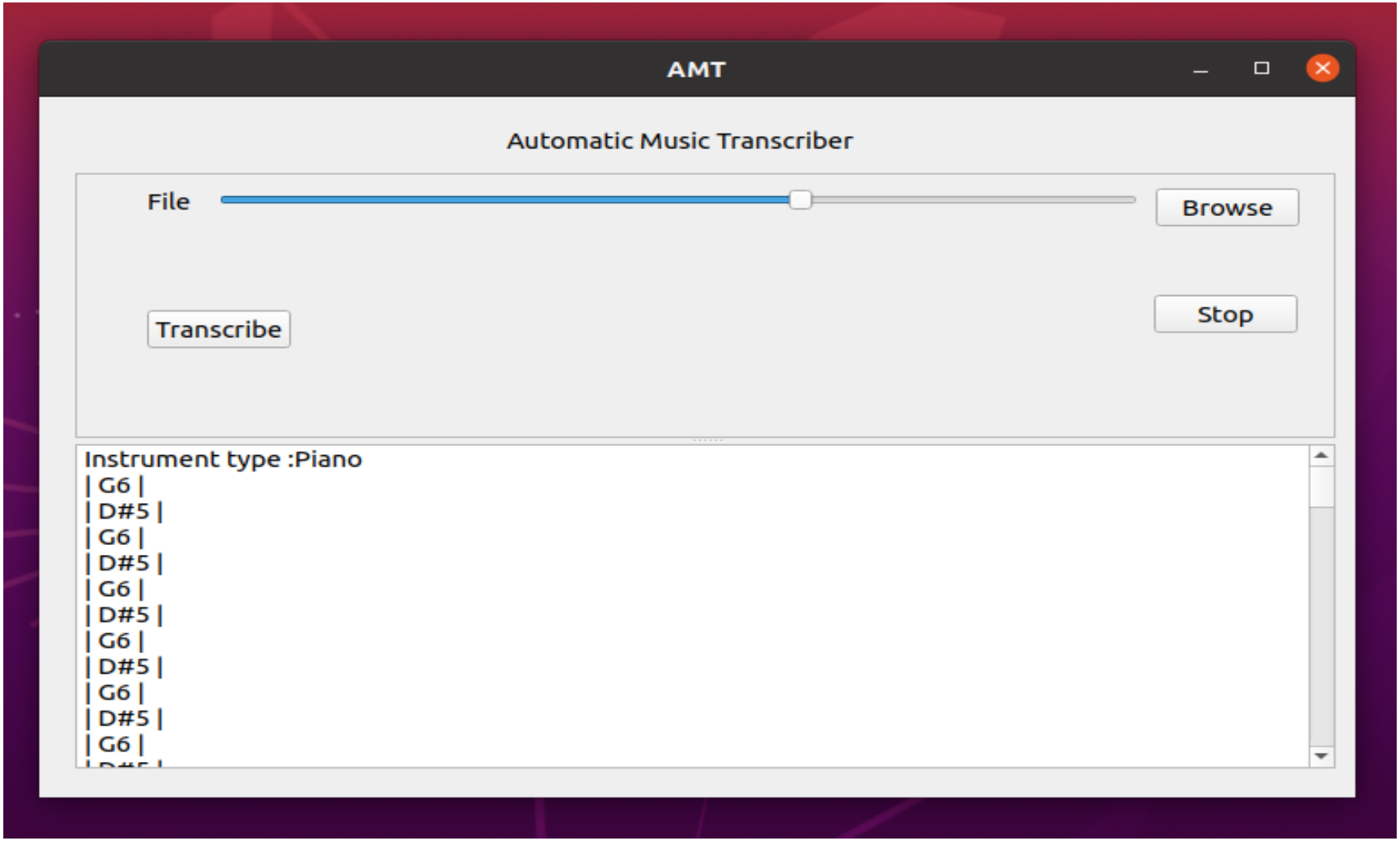


Figure 12: UI in both Ubuntu and Windows os

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.