EEE4114F Course Project
Instrument Classification Machine Learning Task

Prepared by:
Robert Dugmore
Thiyashan Pillay

Prepared for:
EEE4114F
Department of Electrical Engineering
University of Cape Town

May 16, 2024

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

T.P.

May 16, 2024

Robert Dugmore

Thiyashan Pillay

Date

Contents

1 Introduction

The aim of this task was to create an image classification Machine Learning (ML) program that can distinguish between four different instruments - Piano, Snare, Trumpet, and Violin. The report follows a methodical approach proposed by Vimal et al. [1], which involved data collection, feature extraction, machine learning model construction, testing, and results.

The report outline includes conducting a literature review to engage with existing research and scholarly work relating to the task, with the objective of improving insight into the most effective methods of feature extraction and recommended ML models pertinent to this task. What followed was a discussion on the methods of data collection, after which different methods of feature extraction, such as spectrograms (with different windowing methods) and Mel-Frequency Cepstral Coefficients (MFCCs), were explored. The choice of ML model was then explained. With a clear understanding of methodologies, the core objective of testing was formulated - which included an examination of a shallower and deeper ML model and analysis of the different methods of feature extraction. The results were then displayed and analysed, leading to a final verdict which verified the findings.

The conclusion summarises the findings of the task and assesses the model's ability to perform in the real-world.

2 Literature Review

2.1 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 and MFCCs 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.

2.2 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. [1] 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.

2.3 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. [1] looked at using MFCCs as a method of feature extraction. The reason being that it is best for detection of human voices. Zhang et al. [2] 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. [3] 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. [4] conducted research on music instrument recognition. They used 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. [5] have 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.

2.4 Machine Learning Classification Methods

There has been plenty research on different classification methods. Vimal et al. [1] 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. [6] further emphasised that SVMs are 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. [7] quoted a 94% aggregate score for the measure of performance across different classification thresholds.

Zhang et al. [2] compared their method of Auto-Conditioned Recurrent Neural Networks (ACRNN) with existing methods and saw that their method obtained the highest classification accuracy of 93.7%. Other models like multi-stream CNN achieved a similar accuracy when using both raw data and spectrogram information, while ACRNN can with only the spectrograms. ACRNN uses a mix of Convolutional Neural Networks (CNNs) 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. [3] 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 allowed for the model to examine both local and global variations in features. 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. [4] 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 traditional neural networks since it is more controllable. The CNNs extract features from layers in the input data and connect 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.

2.5 Conclusion

The literature review provided valuable insight into the current methodologies used in audio classification, particularly for instrument classification. The review started off with the exploration of different methods of data collection, highlighting the importance of creating a diverse dataset to improve the robustness of the model when deployed in the real-world.

The common methods of feature extraction, most notably MFCC and ZCR, emerged as useful methods to provide the learning model with relevant information for classification tasks.

When it came the the ML classification methods, the options were endless, ranging from Random Forest to SVMs. However, CNNs stood out as being the best classification tool for this specific task based on research conducted on similar work.

Overall, the understanding of audio classification, specifically with regards to instruments, has been greatly improved after looking at the current solutions available.

3 Data Collection

As mentioned previously, four instruments were selected to train the model. Sound clips were collected for the Snare, Trumpet, and Violin, from freesound [8] and pixabay [9], which house massive libraries of royalty free music. It is very important to collect a variety of datasets that are unfiltered to resemble real-world applications. The piano sound files were collected by recording short songs played through computer speakers on Helm [10], an open-source digital synthesiser.

70-100 samples were collected for each instrument, where the data for each instrument is initially split into 80% training and 20% test data, with the training data further split into using into training and validation data using varying techniques at different stages.

4 Feature Extraction

Feature extraction was important in identifying relationships between different portions of the data, which improved the accuracy of our classification task. The data contained in audio files is often difficult to use in ML algorithms in its raw form. This was why feature extraction was needed. The data was converted into an understandable format that the ML model could interact with.

Spectrograms were the primary tool chosen for this task. Spectrograms visusally represent the frequency content of a signal over time, and are widely used in audio classification tasks. The dataset contained soundclips of varying length, which meant that spectrograms produced from this data would not have a uniform pixel count. This would not work since it is good practice to keep the input features to an ML model constant. To guarantee uniformity in pixel count, The spectrograms were generated over a one-second interval of the signal which contained the highest energy (calculated using RMS signal strength). This ensured that the spectrogram was taken for a portion of the signal where the instruments was actually playing. Spectrograms were scaled to the power spectral density of the signal and displayed, ensuring that signals of differing volumes still produced useable data. A decibel scale was used, amplifying small differences between the spectral content of the signals.

A common problem with spectrograms is spectral leakage when performing the Short-Time Fourier Transform (STFT) on the sounds. This would be seen as distortion in the spectrogram. While not always noticeable to the human eye, to an ML model this could be significant. Windowing was used to try and reduce spectral leakage. The options for windowing functions ranged from a simple rectangular window to more complex Blackman and Hann windows. Choosing the correct windowing method could prove to be tricky. It heavily depends on the inherent characteristics of the ML model being used. While a Blackman window offered better leakage, the frequency resolution was compromised. Conversely, rectangular windows offered great frequency resolution but bad leakage. The window which offered the best balance between trade-offs was the Hann window, with moderate frequency resolution and leakage. An analysis of the ML models using these three windows for feature extraction was performed.

In section 2.3, (MFCCs) were regarded as being very useful for classification of sounds; these were also duly investigated. MFCCs give a small set of features, in our case around 20, that describe the shape of the spectral envelope. MFCCs were also described as a way to view the power spectrum of a signal similar to how a human auditory system would perceive it. The MFCC was computed over a short window and resulted in a 2D matrix that consists of features. This matrix can then be plotted as a heatmap. The resulting images were then stored to be used later when the learning model will be implemented. No windowing methods were used here, and as with the spectrograms, each MFCC was calculated over the one-second interval of the signal which contained the highest RMS value. Given that the features were more distinct, it was interesting to view the model performance when using MFCCs as the input as oppose to spectrograms with the overall best windowing methods applied.

5 Machine Learning Classification Methods

In section 2.4, various ML models were looked at, however the one that stood out as being the best suited for this project was the CNN. Shreevathsa et al. [4] verified the CNN performance in a comparative study between the ANN and CNN. The ultimate conclusion was that the CNN worked far better in audio classification, outperforming the ANN.

Rather than comparing the performance between the ANN and CNN when changing the input features, it was deemed more insightful to analyse the model performance between a shallow and deeper neural network. This was tested by feeding both neural networks with spectrograms with different windowing methods applied and MFCCs with no comparison of windowing, as desribed in section 4.

Three different CNNs were utilised in this investigation. The first was a 'shallow' network, consisting of a single convolutional layer, followed by a max-pooling stage and two linear layers. The 'deeper' network consisted of two convolutional layers, each of which was followed by a max pooling stages. This was followed by three linear layers.

The main apples-to-apples comparison of network depth was performed on the 'shallow' and 'deeper' networks by training the networks on spectrograms as inputs. For the MFCCs, a alternative third network was established in lieu of the 'deeper' network. It was very similar to the simple network, but used a horizontal kernel. This was because the vertical features of MFCCs were thought to possibly show limited relational importance, instead, the horizontal (time) axis was likely to contain the most relational importance which ought to be assessed by the kernel. The MFCCs were also significantly lower-resolution images than the spectrograms and already contained sumarised data about large snippets of the original sounds; for this reason the max pooling stage was eliminated from the MFCC network. Additionally, for the MFCCs, both the original 'shallow' network and the horizontal-kernel network had to be modified to accept three input channels, since the MFCCs contained meaningful data in their colour spectrum (compared to the spectrograms which were grayscale).

All networks using Rectified Linear Units for their activation functions. Additionally, the final classification used the logarithmic softmax function to identify the class with the highest probability.

6 Core Objective

In sections 4 and 5, a general hypothesis was outlines regarding the specific questions that should be answered in section 7.

- Does the depth of the neural network affect the model performance? This was analysed by comparing the networks on the three types of spectrograms with different windowing methods.
- How significant is the effect of spectral leakage on the performance of the model? This was analysed by comparing the three types of spectrograms on the 'shallow' network.
- What is the performance of MFCCs on different networks?

Apart from the described changes to the MFCC network, hyperparameter tuning for the models was considered beyond the scope of the investigation. Most models discussed could likely be improved with hyperparameter tuning, and similarly could be improved with a larger dataset.

7 Testing and results

7.1 Comparing depth of neural networks

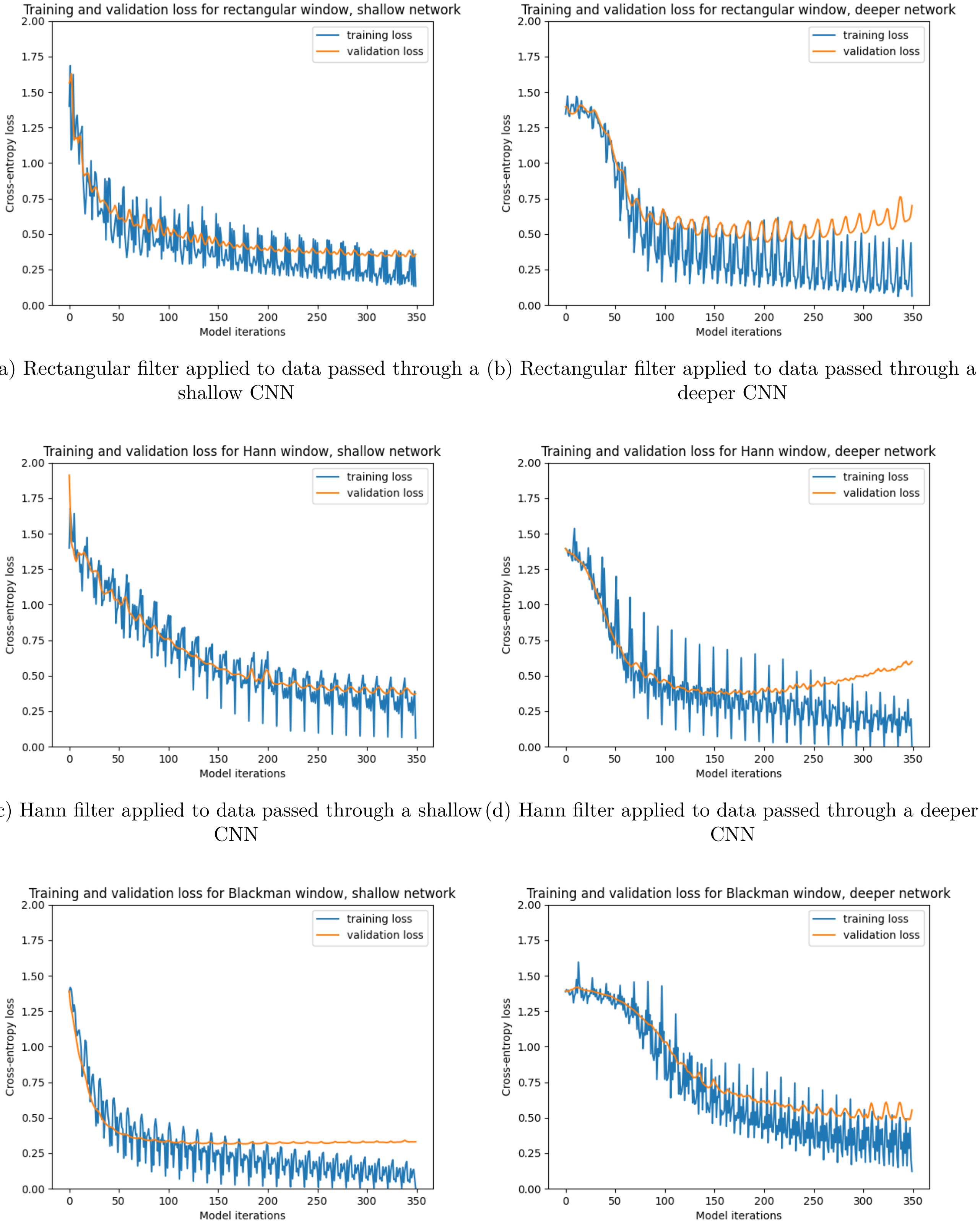
As discussed in section 4, A set of spectrograms were generated when each windowing method was applied to the datasets. In total there were three sets of spectrograms. Each set of spectrograms was passed through a shallow and deeper CNN, with the data shown in figure 1.

The graphs show the training and validation losses as the number of model iterations increase. Each model was trainged for 25 epochs (amounting to just over 350 iterations with batch sizes of 16). Models were trained at a learning rate of 0.0002, after experimentation revealed that gave relatively stable results. Models were trained on a single random split of the training data into 80% training data and 20% validation data.

The clear result from all three windowing methods is that the more complex model tends to overfit to the data more heavily within the number of iterations trained. Deeper networks have increased complexity and more features, making it possible for them to fit more finely to the training data; however, this can have the side effect of overfitting on the training data and poor validation performance. Deeper neural networks are also more likely to experience problems relating to vanishing and exploding gradient. This tends to happens to gradients when they are passed through multiple layers of learning during the training process, slowing it down.

The best-case validation performance for the 'shallow' network is better than that of the 'deeper' network for both the rectangular and Blackman windows, and is comparable for the Hann window. Training a deeper neural network in general should take longer to train due to more parameters having to be learnt, since the deeper networks have more layers where the learning process involves an interative process of adjusting the the learning parameters in an attempt to improve model accuracy, as well as increased complexity. Overall, the added computational requirements of the 'deeper' network cannot be justified.

A limitation of this comparison came from the splitting of data; the single training-validation split could yield results that are overly dependent on the specific split of data. Across the three window types, the conclusion that the simple model is more useful can still be deemed valid, but a meaningful evaluation of the difference between the window types cannot be made from this data.



(a) Rectangular filter applied to data passed through a shallow CNN

(b) Rectangular filter applied to data passed through a deeper CNN

(c) Hann filter applied to data passed through a shallow CNN

(d) Hann filter applied to data passed through a deeper CNN

(e) Blackman filter applied to data passed through a shallow CNN

(f) Blackman filter applied to data passed through a deeper CNN

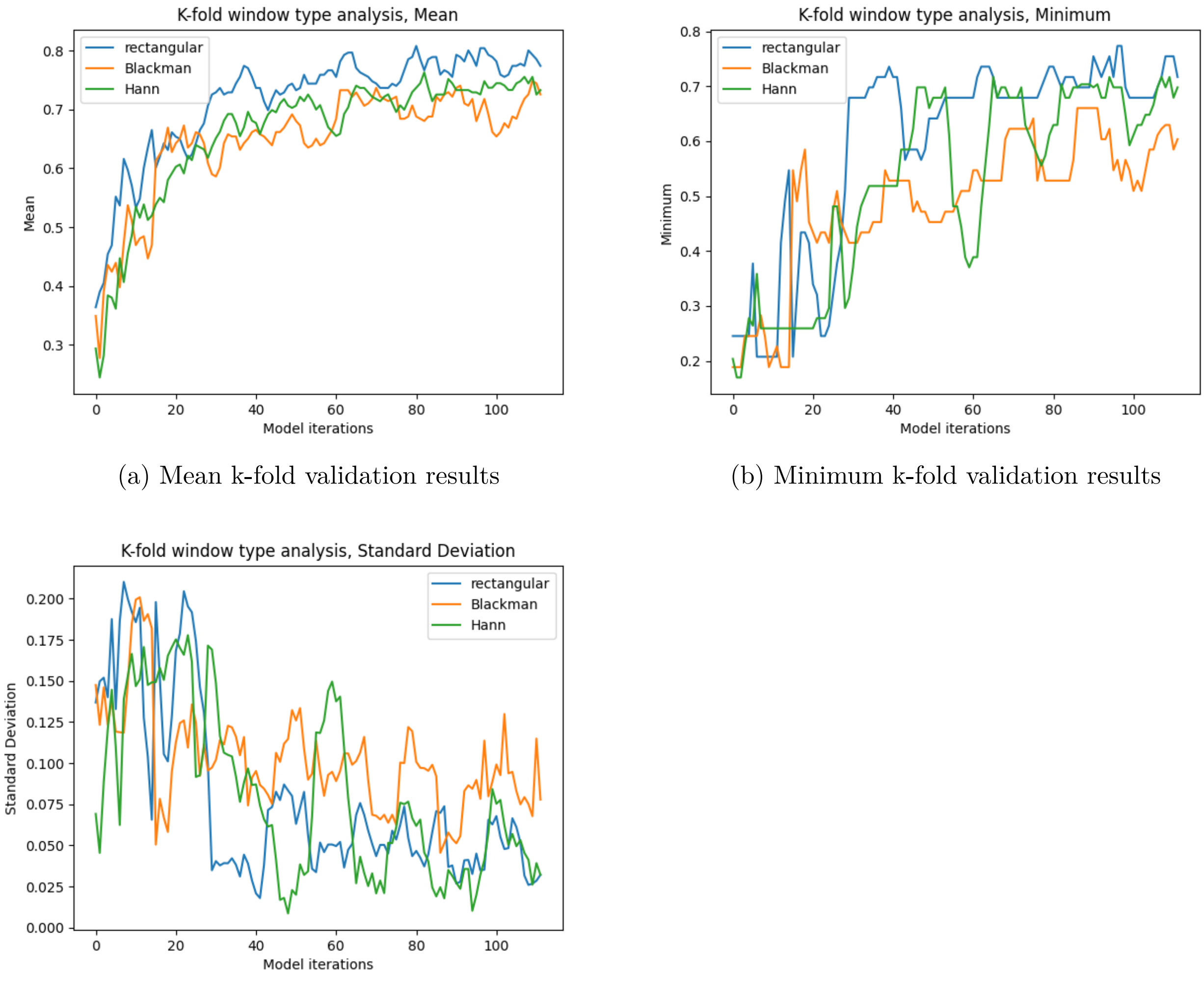
Figure 1: Result after Spectrograms were passed through neural networks of variable depth

7.2 Comparing windowing types

Shallow CNNs, which are far more computationally efficient, have room for improvement when it comes to accuracy. K-fold cross-validation is a is very powerful when it comes to assessing the model's performance on unseen data. The dataset is divided into 'k' subsets referred to as folds. The model is trained 'k' times and uses 'k-1' folds for training and '1' fold for validation. This would provide a more comprehensive result on the model performance.

By applying this to the shallow CNN trained with the data processed with various windowing methods, it provides a guide as to which model parameters to tune to improve performance.

Figure 2 shows the mean, min and standard deviation of the k-fold window type analysis. Table 1 then gives the accuracy of the last 5 k-fold iterations.



(a) Mean k-fold validation results

(b) Minimum k-fold validation results

(c) Standard deviation k-fold results

Figure 2: Graphical Representation of K-Fold Cross Validation Results

Bearing in mind that the rectangle and Blackman window have the worst and best attenuation of spectral leakage respectively, figure 2a shows that the model trained with data processed with the rectangular window had the highest average accuracy in comparison to the data processed with the

Window	k-fold accuracies (%)
Rectangle	[78.9 83.8 83.8 76.4 72.3]
Blackman	[63.1 77.2 64.0 82.6 67.4]
Hann	[69.3 78.7 70.9 73.9 77.9]

Table 1: k-fold validation data for the last 5 k-fold iterations

Blackman and Hann windows. Although, the data processed with the Hann window was not far off comparison. The model trained with data processed with the Blackman window did not perform as well as it was expected to. This shows that the higher frequency resolution associated with rectangular windows had an effect on the performance of shallow CNNs. The features extracted could likely be less affected by spectral leakage.

Figure 2b shows the worst case accuracy across each model iteration, and whats interesting is that the model trained on the data processed with the rectangle window still prevails as the best performing of the three. The model trained on the data processed with the Hann window could be regarded as the worst of the three.

Figure 2c shows that the model trained on the data processed with the rectangular window has the lowest variation of the three models. This means that for the current configuration of the model, frequency resolution is valued above the spectral leakage, proving that using a rectangular window is the best.

Table 1 illustrates the above mentioned conclusion numerically. The average accuracies obtained from the model trained on the data processed with the rectangular window significantly outperformed the other two models in the table.

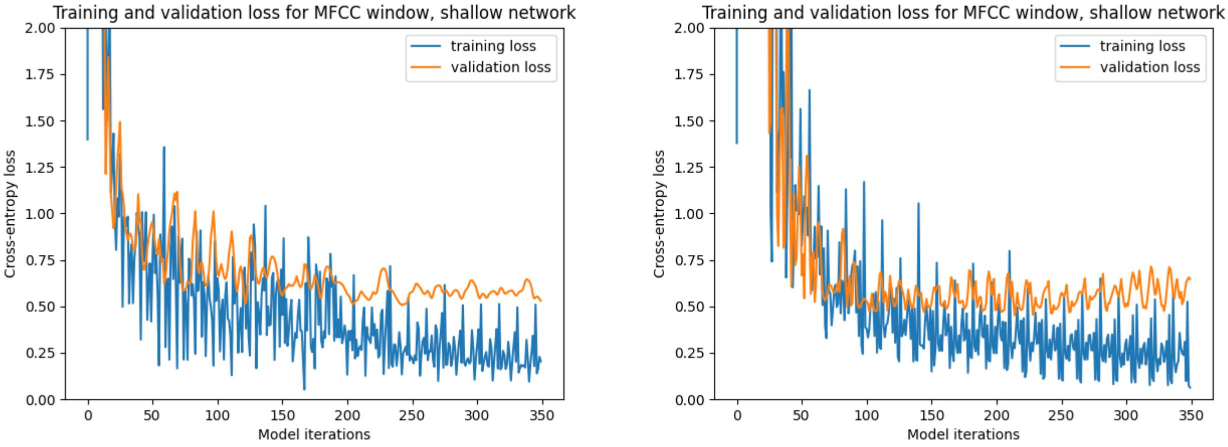
7.3 Analysis of MFCCs

As explained in section 6, additional analysis was performed on MFCCs with the standard ‘shallow’ network, and a modified ‘shallow’ network which featured horizontal (time-only) convolutional kernels and no max-pooling stage. The results using a single training-validation split are shown in Figure 3. The results do seem to indicate better validation performance from the customised model, shown from the lower final validation loss. While a limited amount of overfitting is evident for both networks, it does not seem to be a prevalent issue. However as in subsection 7.1, the single-split used means that this could be quite dependent on the specific split of the data.

A K-fold analysis of both networks as described in subsection 7.2 was performed on both to find validation accuracy. The mean was found to be 74.2%, and the minimum across the 5 folds 68.3%. This is comparable to the results using spectrograms, and it can be concluded that this model did not benefit substantially from the use of MFCCs in place of spectrograms.

8 Conclusions

The purpose of this task was to construct an ML model for instrument classification. The report began with a short introduction, followed by a literature review. Here, information was gathered on how to properly go about this task. The data was then collected from various sites, and different methods of



(a) Standard 'shallow' network

(b) Network with horizontal kernels

Figure 3: Comparison of CNNs for MFCCs

feature extraction were performed to extract the relevant input features. The ML model creation was then explained. With enough information, the core objectives were then outlined, and investigated in testing.

In testing, the deeper neural network tended to overfit the data more frequently, which is why a shallower neural network was preferred, despite yielding a lower model accuracy. With regards to the various windowing methods, the model trained with data processed with a rectangular window proved to be the best, which implied that the neural network was more sensitive to frequency resolution over spectral leakage. MFCCs were briefly examined, including being trained on a custom CNN using a horizontal kernel. Though the horizontal kernel did appear to yield better results, the overall performance of the MFCCs was comparable to that of the spectrograms.

Given these conclusions, the model was retrained on all the training data, and evaluated on the test data to gain a concrete idea of the model's real-world performance. The final model accuracy was quoted at 74.2%. Given that the model was trained on unfiltered data, this is rather impressive.

9 Future development

This model provided an analysis of the impact of various parameters including model depth and spectrogram windowing methods for classification of instruments. Limited hyperparameter tuning was performed on a version of the model using MFCCs, but all models examined could likely be improved through more extensive hyperparameter tuning.

The more complex model examined was seen to be more prone to overfitting to the training data. The conclusion in this investigation was that the simpler model was more appropriate for the given dataset, but overfitting could alternatively be improved through analysis of regularisation techniques.

The dataset used in this investigation included only a limited number of instruments and a small number of samples for each. A more comprehensive model would use substantially more data, and would likely need to sample many more instruments to be useful in the real world.

References

[1] B. Vimal, M. Surya, Darshan, V. Sridhar, and A. Ashok, "Mfcc based audio classification using machine learning," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2021, pp. 1–4.

[2] Z. Zhang, S. Xu, S. Zhang, T. Qiao, and S. Cao, "Attention based convolutional recurrent neural network for environmental sound classification," *Neurocomputing*, vol. 453, p. 896–903, Sep. 2021. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2020.08.069

[3] Y. Wu, H. Mao, and Z. Yi, "Audio classification using attention-augmented convolutional neural network," *Knowledge-Based Systems*, vol. 161, pp. 90–100, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705118303848

[4] P. Shreevathsa, M. Harshith, A. R. M., and Ashwini, "Music instrument recognition using machine learning algorithms," in *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, 2020, pp. 161–166.

[5] S. Essid, G. Richard, and B. David, "Musical instrument recognition by pairwise classification strategies," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1401–1412, 2006.

[6] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using svm and rbfnn," *Expert Systems with Applications*, vol. 36, no. 3, Part 2, pp. 6069–6075, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417408004004

[7] O. A. Olabanjo, A. S. Wusu, and M. Manuel, "A machine learning prediction of academic performance of secondary school students using radial basis function neural network," *Trends in Neuroscience and Education*, vol. 29, p. 100190, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211949322000187

[8] "freesound," https://freesound.org/, 2024, [Accessed 16-05-2024].

[9] "pixabay," https://pixabay.com/, 2024, [Accessed 16-05-2024].

[10] M. Tytel, "Helm," https://tytel.org/helm/, 2018, [Accessed 16-05-2024].