Literature Review

# Introduction:

Music is a big part of the modern world. Without knowing it, each of us has a deep connection to music whether one chooses to listen to or make music. Music is a universal language that connects individuals despite their differences. With the rapid advancements being made in the fields of Machine Learning (ML) and Deep Learning (DL), the ability to classify musical instruments from their sounds is useful in various applications in the music industry. Despite showing promising results, ML and DL are still not as robust or efficient as they should be to be implemented in classification models. This literature review focuses on the methods of classifying instruments with machine learning, specifically with using spectrograms to analyse audio files. The different data capturing procedures and algorithms implanted by researchers are examined, as well as performance evaluation methods are investigated. Once relevant insight is gained, a machine learning-based instrument classification tool is developed that may possibly be deployed into the real world.

# Data Collection:

To train a neural network, a large amount of training data is usually required, especially when developing an audio classification task. Researcher may access repositories like GitHub, Kaggle or Hugging Face, which contain many existing datasets that could be sufficient. Alternatively, freesound.org or pixabay.com offer rich reserves of royalty free music suitable to train the model.

Vimal et al. chose Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) to gather diverse vocal data to train a speech model. Their findings emphasise the importance of choosing a data source with a variety of data samples, to ensure that the resulting model is exposed to a wide variety of training data such that it performs better in the real world.

# Feature Extraction:

Feature extraction is a niche field where the method is purely dependant on the type of audio classification task is at hand. Vimal et al. looked at using Mel Frequency Cepstral Coefficient (MFCC) as a method of feature extraction. The reason being that it is best for detection of human voices. Zhang et al. conducted research on environmental sound classification. The audio clips were converted to energy spectrograms by using a short-time Fourier transform (STFT) with a hamming window size dependant on the number of samples and frequency, along with a 50% overlap. In this specific study, the datasets were limited, so the spectrograms were divided into smaller parts with extra information from the original spectrogram being added before being fed into the classification network. Wu et al. also chose to convert their audio data into spectrograms through applying an STFT on windowed audio pieces, as they conducted research into attention augmented CNNs.

Shreevathsa et al. conducted research on music instrument recognition. They use both MFCC and Zero Crossing Rate (ZCR) to extract information of what instruments have been played in an audio sample. ZCR is a common method in audio processing. Both ZCR and MFCC are put together for efficient instrument recognition. MFCC is a feature set that describes an instrument by its spectral shape, like how one would handle the characteristics of a human voice. This is very useful when recognising instrument tones, even though MFCC is useful for human voice recognition. ZCR simply measures the rate of sign changes in the signal, and this can be used to differentiate between positive and negative zero crossings. It is useful in both speech recognition and music instrument retrieval. Essid et al. has conducted research on musical instrument recognition and supports the fact that feature extraction needs to resemble a humans’ perception of music. They experiment with using genetic algorithms and inertia ratio maximisation are suitable for musical instrument recognition. Pairwise feature selection is a different approach which optimises the features for class discrimination, while offering efficiency benefits.

# Machine Learning Classification Methods:

There has been plenty research on different classification methods. Vimal et al. used a range of classification methods, including Random Forest, Decision Trees, and the most relevant and best performing being the Support Vector Machine (SVM). SVM is useful in supervised learning algorithms, which decides on which class a specific pattern belongs to. It is very easy to implement and works well with smaller training datasets. SVM manages to map data to a high dimensional space to make it easy to categorise the data points regardless of their associated linearity. Dhanalakshmi et al. further emphasised that SVMs is fairly accurate when performing general audio classification tasks, with a quoted accuracy of 92%. Classification tasks were also conducted using Radial Basis Function Neural Networks (RBFNN), with an accuracy of 93%. RBFNN is also a supervised learning method, that uses radial basis functions as activation functions since this is a feedforward neural network. Olabanjo et al. quoted a 94% aggregate score for the measure of performance across different classification thresholds.

Zhang et al. compared their method of Auto-Conditioned Recurrent Neural Networks (ACRNN) with existing methods and sees that their method obtained the highest classification accuracy of 93.7%, while other models like multi-stream CNN achieve a similar accuracy when using both raw data and spectrogram information, while ACRNN can with only the spectrograms. ACRNN uses a mix of CNN and bidirectional gated recurrent unit layers (bi-GRU) to compute the features and timing of sounds, including eight layers of convolution and two layers of Bi-GRU, with slight tuning made for effective training and regularisation. This method may be excessive when it comes to datasets with less variation, but useful in real world applications.

Wu et al. proposed a FreqCNN model which combined CNNs and attention mechanisms to learn the features of the input data, i.e., spectrograms. A mix of basic and attention-based convolution blocks split the spectrogram into segments, which allows for the model to examine both local and global variations in features, with an attempt to improving classification accuracy. The model was tested on its ability to handle accent classification, speaker identification, and speech emotion recognition, mostly speech recognition tasks. The proposed model was compared to traditional learning methods. The use of convolution blocks to handle both local frequency areas and global frequency areas may increase the computational accuracy, and in some cases does not improve the accuracy enough to justify implementing it. However, in an instrument classification model, where an instrument may be played at different notes, FreqCNN would make the model more robust for real world applications.

Shreevathsa et al. utilised CNN as their classification models. This is a neural network where all the layers are not fully connected and use convolution rather than regular multiplication. CNN are also known as Shift Invariant Artificial Neural Networks (SIANN), which is a better version of tradition neural networks since it is controllable. The CNNs would extract features from layers in the input data and connects these features to reduce computational requirements. In testing, they found that CNNs outperformed ANNs in audio classification, while reducing the memory and complexity of the neural network. It does take more time to train and test a CNN model in comparison to ANN.

# Performance Evaluation:

Giannoulis et al. reported on their evaluation techniques to test the performance of their instrument recognition model using metrics like precision and recall. The results provide an overview of model performance when different training data is applied with various instruments.

# Conclusion:

Giannoulis et al. <https://doi.org/10.1109/ICASSP.2014.6854599>

Essid et al. https://doi.org/10.1109/TSA.2005.860842

Shreevathsa et al. <https://doi.org/10.1109/ICCAKM46823.2020.9051514>

Vimal et al. <https://doi.org/10.1109/ICCCNT51525.2021.9579881>

Ren et al. <https://doi.org/10.1109/IJCNN.2014.6889383>

Salvati et al. <https://doi-org.ezproxy.uct.ac.za/10.1016/j.eswa.2023.119750>

Copiaco et al. <https://doi.org/10.1109/ISSPIT47144.2019.9001814>

Wu et al.

Zeng et al. <https://doi.org/10.1007/s11042-017-5539-3>

Felipe et al. <https://doi.org/10.1109/SCCC.2017.8405119>

Olabanjo et al. <https://doi-org.ezproxy.uct.ac.za/10.1016/j.tine.2022.100190>

## Introduction:

The aim of this task is to create a small image dataset consisting of spectrograms of three different instruments for classification and try developing a machine learning system to classify the different instruments based on the spectrogram produced from the sounds created by the instruments. Vimal et al. suggested that the way to successfully approach this task is to divide it into 5 main steps that must be conducted in order.

* Collect raw data to train a model.
* Perform feature extraction.
* Implement a machine learning classification method.
* Train the model.
* Test the model.

This procedure will be followed in order to fulfil every requirement of this task.

## Data Collection

The algorithm will be taught to identify three different instruments: snare drum, trumpet, and violin. The sound clips are collected from freesound.org and pixabay.com, which house massive libraries of royalty free music. It is important to collect a variety of datasets that have noise and external disturbances that resemble real world applications. If the sound samples are all recorded in a controlled environment, i.e., in a music studio where background noise is minimised, it would impact the performance of the model during real world testing when unseen data is applied to the model. 70 samples were collected for each instrument, each of which would need to be normalised to the same length for feature extraction. A .csv file was created, containing two columns, one of which holds the filename of the sound clip and the other has the name of the instrument for the process of supervised learning.

## Feature Extraction

With the raw data collected, it is time to identify and extract the relevant features from the raw data. In this study, different methods of feature extraction were explored. While conducting research for the literature review, spectrograms appeared as a widely used option in this space. A spectrogram may be passed into the Machine Learning Model as an image and use that to characterise which spectrogram corresponds to each instrument. A spectrogram is a way to visualise sound as an image since it captures all the features of a sound clip by displaying the change in frequency content over time. The brighter areas indicate regions of higher energy. The pitch and intensity of the sound can be visualised as well as the smaller details like harmonics. Note that all the sound clips that were collected are of variable length.

The first method was to iterate through all the sound clips and convert the first second to spectrogram form. Once the spectrograms were generated and analysed, it became apparent that some of the sound clips did not start in the first second so all that was getting picked up was noise. Majority of the spectrograms outputted the correct images.

A pink and purple pixelated background

Description automatically generated with medium confidence

Figure 1: Spectrogram of a trumpet sample with just noise.

A purple and orange lines

Description automatically generated with medium confidence

Figure 2: Spectrogram of a trumpet over 1 second.

The same two trumpet samples will be used throughout the feature extraction process. This makes the comparisons more meaningful.

Just taking a spectrogram over a fixed time interval is not a viable solution. A more robust approach would be to find the maximum RMS value, i.e., the power of the signal, over one second. The code first loops through the datasets of sound clips, and then loops through the sound clip itself over 1 second intervals that increment every 0.5s to find the highest RMS region. The same two trumpet sounds are compared to one another again,

A purple and black background

Description automatically generated

Figure 3: Spectrogram for first trumpet sample.

A purple and orange sound waves

Description automatically generated

Figure 4: Spectrogram for second trumpet sample.

We can see that this is an improvement compared to the first method. However, we will only know for sure that it influenced the accuracy after applying it to the model of choice.

The next aspect to consider is the windowing method that is used. A window function may be applied to the to the sound file prior to computing the Fourier Transform. The reason for doing this is to suppress the discontinuities at the start and finish of the files, reducing the spectral leakage. The ensures the frequency contents of each file is accurately presented. The common choices out there are the Hamming, Hanning, and Blackman windows.

Based on information from the literature review, a Hamming window is preferrable. A Short-Time Fourier Transform was applied. The Scipy library was used here since it was much easier to apply window functions to create spectrograms.

A rectangle window was applied as a sort of benchmark upon which all the other window methods were compared to. The Hamming and Blackman windows were the two choices that were implemented.

Old was librosa and new was scipy. Scipy was much faster.

Issues with using scipy: Scaling.

dB scale of 5e-9

Methods used so far:

1. Spectrogram Generation: Transformation of the audio signal into an image that can be interpreted by the learning model. The spectrogram is the representation of frequencies in a signal over time. There is no real filtering, its just a transformation that helps with analysis.
2. Maximum RMS Extraction: Extracting the one second interval of the audio signal where the RMS value is the highest. Seen as filtering since the section of the signal with the highest frequency is selected.
3. Windowing: Applying various windows, Hamming and Blackman filters. Weights the signal, reducing the influence of the parts of the signal at the start and end of the window.