Classification of Guitar Chords and Effects Using Machine Learning

Diwas Lamsal, Sunny Kumar Tuladhar, Bishal Khanal

*School of Engineering and Technology,, Master's in Computer Science,
Asian Institute of Technology, Klong Luang, Pathum Thani, Thailand
[1]Diwas.Lamsal@ait.ac.th

#School of Engineering and Technology, Master's in Data Science and Artificial Intelligence,
Asian Institute of Technology, Klong Luang, Pathum Thani, Thailand
[2]Sunny.Tuladhar@ait.ac.th

*School of Engineering and Technology,, Master's in Computer Science,
Asian Institute of Technology, Klong Luang, Pathum Thani, Thailand
[3]Bishal.Khanal@ait.ac.th

***Abstract*— The music industry has its own share of problems when having to replicate and use the sounds created by other professionals into their own music. This is usually done by experts by merely listening to the sound, comparing its spectrums and trying to recreate the effect using the available tools. It is still difficult for beginners and amateurs to decode these parameters. Using deep learning and different representations of audio , we see that we can classify the sounds based on the effects used and also based on the notes being played. Here we use mel's spectrogram, chromogram, and downsampled signals for classification of guitar effects and chords. We compare these methods of representations and different machine learning architectures to see which works better. We found that MFCCs combined with a CNN gives very good results on the guitar effect classification task. For the chord classification, despite having very few samples for training, the Chromagram, combined with a CNN shows promising results.**

***Keywords*— Music, Guitar effects, Chords classification, Spectrogram, Deep Learning**

1. Introduction

The problem of identifying audio effects and chords in music is always a mystifying task especially for beginners in the field and a lot of it is based on guesswork to get the closest possible sound from the one that is being replicated.

Our work aims to identify different representations tried on different deep learning models for two tasks: Audio effect recognition and chords recognition for guitars based on the IDMT dataset.

2. Representations

We will be using three types of representations in our work. One is the raw input information from the audio file. The other two are visual representations in the form of *Spectrogram* and a *Chromagram*. Our main source of audio preprocessing was the librosa [2] library from python.

2.1 Mel-frequency cepstral coefficients(MFCC)

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum")[4] This approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal spectrum. This frequency warping can allow for better representation of sound

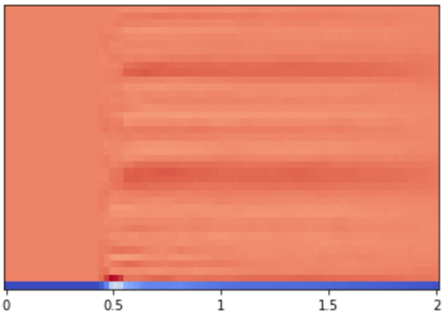

Figure 1 - MFCC representation of a sample from the effects dataset

2.2 Chromagram

Chromagram is defined as the whole spectral audio information mapped into one octave. Each octave is divided into 12 bins representing each one semitone[5]. This means that even if a note is in a higher or lower octave than the other notes, it will still be represented in a single octave. This captures the pitch/frequency class of an audio very well.

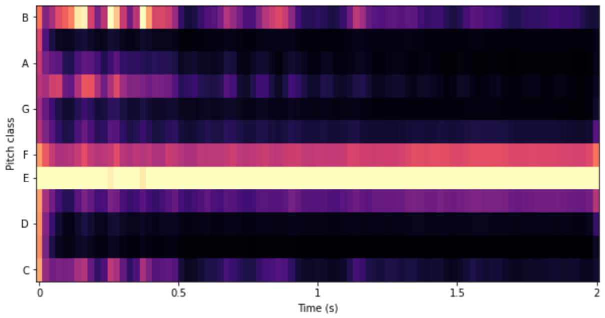

Figure 2 - Chromagram representation of a sample from the chords dataset

2.3 Raw Audio

The natural representation for audio would be the direct use of raw audio waves. Its length and resolution is defined by the *sampling rate* which we will be changing to see its effect on our model performances. *Sampling rate* is the number of samples per second (or per other unit) taken from a continuous signal to make a discrete or digital signal. Higher the value, higher the quality of the sound. 44.1 KHz is the common sampling rate for most audio we hear.

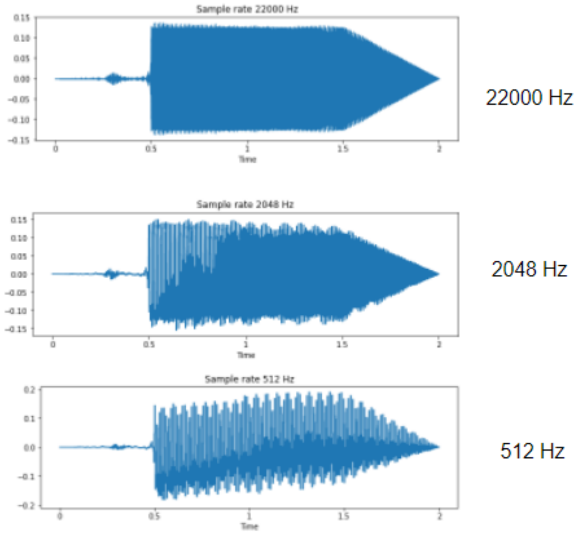

Figure 3 - Different Sample rates and their visualisations

3. Related work

In [1] they have used the IDMT dataset and predicted the guitar effects and parameters of those effects using FxNet, SetNet and MultiNet. Using electric guitar recordings of single notes, 2-note intervals and 3- or 4- note chords; they classified with high accuracy samples processed with 13 overdrive, distortion and fuzz plugins. The inputs to their Neural Network models were mel-power spectrograms. Although they are solving a multi-task problem, we show that our combination of MFCCs with a CNN outperforms their FxNet for classification in the Mono Discrete dataset.

Ever since MFCCs were introduced in the 1980s [12], they have been quite popular. They have empirically shown good results on both speech as well as other audio processing tasks [13]. These representations improved audio classification performance on traditional Machine Learning algorithms such as RBF SVMs and Random Forest classifiers back when CNNs were not common [14]. Even now, these classical machine learning algorithms on many occasions show good performance with MFCCs [15], sometimes beating deep learning architectures. Recent works involve the use of MFCCs together with deep learning models such as RNNs [16], CNNs [17], or more recently, newer models such as Vision Transformers [18] to achieve state-of-the-art performance. CNNs are some of the most popular architectures used together with MFCCs, as the earlier works showed that this combination of CNNs and MFCCs significantly outperformed other methods [19]. These are commonly used for tasks such as cough classification [20], heart sound classification [21], and lung sound classification [22].

In the music domain, [24] shows that other hand-crafted features like chroma spectral features, which represent the energy distribution of a signal's frequency content across the 12 pitch classes, generally perform better than other techniques like MFCC for music classification tasks. In this paper, we see how chromagrams can outperform MFCCs in some specific audio classification tasks.

Tasks such as musical genre classification have seen the use of MFCCs together with other good performing classification algorithms [25]. In [39], they use CNN architecture for chord detection. Similarly, we have also seen CNNs perform human-level accuracy on music genre classification [43]. CNNs are very widely used for such music-classification tasks [44, 45, 46, 47]. On the other hand, [40] shows an SVM model outperforming neural networks where they purely used MFCCs for data representation. We have seen similar cases of simpler machine learning models outperforming or coming close to the level of deep learning architectures [42] and thus find it plausible to compare these models with the deep learning models.

Papers like [23] have also shown that 1D CNNs might work well with raw audio signals. As the raw audio signals can be represented as a vector, these 1D CNNs can extract features from these signals. As described in [26], the convolutional layers in such 1D CNNs capture the time-frequency characteristics of the audio signal and learn filters relevant to the audio classification task at hand (in their case, for music genre classification). In [27], they have shown a comparison between 1D CNN and 2D CNN architectures for audio classification tasks. While the 2D CNNs outperform 1D CNNs, the 1D CNNs might still give acceptable results. Thus we found 1D CNN also as a viable comparison method to which we can directly feed raw audio vectors. [29] also similarly compares the performance of feeding raw audio signals to 1D CNNs and features such as MFCCs to 2D CNNs. Their results show that 1D CNNs tend to perform better (not better than 2D CNNs) if the audio signals are longer.

Audio sound is a one-dimensional vector that stores numerical value corresponding to the sample taken at different time steps, hence it is a time series signal and the use of Recurrent Neural Network would be appropriate. Since the impact of the LSTM network has been notable in language modeling, speech-to-text transcription, machine translation, and other applications [7], we decided to go with LSTM based RNN. According to [8], the Bi-directional RNN can provide faster development of real applications with better results than the unidirectional RNN. Hence, we went for Bi-directional LSTM model. In [28], they have used a Bi-directional RNN structure for emotion recognition from raw audio data.

Attention mechanisms have become an integral part of compelling sequence modeling and transduction models in various tasks, allowing modeling of dependencies without regard to their distance in the input or output sequences [9, 10]. In [11], they proposed the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output. We have also seen some attention-based architectures as in [41] to perform music genre classification. Because of this reason, it seems wise to use Transformer as an encoder for our problem.

4. Datasets

The Institute for Digital Media Technology (IDMT) Dataset provides audio datasets for several tasks [30]. We used the IDMT-SMT-Audio-Effects, and the IDMT-SMT-Chords datasets for our experiments.

The Audio Effects dataset contains guitar-generated audio with different effects. The dataset contains both monophonic (only one note played at a time) and polyphonic (more than one notes played at a time) labelled data. However, we only used the monophonic data for guitar (no bass guitar). The dataset contains information such as the instrument settings, spread, depth, and level of the effects, the notes, string, and the fret of the guitar, and so on. However, we only used the primary labels, i.e., the effect being used. There are six classes of data – Chorus, Distortion, EQ, Feedback Delay, Flanger, and NoFX. The total number of samples is 9360 with 2 seconds of audio sampled at 44.1KHz each. The mean number of samples per class is 1560.0 ± 476.58.

Likewise, the chords dataset consists of 6 audio files generated using MIDI data. All of these files are of equal length and contain the same chords played at a particular point in time, given by the labels. Each file is 9 minutes and 6 seconds long, with 2 seconds per one chord class. Thus, we have 273 samples per file. The six different audio files differ by the different guitar settings used to generate these files. There are 84 classes of data. The total number of samples is 1638 with 2 seconds of audio sampled at 44.1KHz each. The mean number of samples per class is 19.5, ± 2.7581.

5. Methods

5.1 Traditional Machine Learning Algorithms

We used two classical machine learning algorithms before diving into deep learning: Logistic Regression and Linear Support Vector Machines (SVM) (we did not try to use an RBF kernel with grid search of hyperparameters, which might have improved performance). Logistic Regression attempts to maximize the likelihood that a particular sample comes from the given class. The reader is referred to Chapter 4.4 of [31] or other resources to learn more about logistic regression. The idea behind SVMs on the other hand, is to draw a hyperplane, separating the data into classes and attempting to maximize the margin between the nearest data points and this hyperplane. The reader is similarly referred to Chapter 12 [31] or other resources for more details.

For both of these techniques, we trained these models on the raw soundwave data with sampling rates of 2KHz and 4KHz on both of our datasets. We also fed the flattened MFCC spectrograms and Chromagrams to these models. The performance is discussed in the results section.

5.2 1D CNN with Raw Data

The raw audio signals are represented as a vector of length given by the sampling rate and the duration of the audio. For example, the raw representation of a sound wave sampled at 2048 Hz, and 2 seconds in length would have a 4096-element vector. We pass this through a 1-D Convolutional Neural Network to extract the features and perform a softmax to classify the input audio signal. We started with a simpler model and increased the complexity so as to not overfit. We have five convolution layers, followed by two dense layers finally connected to the output layer. We also used dropout of 0.2 [32] after the convolution layers and the first fully connected layer. We used Adam optimizer [33] and cross-entropy loss function to train the model. The final CNN we used to classify the raw audio signals has the following structure:

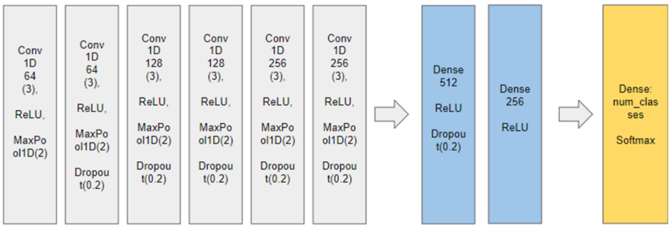
Conv 1D 64 (3), ReLU, MaxPool1D(2)
Conv 1D 64 (3), ReLU, MaxPool1D(2) Dropout(0.2)
Conv 1D 128 (3), ReLU, MaxPool1D(2) Dropout(0.2)
Conv 1D 128 (3), ReLU, MaxPool1D(2) Dropout(0.2)
Conv 1D 256 (3), ReLU, MaxPool1D(2) Dropout(0.2)
Conv 1D 256 (3), ReLU, MaxPool1D(2) Dropout(0.2)
Dense 512 ReLU Dropout(0.2)
Dense 256 ReLU
Dense: num_classes Softmax

Figure 4 - 1D CNN Structure

5.3 2D CNN with MFCC and Chromagrams

MFCCs and Chromagrams are features extracted from the raw audio data and take the shape of a 2-D matrix. For MFCCs, the recommended number of filter banks, which are used to sample the power spectrums of the audio signal, typically ranges from 20-40 [34]. In our case, we used 40. This gives a matrix of size 40 x 87 (the 87 comes from the hop length applied to the MFCC and our audio being 2 seconds). We prepare the data for training by taking each raw audio signal and converting them to respective MFCCs. We then pass this through our 2D CNN model.

For our CNN architecture, we used three convolution layers, followed by global average pooling [35] to reduce the number of parameters passed on to the dense layer instead of flattening. Each convolution layer is followed by a Batch Normalization layer [36] before the ReLU layer. This was then followed by a max pooling layer. Dropout of 0.5 is used on the fully connected layer before passing it to the output layer.

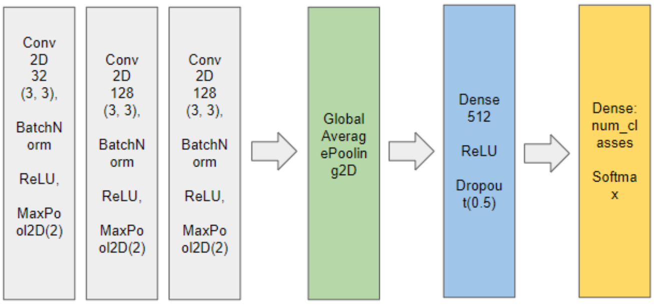
Conv 2D 32 (3, 3), BatchNorm ReLU, MaxPool2D(2)
Conv 2D 128 (3, 3), BatchNorm ReLU, MaxPool2D(2)
Conv 2D 128 (3, 3), BatchNorm ReLU, MaxPool2D(2)
Global AveragePooling2D
Dense 512 ReLU Dropout(0.5)
Dense: num_classes Softmax

Figure 5 - 2D CNN Structure

Chromagram, on the other hand, typically represents the twelve pitch classes in the audio signal. This produces a matrix of 12 x 87 with the default hyperparameters [37]. The CNN used for the Chromagram had a very similar structure, except for one less convolution layer, and the dense layer comprising 256 neurons (previously 512).

5.4 LSTM-based Model

LSTM was introduced to store relevant information over an extended period of time [6]. So, the networks based on LSTMs are well-suited to classifying, processing and making predictions based on time series data. Since the raw audio contains the information about the amplitude over time, it makes sense to use LSTM in our model.

The architecture consists of Bi-directional LSTM to output representations of the given input which is then used by the fully-connected classifier for the classification purpose. For the input, the raw audio is sampled with a sampling frequency of 2048 Hz which is then normalized to make data in the range of -1 to 1. This long input sequence of length n is then divided into multiple patches (np) which are then stacked together along the first dimension to get the new axis as shown in the Figure 6. So the data now is an array of size np x pl, where pl is the patch length (equal to int(n / np)). The first dimension represents the time axis. This is then fed to the Bi-directional LSTM. The output from the Bi-LSTM is the concatenation of output from two directions. Here we sum up the output from the two directions together and use the first and last sequence as the representation of the input. This representation is then unflattened and passed to the fully-connected classifier to predict the corresponding class of the input. The classifier consists of two linear layers, first with dropout of 0.5 and ReLU as non-linear layer, and second with no dropout and softmax as non-linear layer.
We use cross-entropy as a loss function and Adam optimizer as an optimization algorithm.
The detailed architecture of this model is shown in Figure 6.

5.5 Transformer-based Model

The overview of this model is depicted in Figure 7. The model is based on the Transformer Encoder. Here, instead of using input embedding like in encoder of original Transformer paper, we pass 2D representation of the input, like in LSTM based model, to the linear layer followed by ReLU non-linear function and then to the encoder with positional encodings added to the input as shown in the model diagram. The encodings from the transformer encoder are then passed to the classifier to correctly predict the class of the input. The classifier used in this model is the same as the LSTM based model which is described earlier in section 5.6. The input to this model is the same as the input to the LSTM based model.

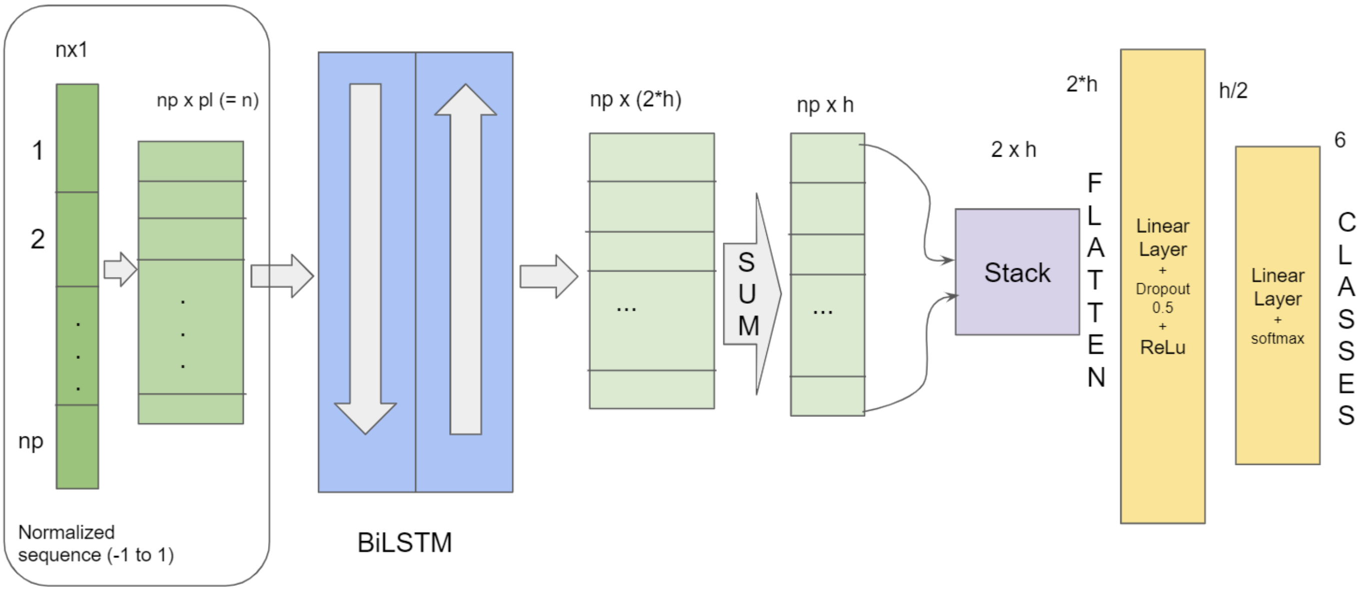

Figure 6 - Block diagram of LSTM based model. Here n (=2 x sampling frequency) is the sequence length, np is the number of patches, pl is the patch length (= n/np). The sampling frequency is 2048 Hz and the number of patches (np) is 16. The input size of BiLSTM is 256 (= patch length), hidden size (h) is 512 and number of layers is 2.

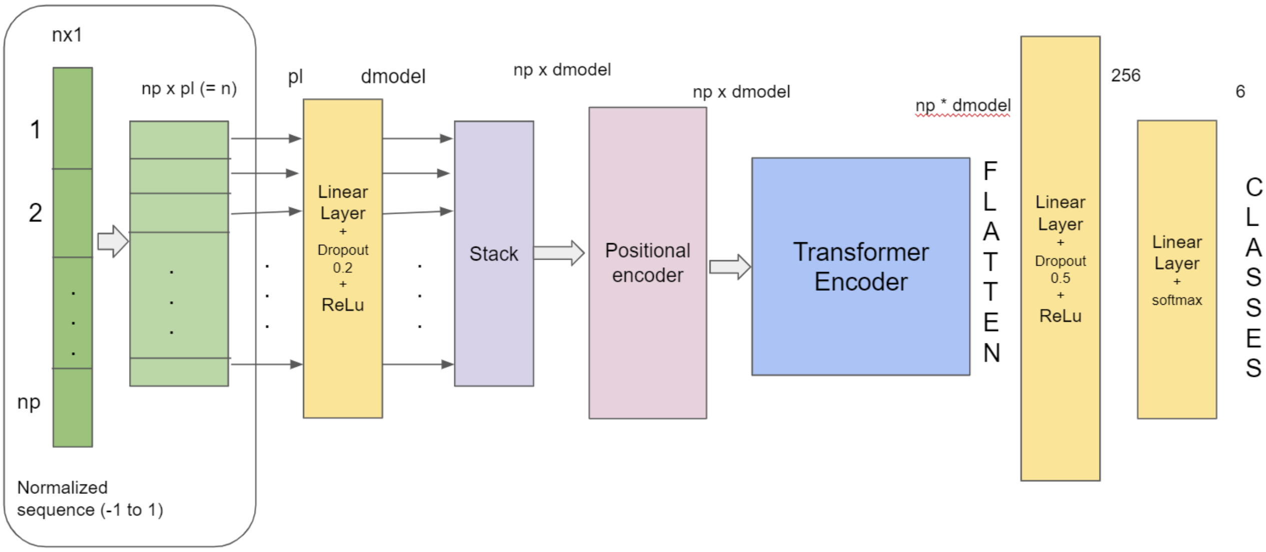

Figure 7 - Block diagram of Transformer-based model. The architecture of the Transformer encoder block is based on following the value of the parameter that is used to define it. num_layers = 3, dmodel = 128, nhead = 2, dim_feedforward = 128, dropout = 0.1. The input for this model is the same as for the LSTM based model.

6. Results

6.1 MFCCs and Chromagrams

Table 1: Results from 2D CNN on MFCC and Chromagrams

Dataset	MFCC+2DCNN	Chromagram +2DCNN
Effects	**99.2%**	92.1%
Chords	68.9%	**80.2%**

The best overall results come from processed audio signals. On the effects dataset, MFCC passed through the CNN classifier gives the best results at 99.2%. Whereas for the chords dataset, the MFCC did not show a good performance as it did using Chromagrams.

6.2 Raw Audio Signal

Table 2: Results from 1D CNN, LSTM, and Transformer on raw audio signals from the Audio Effects dataset

Sampling Rate	1DCNN	LSTM	Transformer
22KHz	-	-	-
2KHz	94%	55%	58%
512Hz	78%	-	-

For the raw audio signal on the effects dataset, passing the raw 2KHz signal through the 1D CNN gave a good accuracy score of 94%. Testing at different levels showed that using a 512 Hz signal does not produce good results.

Table 3: Results from 1D CNN on raw audio signals from the Guitar Chords dataset

Sampling Rate	1DCNN
22KHz	45%
2KHz	42%
512Hz	20%

On the guitar chords dataset, the 1D CNN did not show good results on any of the sampling rates. There is a significant jump in accuracy from 512 Hz to 2 KHz, whereas the increase in accuracy seems to plateau going beyond 2 KHz.

6.3 Classical Machine Learning Methods

Table 4: Results from Classical Machine Learning methods on MFCCs and Chromagrams

Model	Dataset	MFCC	Chromagram
Logistic Regression	**Effects**	92.3%	60.8%
	Chords	62.1%	63.8%
Support Vector Machine	**Effects**	95.8%	60.4%
	Chords	63.6%	64.7%

The classical machine learning algorithms were fed with flattened MFCC and Chromagrams. Both logistic regression and linear SVM show good performance on the effects dataset with MFCC. On the chords dataset, these models are unable to provide a good accuracy.

Table 5: Results from Classical Machine Learning methods on raw data

Model	Dataset	2KHz	4KHz
Logistic Regression	**Effects**	30.3%	30.6%
	Chords	24.7%	30.2%
Support Vector Machine	**Effects**	29.3%	30.3%
	Chords	29.1%	41.0%

These models generally fail to show good performance on raw audio signals. However, we can see an increase in performance on the chords dataset for linear SVM going from 2KHz to 4KHz. It is yet to be seen if this performance keeps increasing at higher sampling rates.

7. Discussion

The MFCC with the 2D CNN performs the best when predicting the Audio Effects. This could be because the shape of the audio changes when effects are applied and this is well reflected in the MFCC.

Chromagram with 2D CNN performs best for Chords dataset. The Chords dataset is very tricky as it has many classes and very few samples per class. Since the chromagram reflects the pitch information accurately the chords could be identified with high accuracy even with less data.

Increasing the sample rate improves the accuracy of predictions but above 2000 Hz the increase in accuracy diminishes. This could be because the resolution at 2000 Hz is enough for the model to identify the key features to classify both the effects and the chords.

The Transformers' low accuracy could be attributed to insufficient amount of data and the length of audio signals. Transformers usually need huge amounts of data to train which neither of these datasets have.

8. Future Work

This particular line of audio recognition for guitar effects is fairly new. Since it is a very widely played instrument, this effects and chord prediction model could benefit a lot of people. Here are some future works that could be useful

- In the future the chord and audio effects predictions could be combined into one network so both could be predicted at once.
- Also a model to predict a combination of effects from the same audio would be useful as most forms of audio are usually a combination of multiple effects.
- A model to predict the level of parameters in each effect in the audio effect would be extremely useful as the exact parameters are the difficult part.
- The transformer model could be trained using bigger datasets
- Other representations such as the Principle Cepstral Coefficients and Principle Spectral Coefficients have been proposed to show better performance than MFCCs [38]. Perhaps these representations could be tested to see if they show a better performance than Chromagrams on the chords dataset.
- Audio augmentation methods such as masking (for MFCCs) and time shifting among others have been proposed [48, 49]. As our chord dataset suffers from a lack of samples, audio augmentation might be able to improve the performance of our models.

9. Conclusion

It seems that we can use Deep learning networks to accurately predict Audio effects for guitars. This could be integrated into Digital Audio Workstation (DAW) so people can reverse engineer the sound.

The chord identification could be used for automatic transcription of music for beginners to learn from.

If parameters of the effects are also predicted properly this could be huge leap for beginner musicians and producers in recreating professional quality sounds (usually behind paywalls)

MFCC seems to be one of the best representations of sounds for effects classification and Chromagram seems to be the best representation for Chord classifications.

Acknowledgments

This research was possible due to the project work we had to do as a part of the Master's in Data Science and Artificial Intelligence course in the subject Recent Trends in Machine learning as taught by Prof. Matthew Dailey. We would also like to thank our Teaching Assistant, Alisa Kunapinun for continuous support throughout the course.

References

[1] Comunità, Marco, Dan Stowell, and Joshua D. Reiss. "Guitar Effects Recognition and Parameter Estimation With Convolutional Neural Networks." arXiv preprint arXiv:2012.03216 (2020).

[2] B. McFee et al., "librosa: Audio and Music Signal Analysis in Python," Proceedings of the 14th Python in Science Conference. SciPy, 2015.

[3] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[4] Mel-frequency cepstrum - Wikipedia", En.wikipedia.org, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum. [Accessed: 08-May- 2022].

[5] H. Ezzaidi, M. Bahoura, and G. E. Hall, "Towards a Characterization of Musical Timbre Based on Chroma Contours," Communications in Computer and Information Science. Springer Berlin Heidelberg, pp. 162–171, 2012.

[6] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[7] Henry W. Lin and Max Tegmark. Criticality in formal languages and statistical physics. Entropy, 19(7):299, Aug 2017.

[8] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014.

[10] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. In International Conference on Learning Representations, 2017

[11] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[12] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 4, pp. 357-366, August 1980, doi: 10.1109/TASSP.1980.1163420

[13] Ruijie Zhang, Bicheng Li and Tianqiang Peng, "Audio classification based on SVM-UBM," 2008 9th International Conference on Signal Processing, 2008, pp. 1586-1589, doi: 10.1109/ICOSP.2008.4697438.

[14] P. Dhanalakshmi; S. Palanivel; V. Ramalingam (2009). Classification of audio signals using SVM and RBFNN. , 36(3-part-P2), 6069–6075. doi:10.1016/j.eswa.2008.06.126

[15] B. Vimal, M. Surya, Darshan, V. S. Sridhar and A. Ashok, "MFCC Based Audio Classification Using Machine Learning," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 1-4, doi: 10.1109/ICCCNT51525.2021.9579881.

[16] Rejaibi, E., Komaty, A., Meriaudeau, F., Agrebi, S., & Othmani, A. (2022). MFCC-based Recurrent Neural Network for automatic clinical depression recognition and assessment from speech. Biomedical Signal Processing and Control, 71, 103107. doi:10.1016/j.bspc.2021.103107

[17] Bardou, Dalal; Zhang, Kun; Ahmad, Sayed Mohammad (2018). Lung sounds classification using convolutional neural networks. Artificial Intelligence in Medicine, (), S0933365717302051–. doi:10.1016/j.artmed.2018.04.008

[18] Y. Khasgiwala and J. Tailor, "Vision Transformer for Music Genre Classification using Mel-frequency Cepstrum Coefficient," 2021 IEEE 4th International Conference on Computing, Power and

Communication Technologies (GUCON), 2021, pp. 1-5, doi: 10.1109/GUCON50781.2021.9573568.

[19] Takahashi, Naoya, et al. "Deep convolutional neural networks and data augmentation for acoustic event detection." arXiv preprint arXiv:1604.07160 (2016).

[20] V. Bansal, G. Pahwa and N. Kannan, "Cough Classification for COVID-19 based on audio mfcc features using Convolutional Neural Networks," 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), 2020, pp. 604-608, doi: 10.1109/GUCON48875.2020.9231094.

[21] Deng, Muqing; Meng, Tingting; Cao, Jiuwen; Wang, Shimin; Zhang, Jing; Fan, Huijie (2020). Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. Neural Networks, 130(), 22–32. doi:10.1016/j.neunet.2020.06.015

[22] Bardou, Dalal; Zhang, Kun; Ahmad, Sayed Mohammad (2018). Lung sounds classification using convolutional neural networks. Artificial Intelligence in Medicine, (), S0933365717302051–. doi:10.1016/j.artmed.2018.04.008

[23] L. Vrysis, I. Thoidis, C. Dimoulas, and G. Papanikolaou, "Experimenting with 1D CNN Architectures for Generic Audio Classification," Paper 10329, (2020 May.). doi:

[24] Birajdar, G.K., Patil, M.D. Speech/music classification using visual and spectral chromagram features. J Ambient Intell Human Comput 11, 329–347 (2020).

[25] Tzanetakis, G.; Cook, P. (2002). Musical genre classification of audio signals. , 10(5), 0–302. doi:10.1109/tsa.2002.800560

[26] S. Allamy and A. L. Koerich, "1D CNN Architectures for Music Genre Classification," 2021 IEEE Symposium Series on Computational Intelligence (SSCI), 2021, pp. 01-07, doi: 10.1109/SSCI50451.2021.9659979.

[27] H. Wang, D. Chong, D. Huang and Y. Zou, "What Affects the Performance of Convolutional Neural Networks for Audio Event Classification," 2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW), 2019, pp. 140-146, doi: 10.1109/ACIIW.2019.8925277.

[28] R. Orjesek, R. Jarina, M. Chmulik and M. Kuba, "DNN Based Music Emotion Recognition from Raw Audio Signal," 2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA), 2019, pp. 1-4, doi: 10.1109/RADIOELEK.2019.8733572.

[29] Wei, Qingkai & Liu, Yanfang & Ruan, Xiaohui. (2018). A report on audio tagging with deeper CNN, 1D-CONVNET and 2D-CONVNET.

[30] Fraunhofer Institute for Digital Media Technology IDMT. 2022. Datasets - Fraunhofer IDMT. [online] Available at: https://www.idmt.fraunhofer.de/en/publications/datasets.html [Accessed 9 April 2022].

[31] Hastie, Trevor,, Robert Tibshirani, and J. H Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York: Springer, 2009. Print.

[32] Park, S., Kwak, N. (2017). Analysis on the Dropout Effect in Convolutional Neural Networks. In: Lai, SH., Lepetit, V., Nishino, K., Sato, Y. (eds) Computer Vision – ACCV 2016. ACCV 2016. Lecture Notes in Computer Science(), vol 10112. Springer, Cham. https://doi.org/10.1007/978-3-319-54184-6_12

[33] Kingma, D. and Ba, J. (2015) Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)

[34] Fayek, H., 2022. Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between. [online] Haytham Fayek. Available at: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html [Accessed 1 May 2022].

[35] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).

[36] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." International conference on machine learning. PMLR, 2015.

[37] Librosa.org. 2022. librosa.feature.chroma_stft — librosa 0.9.1 documentation. [online] Available at: https://librosa.org/doc/main/generated/librosa.feature.chroma_stft.html [Accessed 7 Apr 2022].

[38] Jin, Jesse S.; Xu, Changsheng; Xu, Min (2013). The Era of Interactive Media || Better Than MFCC Audio Classification Features. , 10.1007/978-1-4614-3501-3(Chapter 24), 291–301. doi:10.1007/978-1-4614-3501-3_24

[39] Zhou, Xinquan & Lerch, Alexander. (2015). Chord Detection Using Deep Learning.

[40] Haggblade, M., Hong, Y., & Kao, K. (2011). Music Genre Classification.

[41] Yu, Yang; Luo, Sen; Liu, Shenglan; Qiao, Hong; Liu, Yang; Feng, Lin (2019). Deep attention based music genre classification. Neurocomputing, (), S0925231219313220–. doi:10.1016/j.neucom.2019.09.054

[42] Rajanna, Arjun Raj; Aryafar, Kamelia; Shokoufandeh, Ali; Ptucha, Raymond (2015). [IEEE 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA) - Miami, FL, USA (2015.12.9-2015.12.11)] 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA) - Deep Neural Networks: A Case Study for Music Genre Classification. , (), 655–660. doi:10.1109/icmla.2015.160

[43] Dong, Mingwen. "Convolutional neural network achieves human-level accuracy in music genre classification." arXiv preprint arXiv:1802.09697 (2018).

[44] Pelchat, Nikki; Gelowitz, Craig M. (2020). Neural Network Music Genre Classification. Canadian Journal of Electrical and Computer Engineering, 43(3), 170–173. doi:10.1109/CJECE.2020.2970144

[45] Liu, Xin, et al. "CNN based music emotion classification." arXiv preprint arXiv:1704.05665 (2017).

[46] Lee J, Park J, Kim KL, Nam J. SampleCNN: End-to-End Deep Convolutional Neural Networks Using Very Small Filters for Music Classification. Applied Sciences. 2018; 8(1):150. https://doi.org/10.3390/app8010150

[47] Zhang, Weibin, et al. "Improved Music Genre Classification with Convolutional Neural Networks." Interspeech. 2016.

[48] Nanni, Loris; Maguolo, Gianluca; Paci, Michelangelo (2020). Data augmentation approaches for improving animal audio classification. Ecological Informatics, 57(), 101084–. doi:10.1016/j.ecoinf.2020.101084

[49] Ko, Tom, et al. "Audio augmentation for speech recognition." Sixteenth annual conference of the international speech communication association. 2015.

.