Automated Music Transcriber for two Musical Instruments

Sbonelo Mdluli and Moshekwa Malatji

Supervisor: Prof. Olutayo O Oyerinde

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

SCHOOL OF ELECTRICAL AND INFORMATION ENGINEERING

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].

SCAN ME

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 and be operation on Microsoft Windows and Linux operating systems. The GUI should illustrate the instrument classified and the musical notation that is the output of the AMT.

Crisp-DM Methodology..

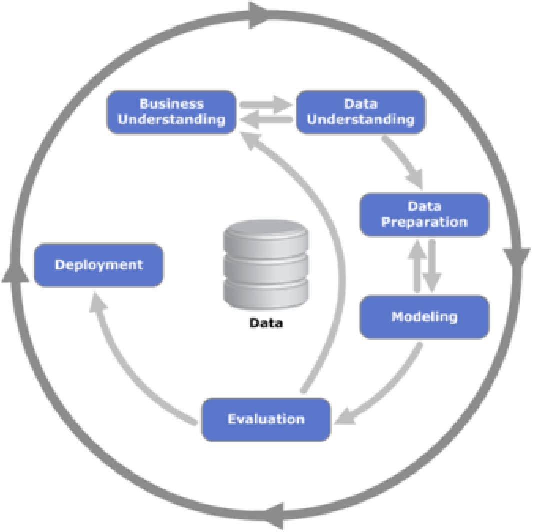
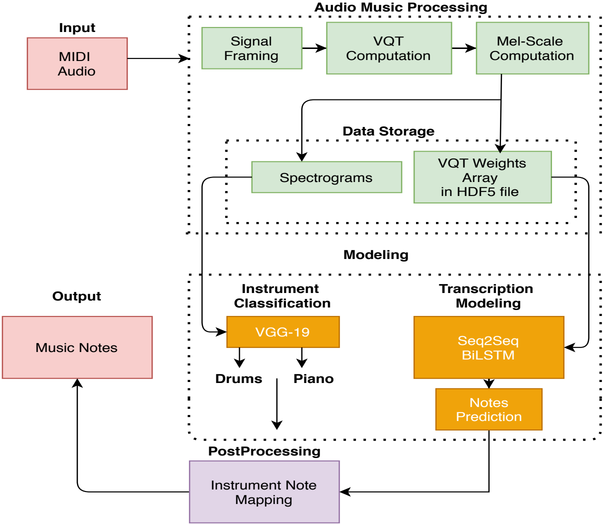


Figure 2: The CRISP-DM Methodology

Crisp-DM Methodology.. (cont.)

1. Business Understanding: This phase focuses on outlining the project specifications, objectives, assumptions, constraints to be adhered to and the relevant success criteria for successful.
2. Data Understanding: Collection of music data and subsequent tasks involve exploration of the data by identifying patterns, describing the data and drawing insights which verify the quality of the data. E.g. Monophonic Data, MIDI Format.
3. Data Preparation: transformation and processing the data.Imperative for pre-processing data for modeling purposes.
4. Modeling: Selecting and applying modeling techniques on the transformed data to achieve certain functionality.
5. Evaluation Deployment: Reviewing and assessing the results.Verify if they are a reflection of the specifications and success criteria. Documentation and presentation.

Automatic Music Transcription Framework



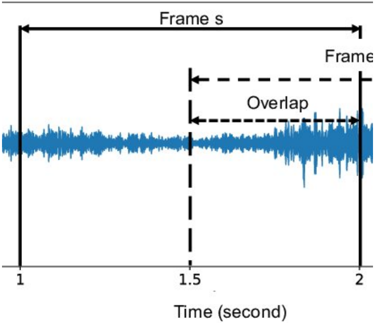
Dataset

The proposed instruments to be transcribed are the piano and drums. The choice of these instruments was inspired by the abundance of their respective dataset and their significance in history of the music industry. are not confined to one genre.

- The 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. Furthermore, the dataset was performed by 10 professional drummers with Roland TD-11 electronic drum kit.
- The Piano dataset contains over 200 hours of paired audio and recorded MIDI data from performances on Yamaha Disklaviers.
- The Groove MIDI Dataset drum and the Maestro piano datasets both consists of recorded MIDI data from performances by professional drummers and pianist eliminating question of human error.

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. 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:



Variable Q-Transform(VQT) (cont.)

Figure 3: Signal Framing Procedure

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 mu, 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.
- Temporal resolution at lower frequencies

Mel-Spectrogram

Classification model should be implemented such that it is realistic as possible. It is for this reason that the Mel-Frequency Scale is employed to achieve non-linear transformation of the frequency scale which is an emulation of how human perceive frequencies.

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.. Logarithmic ally spaced?!!!

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

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

Mel-Spectrogram (cont.)

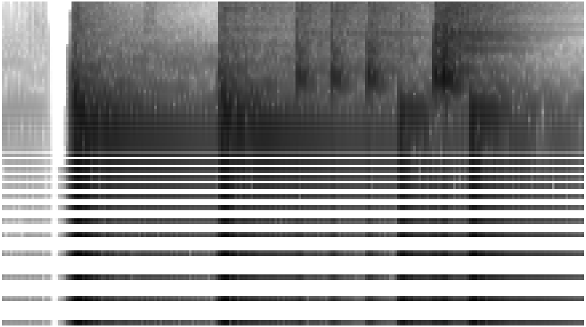


Figure 4: Mel-Scaled Spectrogram

Input Features

Classification model:

- Spectrogram Image from the resulting Mel-Scale Frequency Transformation.
- The VQT data matrix of dimensions in a HDF5 in preparation for modeling
- Dimensions (625,128,7): 625 windows, 7 frames per window for 128 filter banks
- It is imperative we take the absolute value of the VQT matrix values, such that the result only has real numbers to reduce complexity for the Keras implementation of the transcription model.

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.)

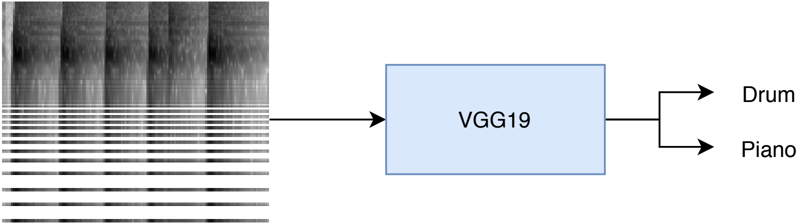


Figure 5: Instrument classification high level

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

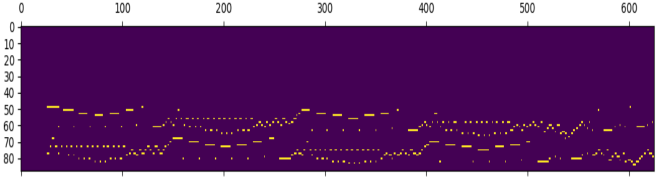


Figure 6: One hot encoded vector for a piano for 20 seconds

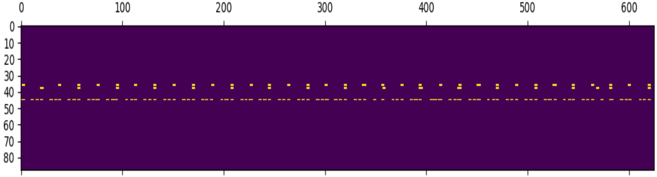


Figure 7: One hot encoded vector for a drums for 20 seconds

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%.

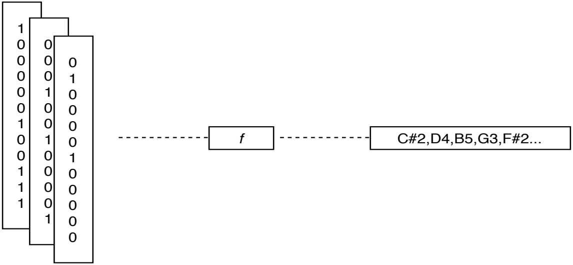


Figure 8: Mapping from hot encoded vectors to musical notes

Results

Performance metric is f1 score(Accuracy):

- VGG16 = 94%

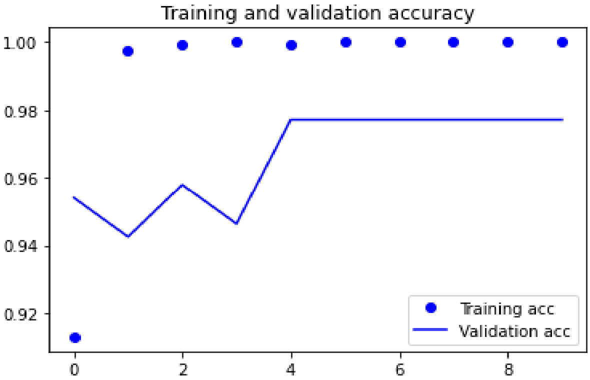


Figure 9: VGG16 Performance

Results (cont.)

- VGG19 = 97%

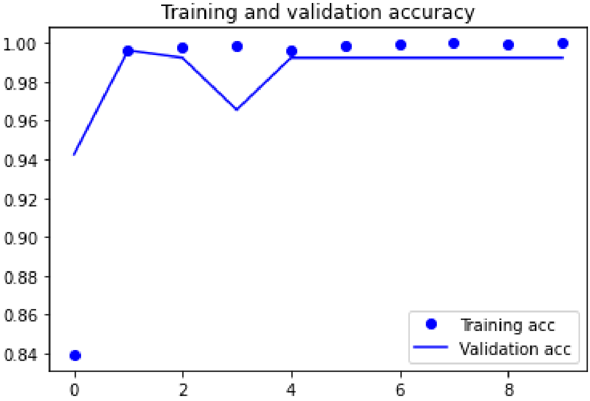


Figure 10: VGG19 Performance

- Sequence to Sequence using BiLSTM F1 Score: 54%

Results (cont.)

- Training Accuracy > Validation :

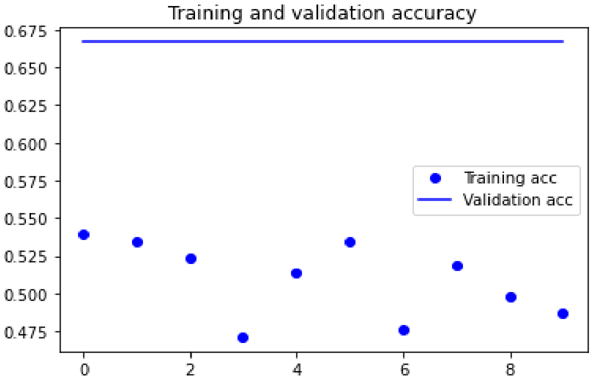


Figure 11: Sequence-To-Sequence Model Performance

Interpretation of results

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

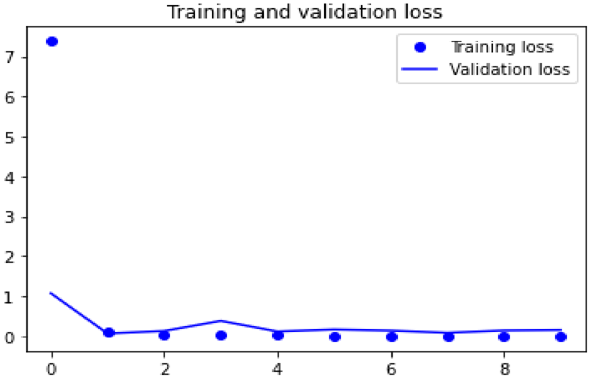


Figure 12: VGG 19 Training and Validation Loss

- Under-fitting

- Causation: Lack of abundance in Data

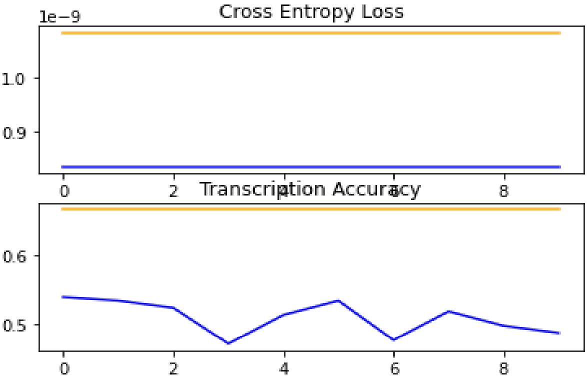


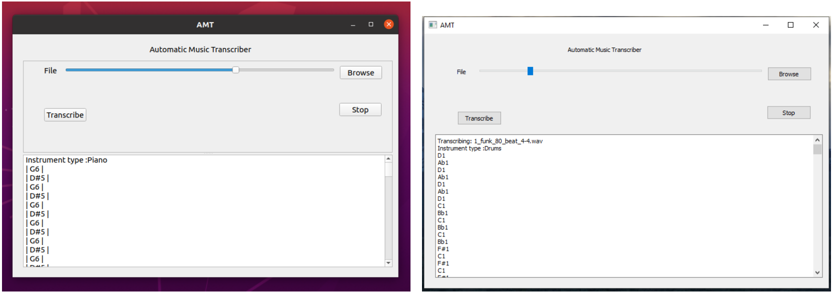
Figure 13: Sequence To Sequence Model Loss profile

Recommendations for future work

- The model the transcription model is under-fitting.
- training the seq2seq model in parallel with a note velocity model in ordered to develop a more comprehensive model that will also predict note duration.
- The seq2seq model can also be improved by using more stacked layers, training the model using more data and increasing the number of epochs.
- Generate more data using MIDI.

User Interface

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

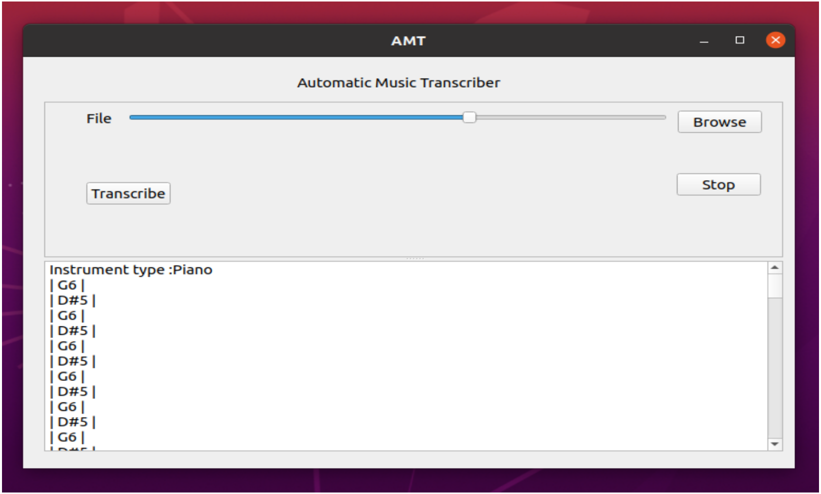


```
Instrument type :Piano
| G6 |
| D#5 |
| G6 |
| D#5 |
| G6 |
| D#5 |
| G6 |
| D#5 |
| G6 |
| D#5 |
| G6 |
```

```
Transcribing: 1_funk_80_beat_4-4.wav
Instrument type :Drums
D1
Ab1
D1
Ab1
D1
Ab1
D1
C1
Bb1
C1
Bb1
C1
Bb1
F#1
C1
F#1
C1
```

Figure 14: UI in both Ubuntu and Windows os

Prototype Demonstration



```
Instrument type :Piano
| G6 |
| D#5 |
| G6 |
| D#5 |
| G6 |
| D#5 |
| G6 |
| D#5 |
| G6 |
| D#5 |
| G6 |
```

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.