Automated Music Transcriber for two Musical Instruments

Sbonelo Mdluli: 1101772

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: This paper presents an Automatic Music Transcription (AMT) model capable of transcribing monophonic piano and drum music. The model is developed using MAESTRO and Groove data sets. The model starts by performing instrument classification based on audio file spectrograms. Variable Q-Transform (VQT) is used to produce the spectrograms because it has a better resolution at lower frequencies compared to Constant Q-Transform (CQT). The transcription model predicts one-hot encoded vectors when given frequency windows. The indices of the one-hot encoded vectors correspond to MIDI note numbers. The one-hot encoded vectors are further processed to musical notes using a dictionary for each instrument. The specific dictionary is determined by the result of the instrument classifier. The one-hot encoded vectors contain binary values indicating onset or offset event per window. A new window is considered every 0,032 seconds. The classification uses VGG19 through transfer learning. The classification model achieves an accuracy of 97%. The transcription model is a seq2seq model with BiLSTM encoder and decoder layers. The transcription model has an F1-score of 54%. A UI is used to display the transcribed notes for user convenience. The model is underfitting and can be improved by using more training data and early stopping. A more comprehensive model can be made to include a note velocity and duration prediction.

Key words: VQT, Automated Music Transcription, seq2seq, BiLSTM

1. Introduction

Automatic Music Transcription (AMT) is the process of converting audio signals to musical notes using computational algorithms. This is a complex and challenging process even for expert musicians. The complexity in transcription is because of multiple factors such as instrument type, harmonic distortion, overlapping notes and a mixture of multiple simultaneous audio signals sources. Generally, human transcribers still outperform software-based transcription systems. AMT can be applied in information retrieval, composition, music education, and music visualization. Musical notes can be represented using music scores, piano-roll representations, or symbolic notation. The features that characterise music are onset detection, offset, pitch, loudness, and velocity.

Previously AMT had been accomplished using signal processing methods. The methods mostly studied for transcription in recent years are non-negative matrix factorization (NMF), Probabilistic Latent Component Analysis (PLCA) and neural networks. NMF is a linear model compared to neural network based approaches as such it fail to generalise which makes it more prone to errors when modelling nonlinear systems [1] [2]. Neural network-based models are mostly limited by the availability of labeled music data, available data sets tend to be biased because they largely contain Western music. Music is played from a variety of instruments and post-processing instruments which makes it difficult to standardise [1].

Transcription approached mainly fall into four categories which are note level, stream level, frame level and notation level. Frame level aim to detect present fundamental frequencies per given time interval. With note level each note is defined with its respective onset and offset time duration which makes it possible to reproduce piano roll representation of the signal. Stream level associates each note to a particular instrument. Notional level aim to produce music score which can be used to produce the original recording, this level is the ultimate goal for AMT [3].

2. Background

2.1 Literature Review

In recent years neural network based techniques have surpassed traditional signal processing methods. A variety of neural network architectures has been explored for AMT by different researchers. Sigtia et al. use a supervised neural network model based on speech recognition [4]. They use Deep Neural Networks to build an acoustic model and a Recurrent Neural Network to develop a Music Language Model, which captures long-range dependencies between notes. They use a CovNets to identify pitches present in a given spectrogram from CQT. Their model achieves an F-Measure of 0.7476. [5] propose a framewise transcription network, which outperforms other convolutional neutral networks (CNN) architectures. The model by Kelz et al. contains a separate model for offset detection; they also explore the effect of learning rate and spectrogram type in transcription accuracy. Their model avoids the entanglement problem and the glass ceiling effect which are common in other models. Motivated by Kelz et al. [6] uses a ResNet instead of a CNN whereby they achieve a better recall score using a ResNet. The ResNet is followed by a Bidirectional Long Short Term Memory (BiLSTM), they train to detect pitch occurrences and onset occurrences separately. [7] uses a U-Net as a front end to create an instrument-specific representation before a CNN. The instrument-agnostic transcription model proposed by [7] achieves about a 1% improvement over VGG and RES18.

2.2 *Project Requirements*

The aim of this project is to develop a transcription system capable of transcribing two musical instruments. The system is required to display notes in real-time in a Graphical User Interface. The specific instruments to be transcribed by the system consist of the piano and drums. This choice is dictated by the availability of annotated data needed to train the models.

2.3 *Project Constraints and Assumptions*

We assume that the signal source only contains monophonic music. It is also assumed that the music to be transcribed contains no vocals over them only instruments.

2.4 *Project Success Criteria*

This project will be deemed as successful if the system can accurately transcribe the selected instruments. The transcription accuracy is required to achieve a minimum transcription accuracy of 50 %. The user must be able to interact with the AMT application through a user interface.

3. System Design & Methodology Overview

The proposed AMT system comprises of several subsystems. The subsystems generally fall into two categories namely; signal processing and deep learning.

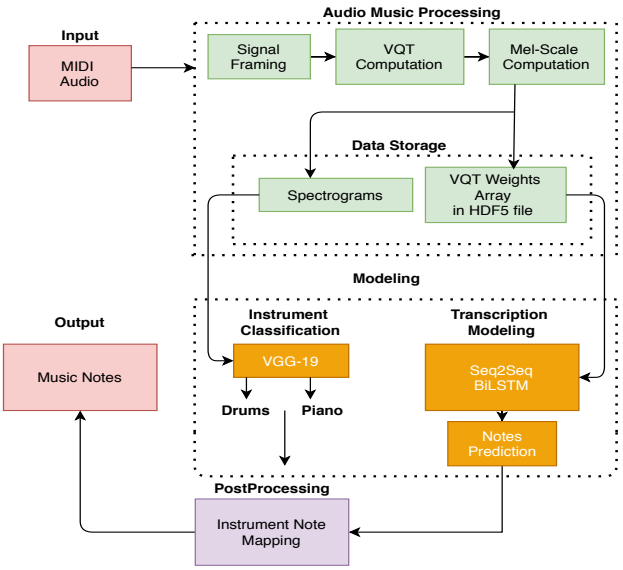

Figure 1 : AMT system overview, with connections between sub systems.

The system takes in an audio file as an input. The audio file is then processed to produce spectrograms and frequency windows using digital signal processing techniques. The spectrograms are used for instrument classification and the frequency windows for note prediction. The output from the note prediction are one-hot encoded vectors with indices corresponding to MIDI note numbers. A post-processing step is necessary to convert these vectors to musical notes. The result from the instrument classification stage is used to determine the appropriate transformation for the notes.

4. Datasets

The instrument choice is largely dependent on available annotated data. The data sets for drums and pianos can be accessed through Magenta. MAESTRO is a data set that contains over 200 hours of virtuosic piano recordings. The data audio and MIDI files are aligned within 3 ms tolerance. The piano audio is recorded at CD quality. The Groove data set has 1,150 MIDI drum files, the data set uses Roland mapping.

5. Signal Processing

Signal processing is performed on the audio signal in order to get a more compact representation of the signal. Common signal processing methods used in literature for AMT systems usually make use of Discrete Fourier transform (DFT), Fast Fourier transform (FFT), Short-time Fourier Transform (SFT) and Constant Q-transform (CQT). These techniques are used to convert the audio signal into a spectrogram which is a time-frequency representation of the signal. CQT outperforms the other aforementioned signal processing methods and is more suitable for music signals [8][9]. In this project, we use Variable Q-Transform (VQT) which is a more general form of CQT to spectrograms. VQT consists of a k passband filter banks, whereby the filters are linearly spaced in the frequency axis. The Quality factor(Q) for VQT is defined using Equation 1 below:

$$Q = \frac{f_k}{(2^{\frac{1}{b}} - 1)f_k + \gamma} \tag{1}$$

Where f_k is the center frequency of the k-th filter bank, b represents the number of bins per octave and γ is a frequency offset between the filters. CQT is a special case of VQT with $\gamma = 0$. The value of γ .The k th spectral components of VQT for a signal $x[n]$ can be obtained using the discrete Fourier transform Equation 2.

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k,n]x[n]e^{\frac{-j2\pi Qn}{N[k]}} \tag{2}$$

where $N[k]$ is the width of the k th bin.

$$N[k] = Q\frac{f_s}{f_k} \tag{3}$$

and $W[k,n]$ is the Hann window function which is defined using Equation 4. Hann window is a special case of the more general Hamming window.

$$W[k,n] = sin^2(\frac{\pi n}{N[k]}) \tag{4}$$

The sampling frequency f_s is calculated based on the human hearing limit which is about 20 kHz from using therefore the Nyquist sampling frequency is determined to be $f_s = 44$ Khz.

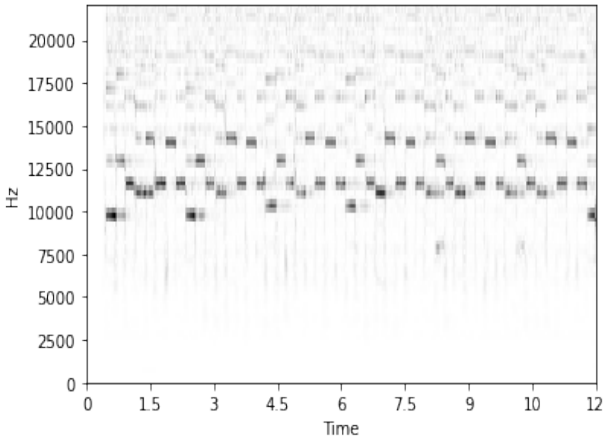

Figure 2 : Variable Q-Transform spectrogram (CQT) $\gamma = 0$.

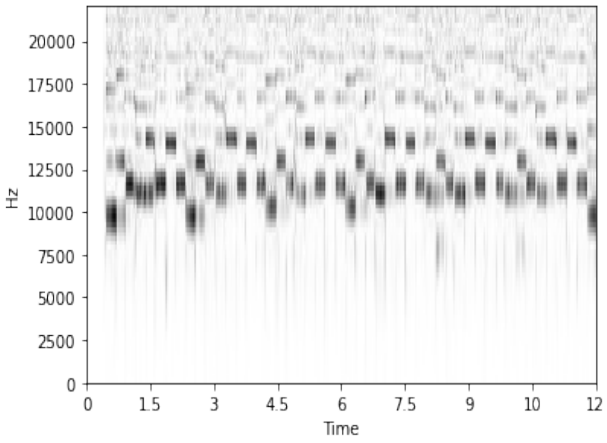

Figure 3 : Variable Q-Transform spectrogram $\gamma = 20$.

VQT has better resolution at lower frequencies compared to CQT, which improves the Q factor. The spectrograms are represented using the Mel scale, which is a nonlinear transformation that resembles the way humans perceive sound.

6. Signal Framing

The audio signal is broken down into several windows for feature extraction. The window length is dependent on hop length and the sampling frequency. The samples per frame are set to 512, the audio is further down sampled from 44 Khz to 16 kHz. This translate to $16,000/512 = 31.25$ windows per second for 20s this results in $31.25 \times 20 = 625$ windows. Each window comprises of 7 frames with each window overlapping the previous window with an offset of 1 frame. This results in the dimension of the data being $(625 \times 128 \times 7)$ where 128 represents the batch size. This batch size is related to the one used in section 8.

7. Instrument Classification

The post-processing step, which requires MIDI score to be translated to musical notes, depends on correct instrument classification. The training data is split into two classes each indicating a different instrument. The classification model developed in the project is based on transfer learning. Transfer learning significantly decreases the time required to train a model.

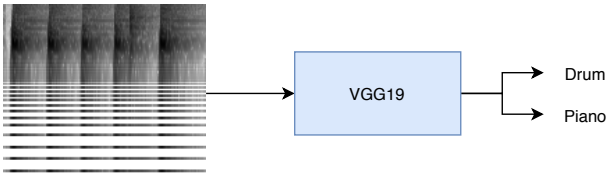

Figure 4 : Instrument classification model.

VGG is a specific type of pre-trained CNN architecture trained using the ImageNet data set. 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. VGG contains filters with a stride length of one and pad size one. They have pooling layers that perform 2x2 max pooling with stride 2 without any padding. The pooling layer performs non-linear down-sampling the output from the feature map.

$$relu(x) = max(0,x) \tag{5}$$

We modify the VGG by appending 2 additional layers at the output. The models are further retrained using spectrograms for instrument classification. The input to the classification model has dimensions $(224 \times 224 \times 3)$, where the first two dimensions represent the image width and height and the last one the image R,G,B channels. The model uses stochastic gradient descent with a learning rate of 0.001 and a momentum of 0.9 to reduce training time. The model is trained using 10

epochs each with a batch size of 10. We use a binary cross entropy loss function (Equation 6) with N labels and sigmoid as the activation function to predict the instrument type.

$$L = \frac{1}{N}\sum_{n=0}^{N} y_n log(\hat{y_n}) + (1 - y_n) log(1 - \hat{y_n}) \quad (6)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

The sigmoid function output a value between 0 and 1. The output represents the likelihood of a spectrogram belonging to an instrument class. A threshold value of 0.5 is used for the output, where the prediction $1 \geq \hat{y_n} \geq 0.5$ indicates a piano and $0 \leq \hat{y_n} < 0.5$ indicates drums.

8. Transcription Model

The transcription model is tasked with predicting the one hot encoded vectors given a frequency window. The musical notes are modelled as a time a dependent sequence with 2^{88} (on or off note for 88 notes) possibilities in a given time step. The problem can be rephrase as follows: given a frequency window $X = x_0, x_1 ..., x_i$ what is the probable one-hot encoded vector $Y = y_0, y_1 ..., y_j$ for that window. This can be summarised using Equation 8.

$$P_\theta(Y|X) = \prod_{j=1}^{J+1} P_\theta(y_j|Y_{<j}, X) \quad (8)$$

A seq2seq is one model that solves such a problem. A seq2seq model takes in a sequence and outputs another sequence. The model uses a BiLSTM layer as an encoder, which scans the input sequence and converts it to hidden vectors. The decoder layer is also implemented using a BiLSTM. The decoder layer computes the probability of the j-th note in Y. BiLSTMs are a type of RNN capable of capturing long-range temporal dependencies. LSTM contain structures called memory cell, which modify the information in a network. The information flow in a memory cell in controlled by gates. The forget gate is used to discard information no longer needed by the memory cell. The input gate dictates when new information is added to the memory cell and the output is responsible for determining the output based on input data. Seq2seq and BiLSTMs have been extensively used in speech recognition, language modelling and translation applications. As such the model is capable of modelling musical note progression. We train the model using batch size 128 with 50 epochs.

In contrast to [10] this model does not have an explicit musical language model; it also does not consider note velocity. This simplifies the modelling processes however, most information is lost due to less training features. Velocity is an important feature, which captures the speed at which a note is played. We get note duration based on the frames, considering that each frame takes 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. This also means that model does not take into account the note duration. The model uses a threshold value of 0.5 to determine an onset or offset event.

8.1 One-hot encoding

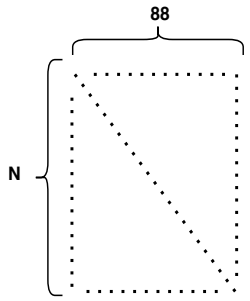

Figure 5 : one hot encoded matrix with binary values. The matrix has dimensions $N \times 88$, where $N = 625$ windows and 88 values per window.

The indices of each row vector correspond to notes at a given widow where 1 indicates an onset event and 0 an offset event. piano = $\{0, 1, 2, ..., 87\}$ and drum = $\{35, 36, 37, ..., 80\}$ which means drum $\subset$ piano for this reason the 88 indices are sufficient to represent both instrument note ranges. The one-hot encoded vector can be represented as a piano roll.

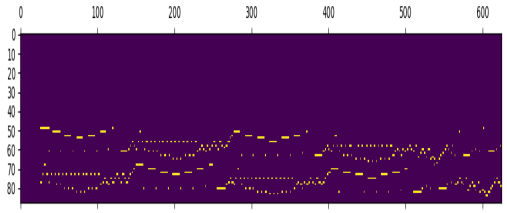

Figure 6 : Piano roll representation for piano for 20 seconds playback. The notes are between 0 and 87.

There are distinct differences between the two instruments. Drum notes generally periodic and do not overlap between different frames.
The piano notes tend to be less predictable and overlap between several frames compared to the simple

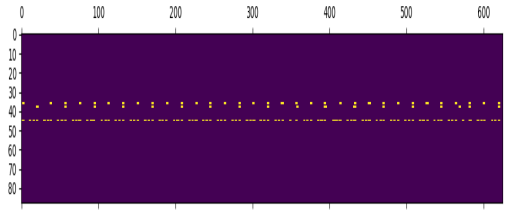

Figure 7 : Piano roll representation for drum for 20 seconds playback. The drum notes are between 35 and 80.

pattern observed with drums. The figures above confirm that the 88 indices can represent both instruments.

9. Post processing

The post-processing step is needed in order to transform the one-hot encoded vectors from the transcription model into musical notes. The indices of the one-hot encoded vectors are transformed to notes N using mapping function $f: Y \rightarrow N$ using the dictionary data structure. Each dictionary maps a MIDI note number to musical notes. We make use of two separate dictionaries one for drums and the other for a piano. The specific dictionary to be used for an audio signal is determined using the output from the instrument classification stage.

10. Transcription Application User Interface

The UI is developed for Windows and Linux operating Appendix E, Figure 8. The UI is developed using `PyQt5`, in Python. The user is able to select an audio file from the navigation tab. The UI displays both the instrument type being transcribed and relative musical notes. The output from the transcription model is saved so that should the user want to transcribe the same audio file the output is automatically loaded in order to save computation time. The UI does not display note duration and only unique notes between frames are displayed.

11. Experiments and Results

The audio files are processed using `librosa` in Python. The VQT frequencies to be used for the seq2seq model are saved in as H5. The H5 format makes it easy to process multidimensional data. The one-hot encoded vectors are saved in a CSV file.

The models are developed in Python using `Keras` which uses `Tensorflow` as a backend. Due to limited computing resources, the models are trained on the Google cloud platform using Tesla K80 GPU. The number of epochs for the seq2seq model was limited by the GPU maximum training time which is 12 hours. The data for classification is split into training date = 75%, testing data = 15% and 10% as validation data. We investigate two classification models mainly VGG16 and VGG19. VGG19 takes much longer to train compared to VGG16 because it contains 3 additional layers compared to VGG16 which only has 16 layers. The classification model results are in Appendix E, Figure 4 - Figure 7.

Table 1 : Instrument classification results.

	VGG16	**VGG19**
Training accuracy	94%	97%
Training loss	0.75	0.1

Based on these results the final transcription model makes use of VGG19 for more reliable results.

The input dimensions to the seq2seq model are (128×7). The window length N is set to 625 by default. However, some songs are less than 20 seconds ($N < 625$), in such cases we use the full length of the audio file for training. The transcription model takes a bit of time to do note prediction due to the large probability space (2^{88}) per window. The transcription model is trained using 2100 files, 80% for training, and 20% validation. Due to windowing, the model performs discrete predictions whereas music is a continuous signal. This means that the model misses notes within two sampling points.

The metrics used to evaluate the performance of the transcription model are precision, recall, and f-measure. The F1-score metric is a ratio that combines all these metrics in a single quantity. The equations below define the aforementioned metrics.

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$F1 = \frac{2RP}{R + P} \tag{11}$$

The F1 score is implemented using a custom callback since Keras does not have an implementation for the F1 score. The transcription model has an overall F1-score of 54%. The results can be seen in Appendix E, Figure 1 - Figure 2.

12. Model Evaluation and recommendations

The paper by [11] uses PLCA and CQT for multi-instrument transcription. They use a HMM as a post-processing unit for smoothing out the result from the

PLCA. The HMM models each note as an active or inactive state, this model has a mean accuracy of 61%. The method used by [7] uses a U-net as a front end instead of instrument classification as we did. The work by [12] for multi-instrument transcription uses combined frequency and periodicity and Harmonic CQT to represent the audio files. They use a CNN model based on DeepLabV3 to predict notes directly from spectrograms. Our model mainly differs from the one mentioned mainly through instrument classification and the seq2seq model. The models mentioned aim to transcribe more than two instruments whereas the one we propose is only for two instruments. This simplifies the overall architecture and makes it possible for us to specify the mapping structure for each of the instruments we transcribe. Our model performs poorly compared to the ones mentioned due to our model underfitting. The underfitting is as a result of data imbalances. There is more piano data compared to other instruments.

Future modifications to the model might include 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.

13. Conclusion

In this paper, we present a music transcription model for monophonic piano and drum music. The model consists of instrument classification and a seq2seq model. Variable Q-Transform is used to produce spectrograms used to train the instrument classification model. The instrument classification model is based on VGG19, the model has an accuracy of 97%. The transcription is based of seq2seq model with BiLSTM layers. The transcription model achieves an F1-score of 54% .The output from the transcription are one-hot encoded vectors with indices corresponding to MIDI note numbers. The one-hot encoded vectors are transformed to musical notes based on the instrument type. A user is able to interact with the system through a UI that is developed through `PyQt5`. The model is underfitting and can be improved to include a note velocity and duration prediction. One can also increase the F1-score by training the model using more data.

Acknowledgements

I would like to thank Prof. Olutayo O Oyerinde from the School of Electrical and Information Engineering. for his guidance and support and Matthews Malatji [13] for his contribution and motivation though out the project.

References

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert. "Automatic Music Transcription: An Overview." *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] E. Benetos, T. Weyde, et al. "An efficient temporally-constrained probabilistic model for multiple-instrument music transcription." 2015.

[3] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza. "A holistic approach to polyphonic music transcription with neural networks.", 2019.

[4] S. Sigtia, E. Benetos, and S. Dixon. "An End-to-End Neural Network for Polyphonic Music Transcription." *ArXiv*, vol. abs/1508.01774, 2015.

[5] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer. "On the Potential of Simple Framewise Approaches to Piano Transcription." *CoRR*, vol. abs/1612.05153, 2016. URL `http://arxiv.org/abs/1612.05153`.

[6] A. C. Jaedicke. "Improving Polyphonic Piano Transcription using Deep Residual Learning." 2019.

[7] F. Pedersoli, G. Tzanetakis, and K. M. Yi. "Improving Music Transcription by Pre-Stacking A U-Net." In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 506–510. IEEE, 2020.

[8] M. Karioun and S. Tihon. "Deep Learning in Automatic Piano Transcription."

[9] R. A. Dobre and C. Negrescu. "Automatic music transcription software based on constant Q transform." In *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pp. 1–4. 2016.

[10] Q. Wang, R. Zhou, and Y. Yan. "Polyphonic piano transcription with a note-based music language model." *Applied Sciences*, vol. 8, no. 3, p. 470, 2018.

[11] E. Benetos and S. Dixon. "Multiple-instrument polyphonic music transcription using a convolutive probabilistic model." In *8th Sound and Music Computing Conference*, pp. 19–24. 2011.

[12] Y.-T. Wu, B. Chen, and L. Su. "Polyphonic music transcription with semantic segmentation." In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 166–170. IEEE, 2019.

[13] M. Malatji. "Automatic Music Transcription for Two Instruments based Variable Q-Transform and Deep Learning methods." 4th Year Project Report 20P26, School of Electrical and Information Engineering, University of the Witwatersrand, South Africa, 2020.

APPENDIX

A PROJECT SPECIFICATION OUTLINE

School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
ELEN4002/4012: Project Specification Outline

Project Title: Automated Music Transcriber for two Musical Instruments

Group Number:	20G04	Supervisor Name:	Dr. Olutayo Oyerinde
Student Name A:	Sbonelo Mdluli	Student Name B:	Moshekwa Malatji
Student Number A:	1101772	Student number B:	1387556

Ethics: ☐ Request for waiver (does not involve human participants or sensitive data)
☐ Copy of ethics application attached (Non-medical) – School Committee
_____ ☐ Copy of ethics application attached (Medical) – University Committee
Supervisor Signature

Project Outline: *(give a brief outline, including the investigation methodology, such that ethics reviewers understand what will be done, and whether or not human participants will be involved, 100 words maximum)*

The aim of the project is to develop an automated music transcriber (AMT) capable of transcribing two instruments in real time. The systems is to trained using neural networks with appropriate architectures and compare performance with those in literature. In order to extract features for the training model, digital to signal processing (DSP) techniques are applied to the input audio signal. The models are trained using publicly available data sets.

Project Specification:

Literature Review:
The end-goal of a successful Automatic Music Transcription(AMT) starts with producing the music signal's spectrogram and to draw features for the models. According to [1] common audio signal processing approaches employ Constant Q-Transform and Discrete Fourier Transform, however, unlike the CQT the latter doesn't exhibit good spectral resolution at low frequencies. Another occurrence [2] of the Constant Q-Transform to obtain a time-frequency spectral representation of music signals. Spectrograms produced are feed to a convolutional neural network (CNN) to produce a piano representation of the input signal, which is then compared to the ground truth labels. Sigtia et al incorporate a recurrent neural network (RNN) to the CNN model as a post processing step to capture temporal dependencies between notes [2]. Other models can generate music notes in a single stage using a CRNN trained using a connectionist temporal classification (CTC) loss function. Other researchers have focused modelling the problem as a music language model (MLM).

Proposed methodology:
The audio signal processing proposed technique is the Variable Q-Transform, this was chosen over the common Constat Q-Transform to improve transcription performance, and obtain better spectral resolutions at lower frequencies. Furthermore, Noise reduction processes will most likely be applied to the music signal using Matlab. The models will likely be a CNN followed by a RNN. The output from the model will be a sequence of one hot encoded vectors, which will then be compared to the ground truth labels. Further post processing might be required to achieve a holistic model. The musical scores and/or notes will be visually displayed.

Data source and model testing:
MAPS database (31GB piano recordings), the MIREX database contains polyphonic music (some with drums only), ENST drums database and others. The confusion matrix, word error rate and character error rate, precision, recall and f-measure as performance metrics.

References:

[1] Shlomo Dubnov, "Unified view of prediction and repetition structure in audio signals with application to interest point detection", IEEE Transactions on Audio, Speech, and Language Processing, Vol. 16, No. 2, pp. 327-337, 2008.

[2] Schörkhuber, Christian & Klapuri, Anssi & Sontacchi, Alois. (2012). Pitch shifting of audio signals using the Constant-Q transform.

[3] S. Sigtia, E. Benetos, N. Boulanger-Lewandowski, T. Weyde, A. S. d'Avila Garcez and S. Dixon, "A hybrid recurrent neural network for music transcription," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, 2015, pp. 2061-2065, doi: 10.1109/ICASSP.2015.7178333.).

Milestones:

1. **Audio Signal Processing**: Obtain a time-frequency spectral representation of the music signal in order to draw relevant features for the models.
2. **Noise Reduction**: An important feature is to reduce the noise in the music signal.
3. **Data gathering and segmentation** (train, validate, test)
   The data will be split according to the recommend ratio (50:25:25), further processing might be required to clean/format the data
4. **Model Selection**
   The model selection and architecture will be based on those in existing work and selection of initial parameters as appropriate.
5. **Training**
   The model will be tested and validated on monophonic music.
6. **Model Evaluation (iterative)**
   The model will be tested using polyphonic. Model performance will be tested using the mentioned metrics.
7. **Parameter Tuning** (**iterative)**
   There might be a need for parameter (number of layers, learning rate, drop out, number of epochs, etc) adjustments depending on the model performance.
8. **GUI development** (**iterative)**
   Start building the Graphical User Interface in MATLAB
9. **Gradual system integration and testing (iterative)**
   Integration the MATLAB GUI with the model to build a complete system
10. **Analysis of results**
    Analyze results using proposed methods and model refinement.
11. **Presentation**
12. **Report/Documentation**

Preliminary Budget & Resources:

This project will be software based and digital resources will be used throughout project completion. Due to the current global pandemic and lockdown regulations, it will be a challenge to access campus and initiate face-to-face meetings and laboratory access. The university currently has a data provision service, assuming that will hold throughout lockdown, it will eliminate large costs of purchasing data.

The project will primarily employ MATLAB software and possible python for various computations and the following packages are required:

- MATLAB R2020a software, which contains the relevant libraries, required. The audio toolbox
- MATLAB Audio Toolbox, which provides tools for audio processing.
- MATLAB Signal Processing, which has tools for signal processing and analysis. This toolbox will also be helpful in extracting features to train the models.

Python provides many machine-learning libraries and most are open source and free to use.
In order to train the models we might need to access cloud services such as (Google/Aws/Azure) in order to reduce training time approximately $120.

.

Risks / Mitigation:

Risk	Chance	Priority	Mitigation
Not being able to host digital meetings on Microsoft teams and Zoom due to bad network coverage	High	High	Properly plan ahead for meetings and prioritize communication over email with all parties in agreement to be patient and understanding.
Limited Access to campus due to Lockdown regulations becoming stronger	Medium	High	Continuously encourage online collaboration and grant assistance where required
Losing data due to poor audio signal processing and noise reduction techniques	Low	Medium	Refine and revise audio signal processing techniques in order to provide a robust solution and good spectral distribution of the audio signal
Losing model data	Low	High	This means having a backup (to intermediate data so as to not waste time in training the model again.

Automatic Music Transcription using Variable Q-Transform and Deep Learning

Sbonelo Mdluli : 1101772, Moshekwa Matthews Malatji : 1387556
Group: 20G04
School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

***Abstract*—Automatic music transcription is the process of converting music signal into musical notes. This paper presents the project plan and milestones used to accomplish the final product. The milestones for the project are; system design, data collection, digital signals processing, model development and testing methodologies. The methodology which will be followed in development of the Automatic Music Transcription is the Cross-Industry Process for Data Mining methodology. This is a structural approach to a data mining project, is adopted in this context to draw knowledge and insight from the dataset. Variable Q-Transform and Deep Learning algorithms are the main concepts to be explored when implementing the transcription system. Music instruments to be transcribed in real time are the piano and drums using the MAESTRO & Groove MIDI dataset, respectively. The Automatic Music Transcription assumes that audio reconstruction and vocal separation is not required and successful transcription the accuracy of the model is required to be at least 50 %. A user interface implemented with PyQt5, Tkinter or MATLAB GUI will provide the user a graphical means of interacting with the AMT system. The project is be completed using an iterative approach in order to refine and improve the model. The project management and planning for the 8 week long project are discussed and the project schedule is registered on the Gantt Chart.**

I. INTRODUCTION

In this paper we present the project plan to be used when implementing an Automatic Music Transcription (AMT) system. The project plan details the project specifications, implementation, testing methodologies and the project management aspect of the investigation. AMT is defined as the design of computational algorithms to convert acoustic music signals into of music notation [1]. This is a process that is of concern to signal processing and artificial intelligence. AMT is mainly implemented using neural networks and non- negative matrix factorization.

The document is structured as follows; Section III outlines the AMT project specifications including the assumptions made, the relevant success criteria and constraints to be adhered to. This is followed by Section II which is a description of the various literature on AMT which is explored in order to derive the proposed methodology. The fundamental approach to the AMT project is discussed on Section IV, it is a structural and procedural approach which has subsequent descriptive sections. An overview of the musical instruments used for music transcription is on Section V, which is followed by the description of the proposed signals processing technique employed for AMT on Section VI. This is followed by a detailed discussion of the artificial intelligent modelling technique for AMT as illustrated in Section VII. In addition, Section IX & X discusses the Post Processing Unit and Testing methodology for the model, respectively. The penultimate section for this document are Section XI which outlines the techniques employed for building a suitable User Interface for the AMT. The project management overview which also outlines the risk analysis and methods employed to ensure project planning techniques by forms of a Work Breakdown Schedule and Gantt Chart are included on Section XII.

II. BACKGROUND

Various literature on AMT has been explored in deriving in deriving suitable to optimize performance and prioritize efficiency. There has been many approaches to AMT with the a common goal to produce musical notation or score from audio signals using different forms of signals processing and modeling techniques. However, [1] denotes that AMT approaches are classified according to the following categories: Frame level, Note level, Stream level and Notation level. Frame-level transcription refers to the estimation of the pitch and number of notes which are present in a frame.This is usually a common level where AMT transcription occurs such as [2] which focuses on multi-pitch estimation of piano sounds using Probabilistic Spectral Smoothness Principle. Furthermore [3] also focuses on

spectral and temporal representations for multi-pitch estimation of polyphonic music, other literature[4] employs the Bayesian methods. Note level or note tracking is similar to the frame level transcription with the addition of connecting pitch estimates into notes. This level of transcription is often incorporates note tracking of the three music notes elements: pitch,onset time, and offset time[1] and pitch estimates of each frame. Median filtering[3] is an example of literature where transcription is classified to be on this level. As the level of transcription increases from the frame level to the stream level, the complexity and degree of transcription also increases as this introduces more musical variables. Stream level transcription focuses on classification of estimated pitches and notes into streams that correspond to certain musical instruments or voices[1]. The work done in [5][6] involves estimation of musical frames which includes pitch and notes and clustering them into different streams/sources.

The AMT approach presented in this paper is within the Notation Transcription Level which is focused on transcribing music into music scores which are readable by humans. Furthermore, this level of transcription focused on digital signal processing and artificial intelligence approaches to successfully transcribe audio signals into music scores. A common approach is obtain a time-frequency transform of audio signals and applying modeling techniques, but the work done in [7][8] only applies neural networks to audio signals in generating a music score. However, literature[9][10] employ the Constant Q-Transform(CQT)[11] audio signals technique to obtain a spectral representation by form of spectrograms. This technique is commonly used as it shows a much better spectral resolution at low frequencies which outperforms the Discrete Time Transforms employed in [12]. This paper proposed the AMT using the VQT and Deep Learning methods. The proposed digital signals processing technique used to obtain spectral frequency analysis and it is is a modification of the CQT, aimed at providing a well defined spectral analysis at low frequencies. The scope of the document is a description of the AMT process including the development methodologies inspired by the literature explored in this section.

III. Project Specifications

1) Assumptions: Assumptions have been made to reduce the scope and complexity while still satisfying the success criteria of the AMT project.It is assumed that the AMT will only be employed on monophonic audio signals instead of polyphonic. The audio signals from the dataset are assumed to have no vocals, thus eliminating the need to implement vocal separation techniques. The last assumptions denotes that the scope of AMT will not be concerned with audio signals reconstruction.

2) Success Criteria: The AMT is required to transcribe 2 instruments in real time. A user will interact with the system through a Graphical User Interface. For a successful transcription the accuracy of the model is required to be at least 50 %.

3) Constraints: The computational power of the machine will dictate the speed at which the models train. The quality of the data source also plays a role in the accuracy of the model. Well labelled and noise reduced music is required in order to improve the model accuracy.

IV. System Design & Methodology Overview

The proposed AMT methodology or approach in development of the AMT has to be precise and procedural in-order to achieve successful implementation. Noting that the design and implementation of the Automatic Music Transcription is a software programming, data science problem within an Electrical Engineering context. Therefore the methodology in development has to be meticulous and precise such that it accounts for Software and Data Science concepts while adhering to Electrical Engineering practices(emphasis on Digital Signals Processing) .The methodology which will be followed in development of the Automatic Music Transcription is the Cross-Industry Process for Data Mining methodology(CRISP-DM) [13]. This is a structural approach to a data mining project, is adopted in this context to draw knowledge and insight from the obtained music dataset. Figure 1 provides a sequential phases of task to provide a robust, well-defined and accurate output for Automatic Music Transcription.

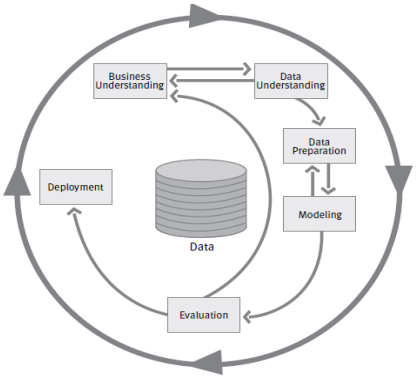

Fig. 1: The CRISP-DM Methodology

A. Business Understanding

Business Understanding is the most important phase in carrying out AMT as this is the skeleton which defines the fundamental processes that lay a foundation for the successive phases. This phase focuses on outlining the project specifications, objectives, assumptions, constraints to be adhered to and the relevant success criteria for successful Automatic Music Transcription. Furthermore, production of a project plan which discusses tools and techniques to be employed and assessing various contingencies i.e. conducting a Risk Assessment are crucial tasks during this phase.

B. Data Understanding

The second phase which is Data Understanding is more involved with the interactions with the data that will be used during AMT. This phase begins with the 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. For AMT, the dataset is assumed to be monophonic, therefore this phase focuses on accounting for such variables.

C. Data Preparation

The Data Preparation focuses on the tasks involved in preparation of the data for modeling techniques and tools to be used. This requires the processing and transformation of the raw data such that it is adequate for modeling. The tasks responsible for transformation and processing the data do not occur in any order as they may be repeated multiple times. For the purpose of AMT, this phase will consist of transformation of raw monophonic music dataset to produce spectrograms using VQT for modeling techniques.

D. Modeling

This phase focuses on selecting and applying modeling techniques on the transformed data from the preceding phase. Noting that there are several modeling techniques which are employed on the transformed data, the experimentation often requires iterating between data preparation and modeling. The modeling of the dataset is not enough for a robust and resolute output therefore other tasks involved include generating tests and possible model refinement techniques. The spectrograms resulting from VQT will be used as input fro the proposed model in Section VII.

E. Evaluation & Deployment

Evaluation is the preliminary phase of the AMT project at the point where the modeling would have been completed. It will focus on reviewing and assessing the results of the overall AMT process and verify if they are a reflection of the specifications, objectives, constraints and success criteria. An important task in this phase is conducting a critical analysis on the results from the preceding phases of experimentation. Deployment is the final phase of the life-cycle for the AMT process. Noting obtaining data preparation, modeling and evaluation are not the last stages of the AMT process. It is imperative that the results are organized in professional and scientific form, this will require documentation and a presentation. The deployment phase will also outline a review of the entire process to focus on reviewing the process outline future recommendations to the scientific flaws and shortcomings encountered.

V. Datasets

Music transcription can be employed to many different instruments. 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.
Drums are profound for their versatility in music as they are not confined to one genre. They are important as they can adapt to what music requires while producing unlimited tonal, melodic, rhythmic and harmonic shading. Drum sets have often been employed in genres such as Rock 'N' Roll, Jazz and the Blues. The Groove MIDI Dataset [14] is to be used for the automatic drum transcription. The dataset is composed of 1,150 MIDI files and over 22,000 measures of drumming including 13.6 hours of MIDI and audio human-performed, tempo-aligned drumming. Furthermore, the dataset was performed by 10 professional drummers with Roland TD-11 electronic drum kit who were inspired to be versatile and experiment with a wide range of playing styles to ensure the dataset is diverse.

The piano belongs to the keyboard family of musical instruments with stuck strings, and it offers a range of all 88 notes of the music scale which stands out from most instruments. Another interesting piano feature is that when a note is played, the musician has an option of releasing the key or playing it again while the note is still active [15]. MAESTRO (MIDI and Audio Edited for Synchronous Tracks and Organization) [16] dataset is employed for automatic piano transcription. This is a raw dataset contains over 200 hours of paired audio and recorded MIDI data from performances by virtuoso

pianists perform on Yamaha Disklaviers in the International Piano-e-Competition. Furthermore, The MIDI Data and paired audio are aligned by approximately 3ms. The Groove MIDI Dataset drum and the Maestro piano datasets both consists of recorded MIDI data from performances by professional drummers and pianists, respectively. The involvement of professional musicians in the creation of the dataset brings into question the veracity, accuracy and precision of the recordings. As stated by [17], It can be argued that a high level of accuracy can be guaranteed in the alignment of audio and MIDI data if the dataset were created by automated self-playing instruments regulated by a MIDI signal. However, discrepancies are bound to occur with the automated recordings as the Yamaha Disklavier incorrectly plays notes when the MIDI velocity decreases, as [18] reports up to 100ms in audio and MIDI data alignment errors due to automated recording.

VI. Digital Signals Processing

A. Audio Signal Representation

The Automatic Music Transcription of the Drums and Piano begins with the digital processing of audio signals. In general, signals are abstract therefore we model them mathematically to study their behaviour in the time and frequency domain. 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. A spectrogram is 3D matrix which represents frequencies of a signal in variation with time[15], and different frequency magnitudes are represented in a variety of colors. The proposed digital signals technique used to obtain spectrograms is the VQT which is a modification of the Constant CQT by introducing a parameter γ [19]. Therefore, it is imperative to have a primitive understanding of the CQT to draw the distinction its from the VQT for Automatic Music Transcription.

The Constant Q-Transform is audio signals processing technique which is suited for music transcription[15] is as it offers well defined low frequency spectral representation rather than at high frequencies with the added benefit of efficiency[20]. The CQT is interpreted as a filter-bank, where the filters banks are geometrically spaced by centre frequency (f_k) as indicated in Equation 2, and f_{min} is defined as the first center frequency. The bandwidth of the k_{th} filter is defined by Equation 1

$$B_k = 2^{\frac{k}{n}} B_{min} \tag{1}$$

where n is the number of octaves per filter and B_{min} is the first bandwidth of the first filter bank. Noting that the centre frequencies(f_k) of the filters are geometrically spaced, the centre frequency of the k_{th} filter is given by Equation 2

$$f_k = f_{min} 2^{\frac{k}{n}}; k = 0, 1, ... \tag{2}$$

where f_{min} is the first centre frequency and n is the number of octaves per filter. The CQT weights have constant equal Quality-factors(Q) which can be represented as a ratio of the centre frequency(f_k) to the Bandwidth(B_k) in Equation 3

$$Q = \frac{f_k}{B_k} = \frac{1}{2^{\frac{1}{n}} - 1} \tag{3}$$

Therefore, the window size(*N*) of each k filter can be represented in terms of the Q-factor as shown by Equation 4, where f_s is the sampling frequency which is defined in Section VI-A.

$$N[k] = \frac{f_s}{B_k} = Q\frac{f_s}{f_k}; k = 0, 1, ... \tag{4}$$

As the stated before, The VQT is a modification of the CQT with the aim of producing a much better spectral representation for AMT. The modification of CQT into VQT occurs with the introduction of the parameter γ to maintain the a constant Q-factor and decreases it at high and low frequencies, respectively. Furthermore, for CQT representation, $\gamma = 0$, and VQT ,$\gamma > 0$ and its value varies depending on the application, noting that large values of γ significantly increase the time resolution at lower frequencies[21]. Figure 2 and Figure 3 are CQT and VQT spectrograms resulting from the experimentation in [22]. A comparison between the spectograms denote a much better spectral resolution at low frequencies for the VQT representation.

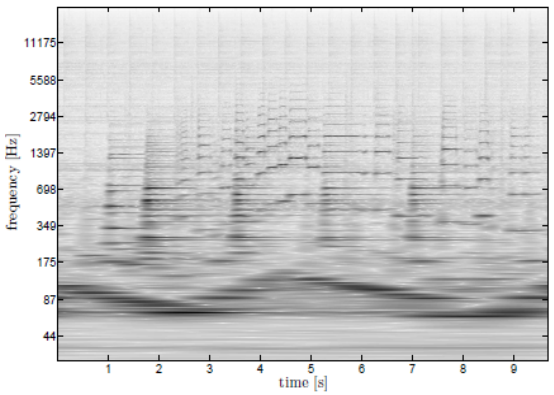

Fig. 2: Constant Q-Transform spectrogram $\gamma = 0$

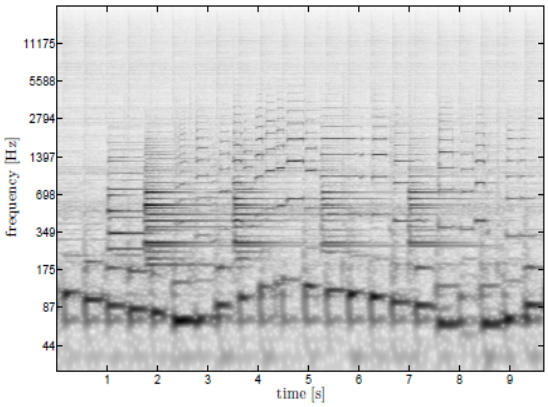

Fig. 3: Variable Q-Transform spectrogram $\gamma = 20$

The bandwidth(B_k) for VQT time-frequency analysis is given by Equation 5, with the additional parameter γ, the centre frequency(f_k) for the k_{th} filter and n the number of octaves per filter.

$$B_k = \alpha f_k + \gamma \tag{5}$$

where

$$\alpha = 2^{\frac{1}{n}} - 2^{-\frac{1}{n}}$$

The Quality factor(Q) can also be represented in terms of γ for the VQT representation using Equation 6 below:

$$Q = \frac{f_k}{(2^{\frac{1}{n}} - 1)f_k + \gamma} \tag{6}$$

The computation of VQT to produce the spectrograms will be achieved using `Jupyter Notebook` & `Librosa`[23] which is a Python package for audio and music digital signal processing. `Librosa` contains predefined computation of CQT which will be modified to introduce γ for VQT representation. However, some parameters require to be modified and redefined for the VQT computations.

Parameters:

1) f_s: Sampling rate/frequency of each filter of window size N such that the Q-factor is satisfied. The sampling frequency is 22050Hz [23].
2) n_{bins} - number of frequency filter banks which determines the maximum frequency of the VQT.
3) n: number of filter banks per octave which influence the frequency resolution of the VQT. Typical values are 12, 24, 36 & 48, however, a suitable on will have to selected during experimentation[15].
4) N(Window Size): the resolution of the VQT spectrogram is dependent on the window size. A variable window size as described by Equation 4 ensures is ideal to obtain high spectral resolution at lower frequencies and high temporal resolution at high frequencies [24].
5) f_{min}(minimum frequency): the minimum frequency will be the first center frequency of the filter bank. According to [15] it should be as low as the frequency of the first notes such that the spectrogram should show the fundamental frequency and the neural networks recognizes the notes.

VII. Proposed Model

The proposed model is based on the work by [25],[19],[26] and the Kelz architecture [15][27]. The model consist of sub tasks which perform different tasks in the training stage namely; onset, offset, velocity and frame prediction as depicted in figure 4. Different machine learning libraries will be explored for the implementation of the model `PyTorch, Keras, TensorFlow` and `matlab machine learning toolbox`. Ultimately the library with the most support, easy of use and less development time will be selected.

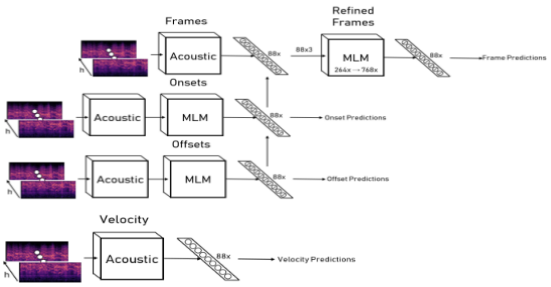

Fig. 4: Transcription model, where h represents the total number of frames

A. Acoustic model

The model consists of a Convolutional Neural Network (CNN) and a post processing unit which is to be determined through experiments. The spectograms produced by VQT are used as input images to the CNN. The spectograms have dimensions $t \times f_s$ where t is the duration of the song in seconds and f_s is frequency (Hz). Unlike normal object images, music is an ordered sequence which means spectograms cannot be transformed to different configurations. The first operation to be done is to convolve a region of the spectrogram with a filter. The filter is a matrix of weights. In this context convolution is an element wise dot product between the considered region and the filter. Each filter has a

corresponding bias matrix which is summed with the convolution operation result to give the output from the convolution layer. The filter and biases are then optimised with each forward pass. The filter shape is structured such that is captures maximal note information (fundamental frequency and corresponding harmonics), as such the filter shape for the proposed model spans the frequency axis and has stride equal to the tone duration [28].

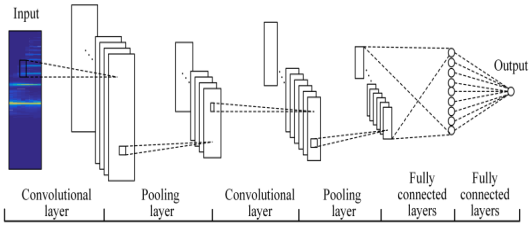

Fig. 5: Acoustic model

An activation layer is introduced between the convolutional layer and the pooling layer. This layer is added to decrease over-fitting between features by introducing non linearity. The activation function used is the Rectified Linear Unit (ReLu) to produce and activation map. The advantage of using this function compared to activation functions such as sigmoid and tanh is that the ReLu converges faster and does not require input normalization because the output is guaranteed to be between 0 and the maximum value x. The ReLu function is defined as:

$$relu(x) = max(0, x) \tag{7}$$

A pooling layer is applied to perform a non-linear down sampling of the activation map. The pooling performed should be such that it reduces the activation map whilst maximising information gain. Usually pooling is done through a statistical parameter namely the average or maximum value. Max pooling deemed more appropriate for this application compared to average pooling. This is because max pooling ensures that the dominant feature is recorded, the problem with average pooling is that the average might be feature that is not dominant in that region. This is important for this application because we want to capture the most dominant note per frame.

Another method to avoid over-fitting is to introduce a drop out which randomly disconnects neurons between layers. This layer is added in between the pooling layer and activation layer. This step is only introduced in the training stage.

A binary cross-entropy loss function is used to measure the difference between the predicted values $\hat{y_n}$ and ground truth y_n. The loss function is defined below, where N represents the number of labels.

$$L = \frac{1}{N}\sum_{n=0}^{N} y_n log(\hat{y_n}) + (1 - y_n) log(1 - \hat{y_n}) \tag{8}$$

The loss function is continuously optimised through gradient descent and is used to update the weights and biases through back propagation until it reaches diminishing results.

VIII. Acoustic Topology

The topology presented is a first iteration approximation design of the training model. Experiments are going to be carried out in order to determine the most optimal parameters.

A. Filter

Various filter sizes will be evaluated those presented in literature and others that appropriate for the selected instruments. Filter sizes to be explored from literature are $\{f_s \times window\_size/2k, 3 \times 3, 5 \times 5\}$ where k is the number of bins per octave and $window\_size$ being the time slice.

B. Learning Parameters

The learning rate is an important parameter used during back propagation in gradient descent. It specifies the amount by which the network weights from be updated. Setting the parameter too low may result in the taking too long to converge to a local minima and setting it too high may result in the model never converging because it would have missed the local minima. It is therefore important to increment this parameter at a reasonable rate.

Dropout is a less computational demanding form of regularisation. Different keep rates used in literature for the dropout layer are to explored specifically $\{0.2, 0.25, 0.5\}$. A threshold value of 0.5 is used for the activation function, where $\hat{y_n} = y_n > 0.5$.

IX. Post processing unit

The MLM is appended at the end of each acoustic model in order to smooth out the output and capture temporal dependencies between frames such as how notes evolve over time. The output from acoustic model is be used as the input for the MLM with the exception velocity prediction. Prediction velocity is a standalone task, velocity takes into account the speed and loudness of a note [29]. The MLM can be implemented using different

techniques namely Recurrent Neural Network (RNN), Bidirectional Long-Short-Term Memory (BiLSTM) or Hidden Markov model (HMM) [30]. The BiLSTM has been shown to be superior compared to the RNN and HMM, nonetheless different variations of the MLM will still be evaluated to get the optimal one. The idea is to follow the Google Brains Onset and Frames Network and Kelz baseline model whereby the results from the sub tasks are appended together in order to get the final frame prediction as depicted in figure 4 In this model a ReLu is after the BiLSTM instead sigmoid in mentioned models.

X. Testing and validation process

The testing stage is used to optimise and fine tune model hyper parameters in order to improve accuracy and performance. The testing stage includes using unseen data in the training stage. This prevent the model to being over fit to a particular dataset.

The model will probably need to be on the trained cloud because of the complex and costly training process. The training process also needs a lot of data which may not be feasible stored on a single machine. Each song is split into defined window sizes and produce a spectrogram for each window. Fine tuning the model may also be time consuming task. Possible cloud providers are google, aws and azure which mainly provide a Python SDK.

A. Metrics

The confusion matrix, word error rate, character error rate, precision(P), recall(R) and f-measures(F1) are some of the common performance metrics used for machine learning models. The metrics to be used to evaluate the model are precision, recall and f-measure. This step also makes use of new unseen data in both the training and testing stage. F1 is a ratio of the total number of errors compared to the number of detectable notes in the music. The equations below are used to calculate the mentioned metrics.

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$F1 = \frac{2RP}{R + P} \tag{11}$$

Where TP, FP and FN represent true positive, false positive and false negative respectively. The `mir_eval` library provides and implementation to evaluate the stated metrics [31]. The onset and offset predictions are considered a TP within a ± 50 ms tolerance of the ground truth. Each frame correctness is verified per 10 ms [15].

XI. User interface

The user interface will provide the user a graphical means of interacting with the AMT system. The particular choice will be dictated by the language used to implement the AMT model. PyQt5, Tkinter or matlab gui. Both PyQt5 and Tkinter are Python wrapper which provide a framework for GUI development in Python. The GUI will provide a window in which a user will select a song to be transcribed. The transcription processes will occur in the same window. The output representation will be at a note level.

XII. Project Management

This section outlines the work breakdown assignment and the project schedule. The scope of the AMT system has two main components which digital audio signals processing and modeling using neural networks. This makes the work breakdown schedule of the tasks convenient as each member on the group can focus on one main component at the time while sharing common tasks and constantly helping each other in efforts of improving productivity, team moral and producing an immaculate output. The high level representation of the work breakdown assignment is registered in Table I. The project duration for the AMT project is 8 weeks. The system development will be guided by the project schedule represented by the Gantt chart on Figure 1 in Appendix A. In the Gantt chart the critical tasks are indicated in red, noting that it will define the shortest duration to complete the project. Furthermore, team members are to commit themselves to working from 8am to 5pm weekdays, however, weekends will also be used to review the work done during the week, complete the tasks which were not done and plan/work ahead for the successive week. Prior to the project commencement team members understand that only good teamwork, conflict resolution, constant effective communication by the form of weekly meetings, hard work and determination will guarantee project success. The project will be done in stages in an iterative manner. The project is broken down to consist of milestones to be completed per iteration. The development process consist of design, prototype, test and deployment stages.

TABLE I: Suggested Work Breakdown

Tasks	**Sbonelo**	**Matthews**
AMT research, specifications and planning	X	X
Dataset Collection		X
Digital Signals Processing		X
Modelling, Testing & Validation	X	
User Interface	X	X
Documentation & Presentation	X	X

The analysis of the contingencies that might occur is within the scope of the fundamental stages of the CRISP-DM methodology described in SectionIV. The risk assessment is registered on Table II in Appendix B.

XIII. Conclusion

In this paper we presented the project plan outlines for the AMT system. The system consists of multitude of tasks which form the complete product. The system is designed to transcribe 2 musical instruments viz. the Drum and Piano using the he GrooveMIDI and the MAESTRO Datasets, respectively. From the literature explored, The Variable Q-Transform outperforms the Constant Q-Transform hence it is the preferred audio signals processing technique to produce spectrograms which are fed into neural networks.The proposed model consist of an acoustic model and MLM unit. These are the essential parts used in training the model. A CNN is used for the acoustic model and HMM, BiLSTM or RNN are some of the options to be explored for the MLM.

REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] 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, 2010.

[3] L. Su and Y. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 10, pp. 1600–1612, 2015.

[4] P. H. Peeling, A. T. Cemgil, and S. J. Godsill, "Generative spectrogram factorization models for polyphonic piano transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 519–527, 2010.

[5] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 138–150, 2014.

[6] V. Arora and L. Behera, "Multiple f0 estimation and source clustering of polyphonic music audio using plca and hmrfs," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 278–287, 2015.

[7] M. Bereket, "An ai approach to automatic natural music transcription," 2017.

[8] R. G. C. Carvalho and P. Smaragdis, "Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 151–155.

[9] S. Dubnov, "Unified view of prediction and repetition structure in audio signals with application to interest point detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 327–337, 2008.

[10] R. A. Dobre and C. Negrescu, "Automatic music transcription software based on constant q transform," in *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2016, pp. 1–4.

[11] C. Schörkhuber, "Constant-q transform toolbox for music processing," 2010.

[12] C. Marghescu and A. Drumea, "Modelling and simulation of energy harvesting with solar cell," 02 2015, p. 92582L.

[13] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 01 2000.

[14] J. Gillick, A. Roberts, J. Engel, D. Eck, and D. Bamman, "Learning to groove with inverse sequence transformations," in *International Conference on Machine Learning (ICML)*, 2019.

[15] M. Karioun and S. Tihon, "Deep learning in automatic piano transcription."

[16] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[17] A. C. Jaedicke, "Improving polyphonic piano transcription using deep residual learning," June, 2019.

[18] S. Ewert and M. Sandler, "Piano transcription in the studio using an extensible alternating directions framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1983–1997, 2016.

[19] Cwitkowitz and C. Frank, "End-to-end music transcription using fine-tuned variable-q filterbanks," 2019.

[20] R. A. Dobre and C. Negrescu, "Automatic music transcription software based on constant q transform," in *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2016, pp. 1–4.

[21] E. Benetos and S. Dixon, "Multiple-f0 estimation and note tracking for mirex 2012 using a shift-invariant latent variable model."

[22] C. Schrkhuber, A. Klapuri, N. Holighaus, and M. Drfler, "A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, Jan 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=17112

[23] C. R. D. L. D. P. E. M. M. E. B. McFee, Brian and O. Nieto, "librosa: Audio and music signal analysis in python," in *In Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[24] K. O. U. T. M. Conforto Silvia, Nisar Shibli, "An efficient adaptive window size selection method for improving spectrogram visualization," in *Computational Intelligence and Neuroscience*. Hindawi Publishing Corporation, 2016. [Online]. Available: https://doi.org/10.1155/2016/6172453

[25] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic music transcription," *ArXiv*, vol. abs/1508.01774, 2015.

[26] M. Mnguez Carretero, "Automatic music transcription using neural networks," 2018-07-02.

[27] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," *CoRR*, vol. abs/1612.05153, 2016. [Online]. Available: http://arxiv.org/abs/1612.05153

[28] J. Sleep, "Automatic music transcription with convolutional neural networks using intuitive filter shapes," 2017.

[29] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," *CoRR*, vol. abs/1710.11153, 2017. [Online]. Available: http://arxiv.org/abs/1710.11153

[30] B. S. Gowrishankar and N. U. Bhajantri, "An exhaustive review of automatic music transcription techniques: Survey of music transcription techniques," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016, pp. 140–152.

[31] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common mir metrics," in *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer, 2014.

APPENDIX

A. *Project Schedule*

Fig. 1: Gantt Chart

B. Risk Management

The table below indicated the risk and its associated attributes. The risk register is used to enure that the project is delivered will less hindrance.

TABLE II: Risk register

Risk	**Probability**	**Impact**	**Response**	**Action**
Interruption of training process	High	High	Avoid	Ensure machine is always charged
Data loss	High	High	Avoid	Use redundancy (cloud storage and external hard drive)
Computational intensive training	High	High	Mitigate	Cloud training or model refinement
Conflict	Low	High	Solve	Effective communication, Confront the conflict & find common solution

EIE Laboratory Project Weekly Meeting – Software (Machine Learning)

Monday 31st August 2020 **Chair: Alice Drozdov**

Meeting began at 11:04.

Participants

Groups	***Supervisors***
Group 04 - Moshekwa Malatji, Sbonelo Mdluli	Alice Yang
Group 10 - Siphiwe Cilo, Nomthandazo Mpanza	Craig Carlson
Group 12 - Michael Asamoah-bekoe, Madimetja Sethosa	Ellen De Mello Koch
Group 18 - Jonathan Meerholz, Mihir Vanmali	Estelle Trengove
Group 28 - Faheem Moolla, Shayaan Salim	Mitchell Cox
Group 33 - Tebogo Nkomondo, Ntsatsi Thubakgale	Olutayo Oyerinde
Group 37 – Rian Breytenbach, Kelly Margalit	Steven Dinger
Group 50 - Alice Drozdov, Shaw Chian Lin	

Absentee
none

Agenda

1. **General Issues**

Prof. Estelle Trengove reiterated the importance of social distancing; masks must be worn for cases where social distancing may not be practical with project partners. Any students not following the protocol will have their access to campus revoked.

2. **Updates/Feedback**

Group 50 *(Machine Learning for Optical Mode Detection)* - worked on the optical (interferometry) setup and successfully generated an interference pattern. Pre-existing neural network architectures (ResNet, GoogLeNet etc.) were trained using simulated patterns. Turbulence will be worked on once the interference patterns are sampled as datasets. Less complex neural networks will be designed and experimented with.

Group 12 *(Virtual Assistant for the Hearing-Impaired Community)* - administration done to change survey platform to Google Forms. Scripts set up to process video into image. Neural network being developed, and simulations tested on Google Collab.

Group 33 *(Real or fake? Detect which tweets about disasters are real or fake using NLP)* - worked on improving base model and pre-processing of dataset (removing punctuations etc.). Primarily focused on metric scoring and neural network accuracy. Used the Python Keras API.

Group 28 *(Validating the Uncanny Valley hypothesis using computer-generated facial images)* – encountered ethics application issues, has been resolved. Primary focus on generating datasets which can be manipulated/reconstructed regarding the Uncanny Valley Hypothesis. Surveys are also being designed in this regard. In a discussion with Craig Carlson, control/reference images will be used as well in the surveys.

EIE Laboratory Project Weekly Meeting–Software (Machine Learning)

Date: Monday 07th September 2020 **Chair:** Michael Asamoah-bekoe

Participants

Group Id	Group Members	Supervisor
20G04	Malatji Moshekwa, Mdluli Sbonelo	Olutayo Oyerinde
20G10	Cilo Siphiwe, Mpanza Nomthandazo	Olutayo Oyerinde
20G12	Asamoah-bekoe Michael, Sethosa Madimetja	Craig Carlson
20G18	Meerholz Jonathan, Vanmali Mihir	Vered Aharonson
20G28	Moolla Faheem, Salim Shayaan	Estelle Trengove
20G33	Nkomondo Tebogo, Thubakgale Ntsatsi	Ellen De Mello Koch
20G37	Breytenbach Rian, Margalit Kelly	Steven Dinger
20G50	Drozdov Alice Lin, Shaw Chian	Mitchell Cox

Absentees
- *none*

The meeting started at 11:03, and was held on MS-Teams.

Agenda

1. General Announcements and issues

A reminder for those working on campus to adhere to the social distancing protocols and to always wear a mask at all times. If you have problems with access on campus or ordering components, do let your supervisor know. And lastly, all groups to plan their work around the load-shedding schedule to make sure that they are not affected.

2. Group Updates (Week 2)

- ***Group 12 (Virtual Assistant for the Hearing-Impaired Community)***

Modified the media-pipe hand tracking framework in order to display only the detected landmarks and not the palm detection border. Managed to get 2d feature extraction for both approaches. 1. Using media-pipe 2. Using CNN pre-trained model. Currently working on formatting the output from the feature extraction, thus that they can link them to an RNN model to extract temporal features. Got their ethics consent approved for using Google forms and also started with preliminary data collection.

- ***Group 33 (Real or fake? Detect which tweets about disasters are real or fake using NLP)***

Managed to label tweets as natural disaster or not. Got three working models, using cloud data and their own data set the models performed relatively the same, the focus was on hyper parameter training (using pre-trained embedding layer vs Keras embedding layer). Also found out that LSTM

EIE Laboratory Project Weekly Meeting–Software (Machine Learning)

Date: 14 September 2020 **Chair:** Tebogo Nkomondo

Participants

Group Id	Group Members	Supervisor
20G04	Malatji Moshekwa, Mdluli Sbonelo	Olutayo Oyerinde
20G10	Cilo Siphiwe, Mpanza Nomthandazo	Olutayo Oyerinde
20G12	Asamoah-bekoe Michael, Sethosa Madimetja	Craig Carlson
20G18	Meerholz Jonathan, Vanmali Mihir	Vered Aharonson
20G28	Moolla Faheem, Salim Shayaan	Estelle Trengove
20G33	Nkomondo Tebogo, Thubakgale Ntsatsi	Ellen De Mello Koch
20G37	Breytenbach Rian, Margalit Kelly	Steven Dinger
20G50	Drozdov Alice Lin, Shaw Chian	Mitchell Cox

Absentees
- none

Apologies
- Prof Estelle Trengove (attended another meeting)

The meeting started at 11:02, and was held on MS-Teams.

Agenda

1. General Announcements and Issues

A reminder that the deadline for project title changes is on 25th September, EIE Open Day is on the 8th October, and the final report submission is on 16th October at 12pm and the conference and interviews will be held between 21-23 October 2020. When going to campus make sure to wear your mask at all times, keep social distancing, complete a screening questionnaire using Screening App (LogBox Patient) which is available from App Store (iOS) or Google Play Store (Android) and make sure to have a letter permitting you to gain access (you may find this letter in your emails).

2. Group Updates (Week 3)

- **Group 33 (Real or fake? Detect which tweets about disasters are real or fake using NLP)**

Investigated different pre-trained embedding schemes (Glove and Word2vec) and used own word2vec model which was trained using given dataset. Optimized own word2vec model using different number of iterations and window sizes and managed to combine best performing word2vec model with SimpleRNN and LSTM NN. Analysed location feature and found that it could have a great impact in determining whether tweets about disasters are real or not.

- **Group 04 (Automated Music Transcriber for two Musical Instruments)**

Able to train classification model which classifies two musical instruments (drums and piano). VGG16 model was used and gave performance accuracy of 96%. Model trained using few epochs as it takes time to train and more investigation will be done to optimize this. Need to complete the spectrograms for both the piano and drums and integrate this with one-hot encoding in order to get the classification model working. Currently working on signal framing as different signal frames are required to get a spectrogram and will be investigating other techniques to use as well.

- **Group 10 (Automatic African Music Genre Classification)**

Managed to extract all the features needed for training and audio was converted into images and also got spectrograms sorted out. Models were trained using five different sci-kit learn algorithms including KNN, SVM, RandomForest and decision tree regression. Used ensemble classifier to generate confusion matrix using the three best performing models in terms of accuracy. This week CNN model will be used together with images generated in previous weeks.

- **Group 12 (Virtual Assistant for the Hearing-Impaired Community)**

Finished linking CNN for spatial features and RNN for temporal features. Due to lack of enough data, dataset from UFC 101 for action recognition was used and the model was tested and gave accuracy of just over 85%. Started collecting data and have sent participation forms to few people. Currently collecting dataset from a-z in terms of SA sign languages. Collected 100 videos per each letter and storing that in Google drive. Need to delete few frames in some videos in order to get the correct gesture. Included a README note to instruct the participants on how to take the video and how to send it (in order to maintain video quality).

- **Group 18 (De-localised Object Motion using Deep Learning with Artificial Data Generation, with Focus on Traffic Tracking)**

Started doing multi-vehicle tracking and are able to get actual positions of the vehicles. Currently working perfectly on a straight-line road. Will be working on some optimization techniques to see which one is better. Managed to fix blender issues, now able to work with Matlab code and are able to get camera parameters. Model was used for vehicle moving in one lane, two vehicles passing one another. Will be looking to get the vehicle tracking to work in different scenarios like traffic circles and hills.

- **Group 28 (Validating the Uncanny Valley Hypothesis using Computer-generated Facial Images)**

Got the latent vector (of images) approximation last week and tried to get the approximation to the original image as close as possible. Currently working on attribute classifier where images are slightly changed to uncanny. Able to change attributes like age, smile, pose etc as already got pre-trained weights for those dimensions. Will need to get dataset for pre-trained weights for changing things like eye size, shape of the eye, or nose size. Data collection to be finalized by next week Monday.

- **Group 37 (Automated Morphological Classification of Mosquitoes using Artificial Intelligence)**

Worked on pupae gender classifier to see which models performed better. GoogleNet seemed to perform better than ResNet Large(quick and accurate). When model misclassify, it only misclassify male pupae as female pupae. Worked on adult mosquito classification and did well in gender classification and will be looking into species classification this week. Planning to deploy GoogleNet onto a web server and capture more data at NICD as well and do more training.

- **Group G50 (Machine Learning for Optical Mode Detection)**

Got picture of actual mode and was able to add turbulence. Will be working on taking actual pictures of beams with turbulence. Managed to reduce complexity of the resonate architectures. Will be looking to take more pictures and do proper testing this week.

3. Additional Comments and Announcements

A reminder that we are halfway through and should be looking at how to manage our time properly. Keep track of submission dates, keep social distancing and wear mask.

4. Group to Chair the Next Meeting

Group G10
Cilo Siphiwe, Mpanza Nomthandazo
Date: 21 September 2020

Meeting ended at 11:42

Minutes submitted by
Ntsatsi Thubakgale

EIE Laboratory Project Weekly Meeting–Software (Machine Learning)

Date: 21 September 2020 **Chair:** Siphiwe Cilo

Participants

Group ID	Group Members	Supervisor
20G04	Malatji Moshekwa, Mdluli Sbonelo	Olutayo Oyerinde
20G10	Cilo Siphiwe, Mpanza Nomthandazo	Olutayo Oyerinde
20G12	Asamoah-bekoe Michael, Sethosa Madimetja	Craig Carlson
20G18	Meerholz Jonathan, Vanmali Mihir	Vered Aharonson
20G28	Moolla Faheem, Salim Shayaan	Estelle Trengove
20G33	Nkomondo Tebogo, Thubakgale Ntsatsi	Ellen De Mello Koch
20G37	Breytenbach Rian, Margalit Kelly	Steven Dinger
20G50	Drozdov Alice Lin, Shaw Chian	Mitchell Cox

Absentees
- none

The meeting started at 11:02, on Microsoft Teams.

Agenda

1. **General Announcements and Problems**

   A reminder that we are in the 5th week of the project, so if there are any groups that still do not have components, they must contact their supervisors. There are two and a half weeks left before the Virtual Open Day, details on the Virtual Open Day will be released during the week (21 September – 25 September 2020). Groups should be targeting to be rounding off their projects in the time that is remaining.

2. **Group Progress and Updates**

   - ***Group 20G10 (Automatic African Genre Music Classification)***

     Trained a ResNet with the spectrograms generated during the previous week and achieved an accuracy of 83%. Iterated through the ResNet changing the batch size to observe its impact on the accuracy. Noticed that changing the batch size increases the time it takes the model to train, at a batch size of 20, the model took more than 4 hours to train. Settled for a batch size of 5 which took an hour to generate results. Worked on the web application which will present the model that is being created.

   - ***Group 20G04 (Automated Music Transcriber for two Musical Instruments)***

     Worked on data generation and cleaning. Currently, using test data for the BiLSTM model because data cleaning has not been completed. Also managed to train the model on dummy data and will train the model on the correct data once it has been cleaned. Worked on signal framing which allowed the processing of more data into required frames at a certain sampling rate. Only issue was understanding how the data is stored within the frames of the audio.

   - ***Group 20G12 (Virtual Assistant for the Hearing-Impaired Community)***

     Changed the way the landmark features of the hand were being extracted. The features will now be extracted to a text file and use that for training the model. The

new method has a text file which contains the landmark features of the hand and achieved an accuracy ranging between 60%-100%. The model is currently overfitting because the data set is too small. Also created a prediction function which can predict the input of any video, but the function confuses certain hand gestures. Waiting on participants that signed consent forms but have not sent through their videos. Currently working on implementing a user interface and will be implement text-to-speech if time allows. Requested the participation of meeting attendees to assist with data collection.

- ***Group 20G18 (De-localised Object Motion using Deep Learning with Artificial Data Generation, with Focus on Traffic Tracking)***

  Managed to get interaction between the vehicles and pedestrians with the simulation, also simulated different scenarios to showcase the application and using live tracking. Worked on trying to find a way forward with the project and will deploy the project on a Raspberry Pi and create a real-world application.

- ***Group 20G28 (Validating the Uncanny Valley Hypothesis using Computer-generated Facial Images)***

  Worked on finding the latent space of the images and used it to manipulate the images and altering small facial features. Working on manipulating more images and putting them on the survey. Received an announcement from the committee that clearance has been granted and data collection can be done. Requested participation of the attendees in distributing the link for data collection on their social media networks.

- ***Group 20G33 (Real or fake? Detect which tweets about disasters are real or fake using NLP)***

  Worked on finalising the model to find the best performing one. The best forming model uses the pre-trained Glove embedding model and bidirectional LSTMs it achieves an accuracy of 81%, a precision of 80% and a recall of 70%. Settled with this model because of the balance between precision and recall. Also worked on real-time classification of tweets with Twitter API, currently working on classifying the extracted tweets. Worked on dealing with repeated tweets because of retweets using the location of the original tweet and location of the retweets. Replaced missing location with the top 5 locations in the dataset and computed the distance between the original tweet and the retweet location, however, could not quite complete the feature. Currently working on extracting the location of the tweet in real-time and gathering findings for the report.

- ***Group 20G33 (Automated Morphological Classification of Mosquitoes using Artificial Intelligence)***

  Went back to the NICD for data collection to increase accuracy. Continued the adult classifier using GoogleNet with an accuracy on 66%. NasNetLarge gave an accuracy of 83-84% for the species classifier. Captured images of the pupae and observed the difference the male and a female pupa. Worked on developing the mobile app using MATLAB User Interface. Looking into exporting the trained network as SDK that will run on the web application so that the classification of gender and species can be done instantly after uploading the image. Only issues were with the low accuracy of GoogleNet but rectified by using a different network.

- ***Group 20G33 (Machine Learning for Optical Mode Detection)***

  Collected training data and received promising results. Achieved an accuracy of

90% at a Strehl ratio (a measure of turbulence) of 0.5 with a 6-layer ResNet. Working on completing results to compare and analyse.

3. Additional Comments

A reminder that we are currently in the 5th week of the project. Groups are advised to start working on the structure of their reports and gather final data and results.

Details on the requirements for the Virtual Open Day will most likely be released before Thursday, 24 September 2020 to allow groups enough time to plan and prepare.

4. Next Group to Chair the Meeting

Group 20G04

Malatji Moshekwa and Mdluli Sbonelo

Date of Next Meeting: 28 September 2020

Meeting ended at 11:39

Minutes submitted by:

Nomthandazo Mpanza

EIE Laboratory Project Weekly Meeting – Group 1 Machine Learning

Date: 28 October 2020 **Chair**: Moshekwa Malatji

Participants

Group ID	Group Members	Supervisor
20G04	Malatji Moshekwa, Mdluli Sbonelo	Olutayo Oyerinde
20G10	Cilo Siphiwe, Mpanza Nomthandazo	Olutayo Oyerinde
20G12	Asamoah-bekoe Michael, Sethosa Madimetja	Craig Carlson
20G18	Meerholz Jonathan, Vanmali Mihir	Vered Aharonson
20G28	Moolla Faheem, Salim Shayaan	Estelle Trengove
20G33	Nkomondo Tebogo, Thubakgale Ntsatsi	Ellen De Mello Koch
20G37	Breytenbach Rian, Margalit Kelly	Steven Dinger
20G50	Drozdov Alice Lin, Shaw Chian	Mitchell Cox

Agenda

General Annoucements

Note that virtual open day is on the 8th of October. The traditional conference presentation will be replaced by a video presentation, which must be completed on the 7th of October 12 by mid-day. The final date for project title change is on the 28th of October. The meeting could not be recorded on team due to technical problems.

Group 20G04 (*Automated Music Transcriber for two Musical Instruments)*

- Team worked on data cleaning for the transcription model. Developed the transcription model, which is yet to be trained. Worked on post processing conversion from model output to instrument note mapping. There are issues with running tensorflow due to hardware limitation. Currently working on integrating models to UI.

Group 20G10 *(Automatic African Genre Music Classification)*

- Extracted features and trained ResNet model using spectrograms. Experimented with hyperparameters in order to reduce model-training time. Worked on the web application, which will visualise the genre classification. The audio files are split into segments before being classified. Currently working on exporting the model to use incorporate in the web application.

Group 20G12 *(Virtual Assistant for the Hearing-Impaired Community)*

- Currently busy with video collection. The team has received 28 videos so far, still receiving more videos from participants. Pre-process the videos by removing audio and invert the videos to black and white in order to create more training data. The model achieved a validation accuracy of 53%. The team is currently developing a second model for hand gestures and if time permits a real time prediction and text to speech. Work on improving the models.

Group 20G18 ***(De-localised Object Motion using Deep Learning with Artificial Data Generation, with Focus on Traffic Tracking)***

- The team was able to gather real world video footage. They have issues with their model in moving from simulated data to real data. Compare model performance with MatLab toolbox implementation. The team is working on getting more results and work on presentation

Group 20G28 ***(Validating the Uncanny Valley Hypothesis using Computer-generated Facial Images)***

- The team only got ethical clearance on the 26th of October. They have started with generating uncanny images are awaiting for data from survey, which will be sent out on the 28th of October.

Group 20G33 ***(Real or fake? Detect which tweets about disasters are real or fake using NLP)***

- The team is working on live prediction feature classify tweets in real time. Working on improving current model by including location the current model is overfitting. Problem with determining location from model

Group 20G37 ***(Automated Morphological Classification of Mosquitoes using Artificial Intelligence)***

- The team is working on the desktop application for the project and on collecting the results for the report. The team developed separate models for gender, species and pupae classification. The pupae classifier stops working after packaging the desktop app from MatLab.

Group 20G50 ***(Machine Learning for Optical Mode Detection)***

- Use neural net with multiplexing. Set up script to collect images for the model. Still need to validate turbulence. Trained the model using images and compare results. Still need to train model for multiplexing.

Closing comments

Teams must try to send video before Wednesday to get feedback from supervisors.

Chairs must start meeting recording before the beginning of the meeting.

Meeting ended 11:35

EIE Laboratory Project Weekly Meeting – Group 1 Machine Learning

Date: 5 October 2020 **Chair:** Faheem Moola

Participants

Group ID	Group Members	Supervisor
20G04	Malatji Moshekwa, Mdluli Sbonelo	Olutayo Oyerinde
20G10	Cilo Siphiwe, Mpanza Nomthandazo	Olutayo Oyerinde
20G12	Asamoah-bekoe Michael, Sethosa Madimetja	Craig Carlson
20G18	Meerholz Jonathan, Vanmali Mihir	Vered Aharonson
20G28	Moolla Faheem, Salim Shayaan	Estelle Trengove
20G33	Nkomondo Tebogo, Thubakgale Ntsatsi	Ellen De Mello Koch
20G37	Breytenbach Rian, Margalit Kelly	Steven Dinger
20G50	Drozdov Alice Lin, Shaw Chian	Mitchell Cox

Agenda

General Announcements:

- The video presentation on Wednesday
- Maximum length of 3 minutes for video

Group Updates:

Group 20G28 (Validating the Uncanny Valley Hypothesis using Computer-generated Facial Images)

- The group has sent out survey and are analysing data and have also started their video

Group 20G04 (Automated Music Transcriber for two Musical Instruments)

- Improved classification algorithm
- Changed from VGG 16 to VGG 18
- Improved accuracy
- Have to integrate with UI

Group 20G10 (Automatic African Genre Music Classification)

- Finished the app they were going to export their model into

- Can upload a song and it can classify it into a genre

Group 20G12 (Virtual Assistant for the Hearing-Impaired Community)
- Deleted the signs in dataset that media pipe was not accurately able to place landmarks on
- Trained model that uses media pipe with dataset that was split into different validation set
- Implemented text to speech for their model
- Live model prediction takes some time

Group 20G18 (De-localised Object Motion using Deep Learning with Artificial Data Generation, with Focus on Traffic Tracking)
- Wrapping up and have a complete system
- Added a distinguishing feature

Group 20G33 (Real or fake? Detect which tweets about disasters are real or fake using NLP)
- Have been working on the video
- Will send it to supervisors to make sure its correct

Group 20G37 (Automated Morphological Classification of Mosquitoes using Artificial Intelligence)
- Fixed a bug and have a fully functional app
- Working on the video and collecting results

Group 20G50 (Machine Learning for Optical Mode Detection)
- Finalising their project

Closing comments:
- Reports are due soon, please complete them after the video

Meeting ended: 11:20

Minutes by: Shayaan Salim

E MODEL RESULTS

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional	(None, 400)	332800
dropout (Dropout)	(None, 400)	0
repeat_vector (RepeatVector)	(None, 88, 400)	0
bidirectional_1 (Bidirection	(None, 88, 200)	400800
time_distributed (TimeDistri	(None, 88, 1)	201

Total params: 733,801
Trainable params: 733,801
Non-trainable params: 0

Figure 1 : Transcription model parameters

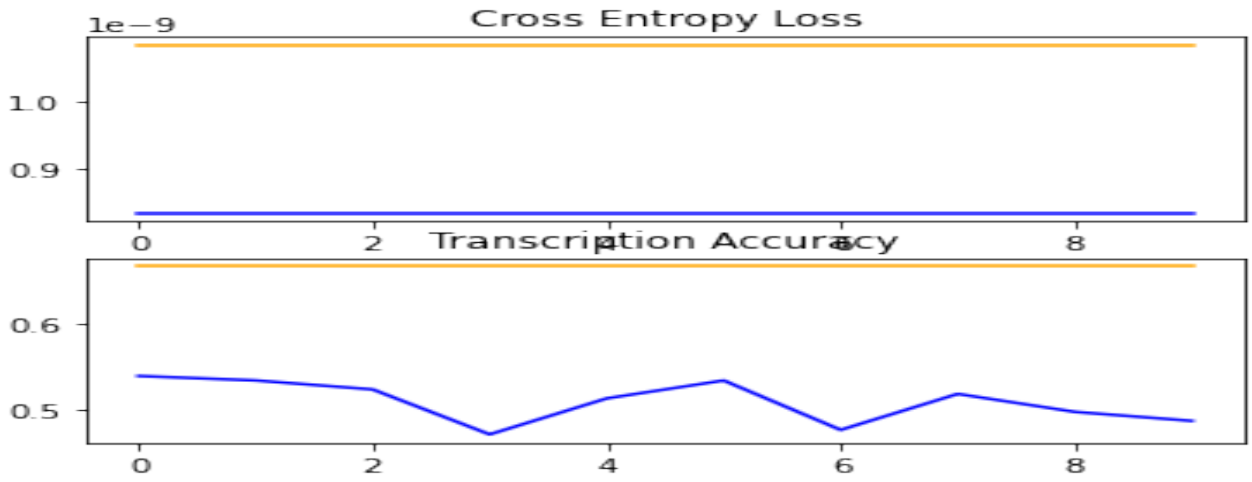

Figure 2 : Transcription model accuracy loss

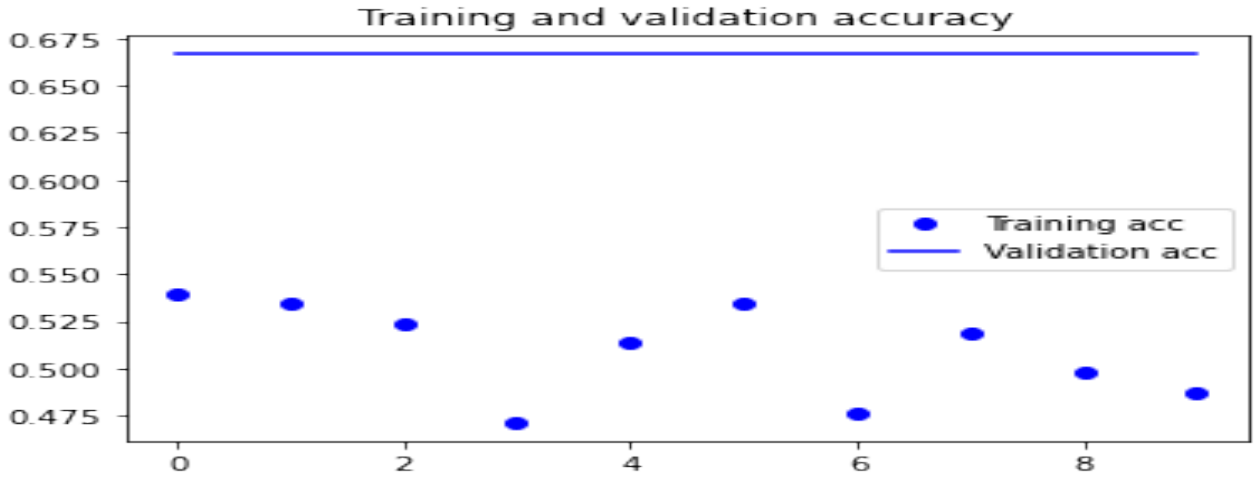

Figure 3 : Transcription model F1-score

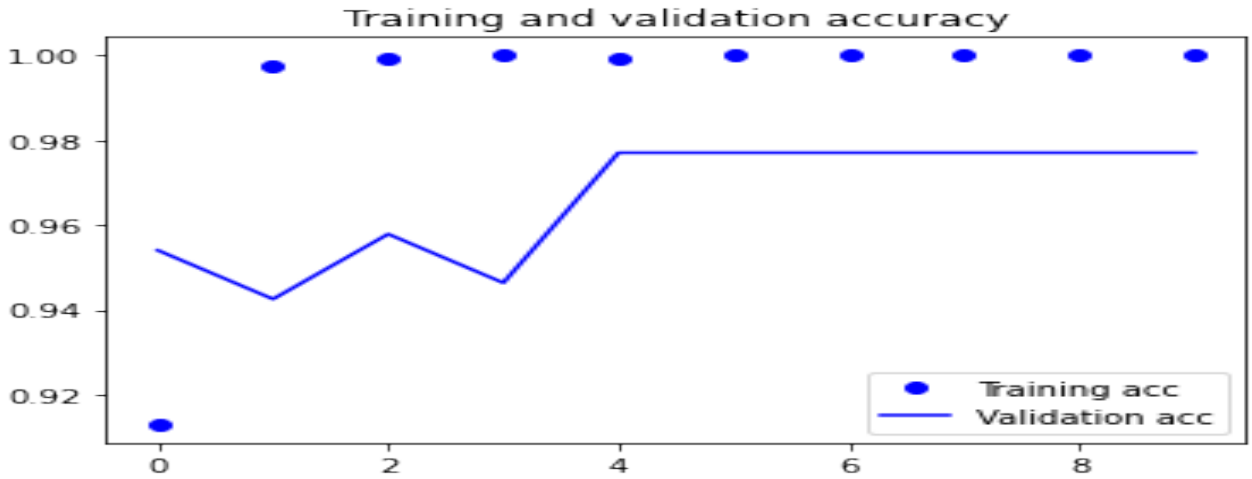

Figure 4 : VGG16 accuracy loss

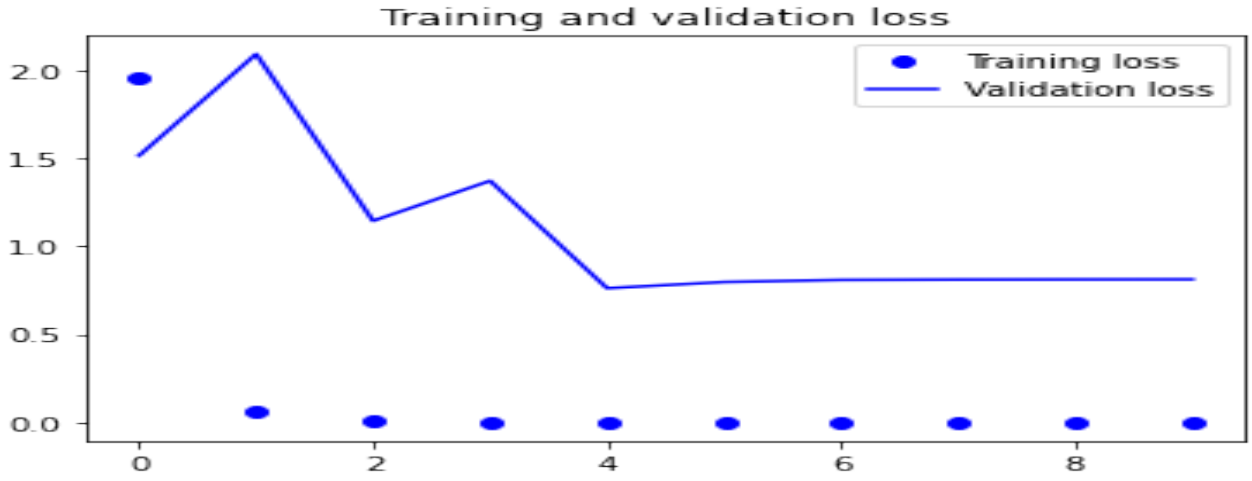

Figure 5 : VGG16 training loss

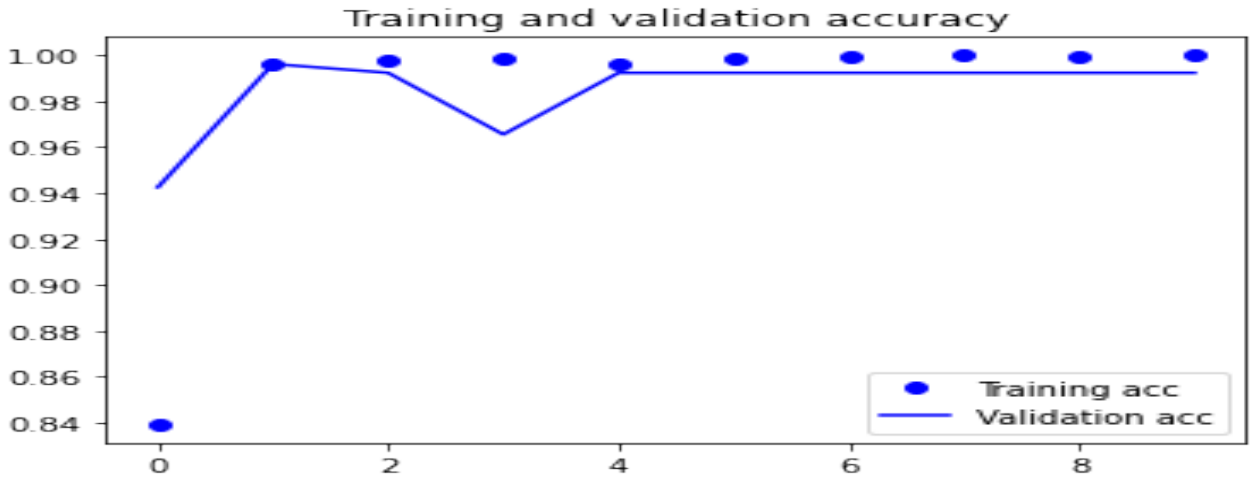

Figure 6 : VGG19 training accuracy

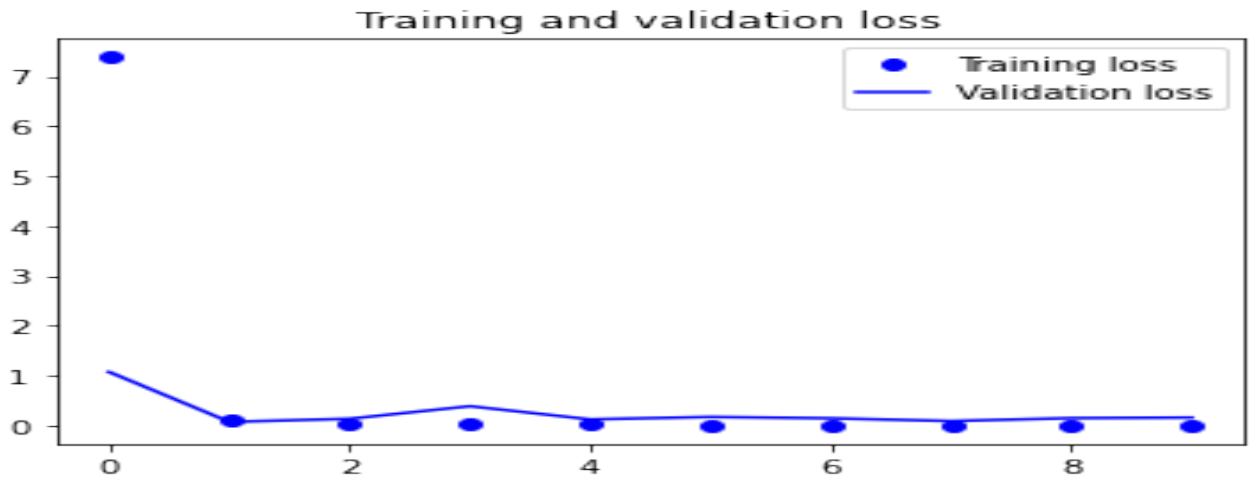

Figure 7 : VGG19 training loss

Figure 8 : User interface in Windows operating systems