Playing Style-Based Guitar Effects Control with Deep Learning

Sean Levine
slevine31@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia

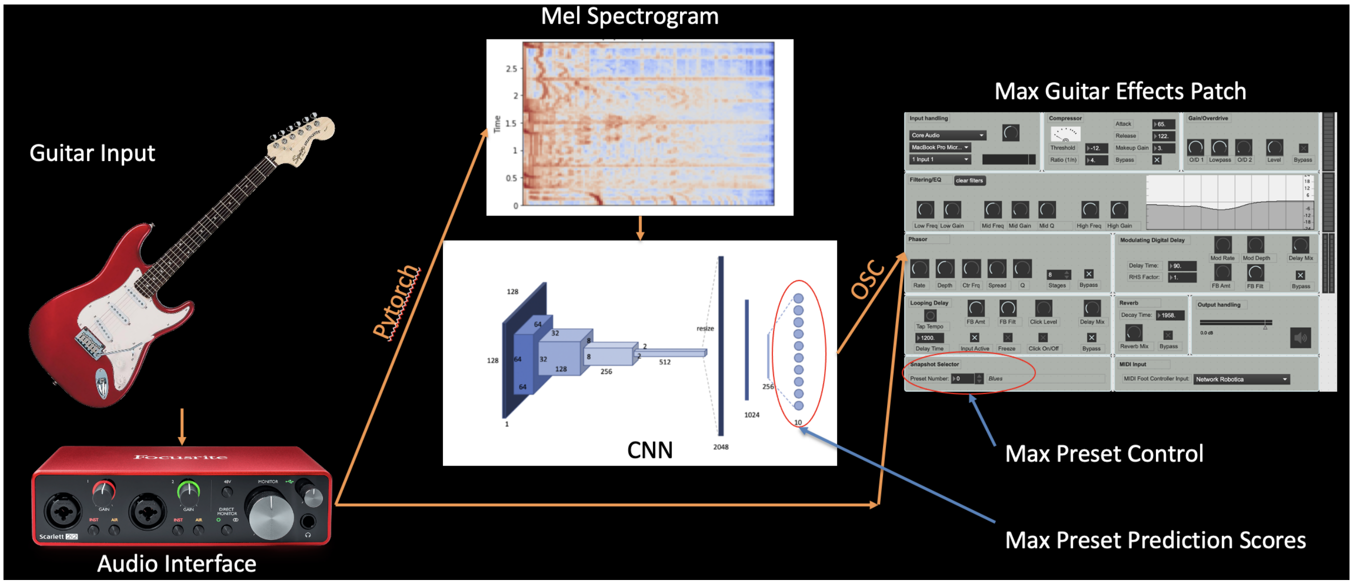

Figure 1: Full signal chain for the baseline guitar effects control interface with a CNN

KEYWORDS

playing style, neural networks, guitar effects

ACM Reference Format:
Sean Levine. 2022. Playing Style-Based Guitar Effects Control with Deep Learning. In *Proceedings of ACM Conference (Conference'23).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/

1 INTRODUCTION

Guitar effects sound design has been associated with guitar amplifiers since the early 1930s and guitar pedals in the early 1960s. Guitar pedals are foot-pedal boxes that provide sound effects for guitarists. They became an effective and compact way to electronically manipulate guitar sounds in ways that mechanical systems were not able to without considerable costs, such as the wah-wah effect and chorus effect. These guitar pedals eventually became embedded into digital amplifiers with long lists of tone presets that span various genres and combinations of pedals. These digital recreations of pedals and amplification are alternatives to the sound of analog electronics and vacuum tube amplification and are easily accessible within digital audio workstations or DAWs. Recent developments in deep learning and neural networks are able to accurately emulate analog pedals and vacuum tube amplification in a way that is indistinguishable from the original equipment [27]. While these technologies make these desirable sound design options more accessible, they do not advance the creative experience of the guitarist; maintaining the old system of manipulating pedal parameters to preset tones that the guitarist desires before performance or improvisation.

This project implements the creation of a guitar-playing style-based guitar effects control using deep learning that interacts with the guitarist in real-time. This system takes in the guitarist's input audio with no added effects, classifies the style that is being played, manipulates guitar pedal parameters based on the classification, and outputs the guitar performance with the newly added effects. The system classifies the guitar style by genre with a convolutional neural network or CNN, a class of artificial neural networks that are commonly used for music genre classification and recommender systems. While guitar playing style is commonly associated with the genre that is being played, the timbral texture caused by the addition of guitar effects is especially important for genre classification [3]. Unprocessed guitar genre classification is currently a gap in research. This project aims to determine feasibility by asking the question: How would a deep-learning model classify unprocessed guitar playing styles by genre, without relying on the timbral cues of effects?

Guitar pedal presets that are found in digital amplifiers are often labeled by the genre of music that the effects are most associated

Conference'23, May 2023, Atlanta, Georgia, USA

Proceedings of ACM Conference (Conference'23), https://doi.org/10.1145/.

with, so the CNN genre classification of the guitar playing is likely to be an effective method for determining guitar pedal choices that combine with the guitar playing in a sonically pleasing manner. The Guitar Rig 6 user library shows that many effects presets are categorized by genre [14] as well as the Line 6 Spider 6 digital amplifier presents as shown in Figure 2. Convolutional neural networks attain a high accuracy for predicting the genres of full instrumental mixes, but the accuracy of predicting genre or playing style for clean guitar is unknown. Regardless of the precision of the neural network at classifying unprocessed guitar by genre-associated labels, would the output of an interactive, dynamically changing guitar-effects system be meaningful or enhance creativity?

09A	Rock: Rhythm	Genre: Rock
09B	Rock: Clean	Genre: Rock
09C	Rock: Crunch	Genre: Rock
09D	Rock: Solo	Genre: Rock
10A	Metal: Rhythm	Genre: Metal
10B	Metal: Clean	Genre: Metal
10C	Metal: Djent	Genre: Metal
10D	Metal: Solo	Genre: Metal
11A	Indie: Rhythm	Genre: Indie
11B	Indie: Clean	Genre: Indie
11C	Indie: Lead	Genre: Indie
11D	Indie: Solo	Genre: Indie
12A	Blues: Rhythm	Genre: Blues
12B	Blues: Clean	Genre: Blues
12C	Blues: Lead	Genre: Blues
12D	Blues: Rotary	Genre: Blues
13A	Country: Trem	Genre: Country
13B	Country: Slap	Genre: Country
13C	Country: Lead	Genre: Country
13D	Country: Boost	Genre: Country
14A	Jazz: Solid State	Genre: Jazz

Figure 2: Line 6 Amplifier Effects Excerpt

If neural networks were found to be effective at classifying solo unprocessed guitar tracks, this could lead to further research into the classification of more individual instruments, their respective playing styles, and genres. These analyses can be used to create interactive systems that react to the playing style of the instrumentalist. Real-time automatic mixing and dynamic effects along with enhancing the complexity of automated music accompaniment can all benefit from development in individual instrument high-level classification. If the guitar classification system is not accurate at classification but creates fascinating outputs, the result would be a new interactive system that saves guitarists time and money on sound design compared to pedalboard design and multi-effects suites as well as enhances their creative process while practicing. This project would be another example of how deep learning is becoming more prevalent for music technology, changing how musicians interact with computers, and enhancing creative means of music production.

2 BACKGROUND

2.1 Feature Extraction

Feature extraction is a crucial step in many audio signal processing tasks, including music genre classification. The goal of feature extraction is to transform raw audio signals into a compact and representative feature space that can be used as input to machine learning and deep learning models [2]. Mel-frequency cepstral coefficients or "MFCCs" (Figure 3) are a popular hand-crafted feature that captures the spectral characteristics of an audio signal by mapping the frequency scale onto the mel scale, which is more perceptually relevant to human hearing. MFCCs are the discrete cosine transform or the DCT of the mel-spectrogram. In this research, mel-spectrograms are used as an input feature to a convolutional neural network (CNN) for music genre classification. By using mel-spectrograms (Figure 4) as an input feature to a CNN, the approach takes advantage of the CNN's ability to automatically learn hierarchical representations of audio signals without the need for manual feature engineering.

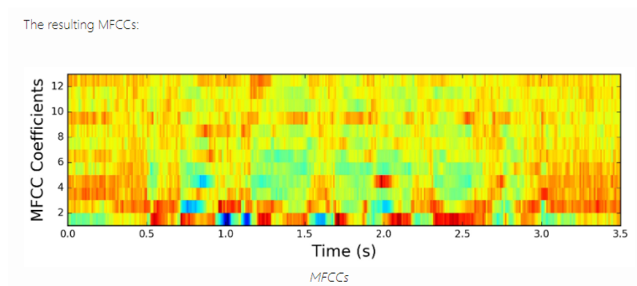

Figure 3: Example of an MFCC Spectrogram from [22]

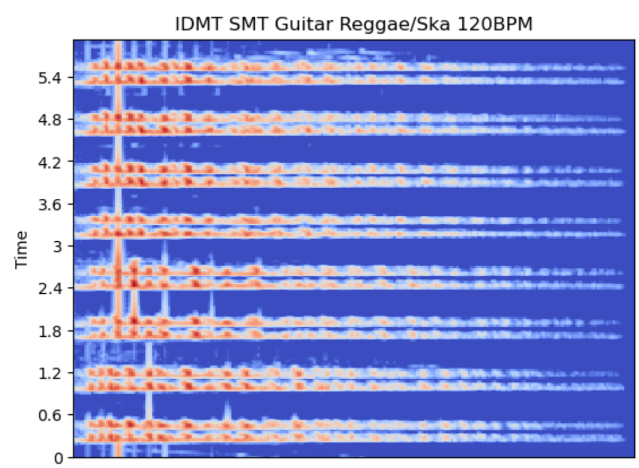

Figure 4: Example of an Mel Spectrogram of IDMT-SMT Guitar Data

2.2 Deep Learning

Deep learning is a subset of machine learning that uses neural networks of three or more layers to cluster data and make accurate predictions [1]. While machine learning requires experts to create feature extraction tools for machine learning algorithms to process and organize data, deep learning can take in unprocessed data and organize using backpropagation and gradient descent. Convolutional neural networks are a type of deep learning that is commonly used for computer vision, image classification, and music genre classification. Since MFCCs are filtered spectrograms that resemble an image matrix, they are useful input data for convolutional neural networks to detect patterns. Experimental results show that using MFCCs as input features to a CNN can achieve high accuracy in music genre classification tasks [8], particularly when combined with data augmentation techniques to improve the model's robustness to variations in pitch, tempo, and other factors. In a Convolutional Neural Network (CNN), three primary types of layers work together to process and analyze data: convolutional layers, dense (fully connected) layers, and softmax layers [21]. Convolutional layers are the building blocks of CNNs, responsible for feature extraction. They apply convolution operations to the input data, using multiple filters to detect specific patterns, such as edges or textures, at different spatial scales [5]. Dense (fully connected layers) are used to combine extracted features and make predictions. They connect every neuron in one layer to every neuron in the next layer, allowing the network to learn complex representations of the input data[11]. Softmax is an activation function typically used in the final layer of a classification network. It converts the network's output into a probability distribution over the target classes, allowing the model to provide a clear prediction by selecting the class with the highest probability [21]. These layers are often interleaved with other layer types, such as activation layers and pooling layers, to build a complete CNN architecture.

2.3 Guitar Effects

Guitar effects are commonly used in popular music to shape the tone of the guitar based on the genre of music that it is accompanying and or to add musical variety to the musical composition [13]. Audio effects are utilized for guitar and a multitude of other instruments including harp, saxophone, and vocals to aid in the creative process of music production. While many instruments are likely to use minimal effects to enhance the sound or make the instrument sound like it is playing in a particular room, guitars tend to have a combination of effects that make them distinctive and unique. Effects such as distortion, overdrive, phaser, delay, reverb, and compression have been the most used for guitar performances. During a guitar performance, these effects are used dynamically, turning certain effects on or off depending on the section of a musical piece or parameters of the effects being tweaked between songs to better match the accompaniment. The layout of the effects is important and when put in the wrong order can create an undesirable sound. An example guitar effects signal chain is shown in Figure 5. The signal chain can be simplified to be dynamics/pitch altering, nonlinear gain, modification of sound, sound replication, and ambiance creation. Modern guitar players use this layout to decide the way that they want to shape their tone. Although the money needed to recreate this effects chain can be above the budget of many learning guitarists, alternatives such as multi-fx pedals have been made to combine these pedals into one enclosure. These multi-fx pedals can also be confusing and above the budget of guitarists. Many guitarists have been turning to migrate their entire guitar signal chain through digital audio workshops on their personal computers to decrease cost and allow for more portability. However, running guitar effects on a computer makes impromptu interaction with the individual effects difficult.

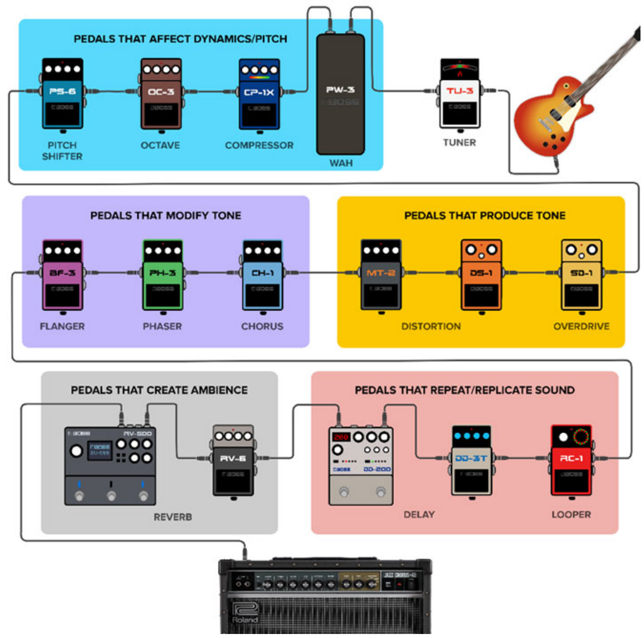

Figure 5: Common guitar effects signal chain [17]

3 LITERATURE REVIEW

3.1 Deep Learning for Guitar Amp Emulation

Audio classification, text-to-speech, audio source separation, and music generation are all complex systems that within recent years, have all been achieved through computer algorithms. The use of deep learning with the guitar has created new technologies that are accessible and enhance the creative process of guitar playing and tone shaping.

Neural DSP utilized WaveNet-ish neural networks to emulate nonlinear transforms of analog guitar amplifiers and guitar pedals [27]. The guitar before any added effects was compared to the guitar signal after being affected by the amplifier or pedal and the transfer function of the effect was accurately recreated through the neural network. The result of the emulation was indistinguishable from the original amplifiers and pedals. NeuralPi is an open-source version of this guitar effect emulation using a Raspberry Pi and a Stateless-LSTM neural network [4]. The result was similar to the work of NeuralDSP but this system was implemented on a much less computationally powerful interface, demonstrating that this system can be accessed with the recent advancements in personal computing. However, these systems are limited to emulating gain-related effects such as overdrive and distortion, this excludes effects such as reverb, delay, and phaser.

3.2 Live Computer Music Interaction

Live computer music interaction is a subject that has been investigated since the 1970s. With systems such as Voyager [16] that use complex algorithms to extract features from the input audio and process the data to create accompaniment, deep learning has allowed for further complex interaction similar to performing with another human. ML-Jam, developed by Google Magenta, allowed for structured improvisation with computer-pre-trained deep learning models [7]. The user inputs a melody line in real-time through a web app and the deep learning models analyze the MIDI data to create drums and a bassline using DrumsRNN and MelodyRNN [26] from Magenta. This system demonstrates that deep learning models can interact in real-time with user input to create meaningful outputs that enhance the user performance experience. ML Jam is an example of deep learning taking in pitch data and outputting accompanying pitch data but can a system that alters the timbre of the input data be effective and enhance creativity?

3.3 Deep Learning for Guitar Effect Parameter Approximation

Accurately determining guitar effect parameters from a full instrumental mix of music has been a challenging task, even with machine learning algorithms. This information is of great interest to many instrumentalists because it allows for definitive or similar replications of guitarist sound design. In other words, it enables musicians to recreate the sound of their favorite guitarists by replicating the effects and other processing applied to their guitar sound. Despite the difficulties in extracting this information, there is a growing need for automated tools that can analyze guitar audio and extract the relevant effect parameters, as this can greatly benefit the music production process and facilitate the creation of new music. Deep learning has shown to be an effective method of extracting these guitar effects parameters from full instrumental mixes compared to machine learning algorithms such as support vector machines (SVMs) [12]. Convolutional Neural Networks (CNNs) can achieve a 97.4% accuracy in estimating guitar effect parameters in full instrumental mixes for distortion, chorus, phaser, delay, reverb, overdrive, and tremolo. There were two CNNs, one trained on the GEC-GIM dataset [12] and the other trained on the GEPE-GIM dataset [12]. The GEC-GIM dataset consisted of guitar audio tracks that were clean and with added effects. This dataset is used to classify what guitar effects were active in the audio data. The GEPE-GIM dataset is a slight modification of the GEC-GIM dataset. The GEPE-GIM dataset consists of the same effects as the GEC-GIM dataset but has audio samples for each audio effect at intervallic values for each parameter (i.e. 0, 1, 2, 3... 127). Both the GEC-GIM and GEPE-GIM datasets are limited to single-note guitar tracks. This means that polyphonic guitar has not been tested on these systems to accurately estimate guitar effects parameters. While guitar effect parameter estimation has been shown to be effective with CNNs, the use of CNNs to control guitar effects for an input guitar signal has yet to be tested.

3.4 Machine Learning Model for Guitar Effect Control

The concept of a guitar effect that changes depending on the user's playing style has not been made into a commercial product. Tone shaping normally occurs manually by the player either by activating a guitar effect on a pedalboard or by adding effects or automating predetermined changes of effects in a DAW, however, there have been efforts to interpret and predict guitar effect changes using machine learning. In 2014, a YouTube video by jpgsloan demonstrates how to change guitar effects based on how the user plays using Max MSP and additional machine learning libraries [23]. The Max patch is a simple proof of concept that if the guitar player plays with a certain speed and loudness, there are guitar effects that accentuate the way the guitarist plays. When the guitarist plays slow and quiet, the reverb effect will be activated, when the user plays loud and slow, distortion and reverb will be activated and when the user plays quiet and fast, there will be effect activated. The Max patch processes the audio data to find when the volume of the guitar signal reaches a certain threshold and if the time difference between beat onsets is fast, then sends binary values to a machine learning Adaboost module to determine the class [24]. The class is finally routed to the corresponding pedals that need to be turned on when the guitar style is classified. Despite the interesting results of the patch, the demonstration shows the system had a large amount of latency and unwanted artifacts when the pedals were turned on. This patch was recreated in Max 8 for this project, as seen in Figure 6, to determine if these issues were still occurring. The recent recreation was able to mitigate the latency dramatically to become almost instantaneous but the artifacts when pedals were activated still occurred. Having a deep learning model that can linearly change between classes would be of interest to prevent these artifacts and possibly create interesting changes in sound over time that are done through periodic effect predictions.

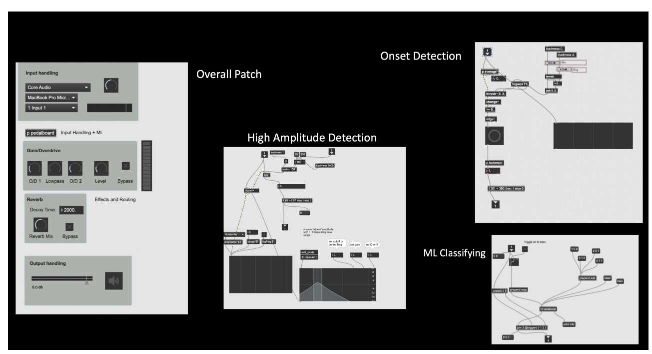

Figure 6: Prototype Rule-Based Guitar Effects Control in Max MSP

3.5 Music Genre Classification

Guitar pedal presets are often linked to specific genres and playing styles of music. Guitar playing styles are commonly distinguished by the genres that they are commonly played in as shown in the guitar styles guide by Berklee College of Music [19]. These presets are unique and can be identified by the genre of music they are associated with. Therefore, music genre classification algorithms were initially considered as a method to classify and predict unprocessed guitar tracks based on their playing styles. In other words, by training the music genre classification model on unprocessed guitar tracks that are labeled by genre and have enough variance in playing style by genre, then the classifier will predict the guitar pedal layout that best accompanies the genre and playing style. Music genre classification has been a widely researched topic, showing a multitude of neural network architectures is able to determine the genre of full instrumental mixes. Chenyu Xi [28] utilized methods from Keunwoo Choi [9] to classify music genre for the GTZAN dataset [25] with an accuracy of 88.05% with a Convolutional Neural Network (CNN), 85.08% with a Convolutional Recurrent Neural Network (CRNN) with 3 hidden layers, and 88.45% with a CRNN with 4 hidden layers. Since a CRNN can increase computational complexity significantly, a CNN became the baseline system for this project to allow access to more users and for its known proficiency in music genre classification. Despite CNNs showing the significant capability to classify genre for full instrumental mixes, individual instrument mixes especially without any effects processing may require tweaking of the architecture and input data to create accurate or significant predictions. The GTZAN dataset [25] includes 1000 full instrumental songs labeled with 10 genres, having a dataset of similar breadth will be needed to train the CNN effectively for individual guitar tracks. Luckily, there are datasets for raw guitar audio that span several genres. IDMT-SMT [15] is a large dataset for automated guitar transcription that includes a subset with 64 short musical pieces grouped by genre. Within this dataset is a subset that includes 512 unprocessed guitar tracks played at fast and slow tempos, with two different types of electric guitars, and two recording methods for an acoustic guitar. These guitar tracks are labeled into 8 genres, Rock/Blues, Reggae/Ska, Pop, Metal, Latin, Jazz, Country/Folk, and Classical. This project utilizes only the electric guitar tracks for a total of 256 guitar tracks to train the Convolutional Neural Network.

4 CNN FOR CLASSIFICATION OF GUITAR PLAYING STYLES

4.1 Preprocessing

Before the CNN is able to accurately classify guitar pedal layouts from guitar playing style, the input data processing will have to be altered to accept the new training data. Since the music genre classifier tends to eliminate the first and last 10-15 seconds of full instrumental mixes, when the same method is done for unprocessed guitar playing, it can reduce the amount of important training information for the neural network. The IDMT-SMT genre-separated dataset has a 44.1 kHz sampling rate in the uncompressed ".wav" format. Processing these large files takes much more time for the model to compute compared to compressing the file to a .mp3 and resampling at 16000 Hz. IDMT-SMT organizes their folders separating files by their genre, type of guitar used in recording, and tempo. This formatting allows for simplified genre-labeling of guitar tracks when the folder structure is reduced to separation by only genre. Once the folder structure is established and the data is formatted correctly, the training data is processed by the *librosa* [18] mel-spectrogram function in Python. Different frame lengths of the mel-spectrogram were tested but 128x128 was ultimately chosen because it was found to maintain genre classification accuracy while requiring less time data, allowing for faster succession between genre predictions. Each mel-spectrogram was paired with the labeled genre, for all 256 tracks, and the dataset was randomized and split into 0.7 train, 0.2 test, and 0.1 validation sets.

4.2 Neural Network Tuning

The Convolutional Neural Network (CNN) for this system has 4 hidden layers with rectified linear unit (RELU) activation and max pooling, 2 linear layers with 0.5 dropout, and a final softmax layer as shown in Figure 7. Different learning rates, batch sizes, and epoch numbers were investigated to allow for the model to prevent over-fitting and to accurately classify and predict the genre of guitar test data. The hyperparameters that allowed this were a learning rate of 1e-5, a batch size of 32, and an epoch number of 40. This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA [20].

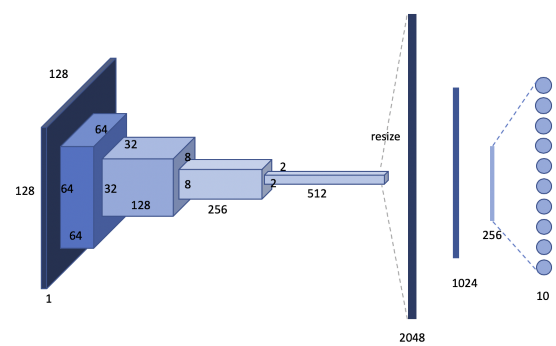

Figure 7: CNN Model Architecture Used for Classification

4.3 Additional Data Creation

During the initial testing of the classification model with guitar data outside the dataset, predictions were found to be highly inaccurate and dramatically altered by slight changes in the volume or mean amplitude of the audio file. The IDMT-SMT dataset was also shown to be limited by its 64 total various licks (short stock patterns of notes) and progressions. The 64 licks and progressions were replicated 4 times each to ensure models are robust to different tempos and guitar types. To mitigate these issues, this project created 64 new unprocessed guitar tracks with more variance in volume, and licks/performances. The tracks used in the study included an 8-beat count-in played by muted guitar strums, followed by 8-bar phrases or progressions that were either repeated once or slightly altered. These recording regulations were put in place to maintain consistency with the IDMT-SMT tracks. A total of 64 new tracks were created, with 8 tracks per genre, evenly split between the genres.

4.4 Classification Evaluation

The CNN is evaluated by measuring the testing and validation accuracy, and the binary cross-entropy loss over each epoch. To ensure the evaluation of the accuracy is valid, the precision, recall, and F1 score along with a confusion matrix was computed. Precision, recall, and the F1 score are standard methods for evaluating classification systems especially when the input data has a significantly common class that dominates the examples [6]. The confusion matrix is a helpful visualization to determine how well the prediction of the system was by showing important metrics such as the number of false positives and false negatives.

5 IMPLEMENTATION OF CNN AND SYSTEM DESIGN

5.1 Overall System Design

This project implements a playing style-based guitar effects control with deep learning. This system takes in the guitar audio input, extracts the mel-spectrogram of the audio, predicts the preferred pedal layout using a convolutional neural network, and routes the prediction values to a Max MSP guitar effects patch. The audio is streamed through the system simultaneously between Pytorch and Max MSP so the guitarist can interact with the system in real-time. The pedal layout predictions are the output of the CNN and these values will be sent from Pytorch to Max MSP over OSC where Max MSP appends the guitar effects onto the guitar. Figure 1 demonstrates the overall signal flow of the proposed system.

5.2 Effect Control

The guitar effects Max patch is implemented in Max MSP 8.1.8 and follows the tutorial made by Cycling '74 [10]. The Max patch includes important effects that follow common guitar pedal signal chain practices, compression, overdrive/distortion, filtering/EQ, phaser, delay, reverb, and output volume control. An important addition to the system that is not the guitar effects is the ability to change and save effect presets. The currently available guitar effect presets are based on the genre labels used in the IDMT-SMT dataset. These presets include Rock/Blues, Reggae/Ska, Pop, Metal, Latin, Jazz, Country/Folk, and Classical. The sound design for these effects was chosen based on popular effects that are used for these genres and the correlated tonal qualities for the genre that are achieved through equalization. Users are able to have full control of these presets including saving, editing, and creating new presets. However, as mentioned in the next section, the CNN interpolation for this system does not have direct access to presets or Max snapshots and is based on the presets made during the creation of this project.

5.3 Semi Real-time Implementation

To allow the user to be able to use the system continuously, audio is streamed to Max MSP and Python. Max MSP allows the guitar signal to play regardless of processes in Python. Pyaudio streams the audio at 2048 frames per buffer, compresses the audio to mp3 every 10 seconds, computes the mel-spectrogram for this excerpt, and inputs the mel-spectrogram into the Convolutional Neural Network. The CNN outputs the genre prediction for each mel-spectrogram frame and if the input into the CNN has multiple frames, then it outputs all predicted genres and the percentage of frames for each genre in order of prevalence. Each genre preset parameter value for each genre is saved into a dictionary within Python. This includes the reverb mix, the amount of distortion, and EQ values for each Max preset. Once the prediction is finished, the dictionary for each genre that is predicted is accessed. The predicted genre dictionary key values are all weighed by the genre prevalence percentage and added together to create the interpolated parameter values. These interpolated parameter values are sent through OSC to Max with differing address names so Max can parse these messages and append the corresponding value to each effect. Changing the parameter values simultaneously is disorienting for the performer, so there is a fade-in transition of 2 seconds from the previous value to the new value for smoother interaction.

5.4 Evaluation of the Interactive System

To determine if this interactive system is working as hypothesized the following tests were performed. External data classification validation to ensure that the classification of the system correctly classifies guitar input for different types of guitars at different volumes. External data interpolation validation, ensuring that genre-blending test audio outputs mixed parameter values that are sent to Max MSP. Audio to effect consistency, to determine that the audio being tested is consistent with the effects in a satisfying combination.

5.5 User Evaluation

To evaluate the overall system, a user experience questionnaire was designed focusing on the aspects of enjoyment, creativity, and expression. Due to time constraints, the author of this paper was the sole participant in the evaluation.

The questionnaire consisted of four open-ended questions:

- 1. Did the system inspire you to play in ways you normally would not?
- 2. Does the system make practicing more fun?
- 3. Do the changes in effects feel natural or do they distract from your playing?
- 4. Does the system add effects that are similar to your expectations or are they unpredictable?

The author answered the questions based on the experience with the system after 30 minutes and provided qualitative feedback. While the sample size of one limits the generalizability of the findings, the results offer valuable insights into the participant's experience with the system.

However, it should be noted that self-evaluation may introduce bias, as the author was aware of their own expectations and experiences with the system. Future studies should aim to recruit a larger sample size and use objective measures to evaluate the system's performance. Additionally, the open-ended nature of the questionnaire may limit the ability to compare results across participants or studies. Future studies could consider using standardized questionnaires with Likert scales to allow for quantitative analysis and comparison of results. As a whole, while the current study provides initial insights into the user experience of the Max effects patch, further research is needed to fully evaluate its potential.

6 RESULTS

This project includes the creation of an interactive music system that controls guitar effects based on the playing style of the player using deep learning. The project aims to determine if deep learning is capable of accurately classifying solo unprocessed guitar tracks by genre and if the predictive change in guitar effects based on guitar playing style is beneficial as well as meaningful to the creativity of guitarists. The CNN model evaluation before and after additional unprocessed guitar tracks were included in the dataset is indicated in this section along with the user evaluation and interactive system validation.

6.1 CNN Model Results Before Data Creation

To determine if the Convolutional Neural Network (CNN) is able to accurately and precisely classify the genre of unprocessed guitar tracks, the accuracy per epoch, binary cross entropy loss per epoch, model accuracy between genres, precision, recall, f1 score, and confusion matrix were all computed and analyzed.

Figure 8 demonstrates the accuracy results for the CNN before additional data was created. The training set achieved an accuracy of 93.1% and 87.5% for the validation set. Despite this acceptable accuracy level, the validation set did not include data from outside the IDMT-SMT dataset and therefore not thoroughly cross-validated. The binary cross entropy loss (BCE loss) for the model before the additional training data is shown in Figure 9. The ideal loss value for a model is 0, indicating perfect predictions. The validation set loss follows a similar path to the training set loss indicating that the system is not overfitting to the training data but to the method of recording used for the IDMT-SMT dataset along with the genre-discriminating decisions for playing style. The training set loss and validation loss at the 50th epoch were 0.045 and 0.075, respectively. This indicates that the predictions are not ideal but the model is working correctly. Since some genres are known to have similar sound profiles such as Pop with Rock/Blues or Jazz with Latin, the accuracy between genres is plotted in Figure 10. All genres performed with accuracy above 90% indicating the model was able to make fine distinctions between closely related genres or playing styles.

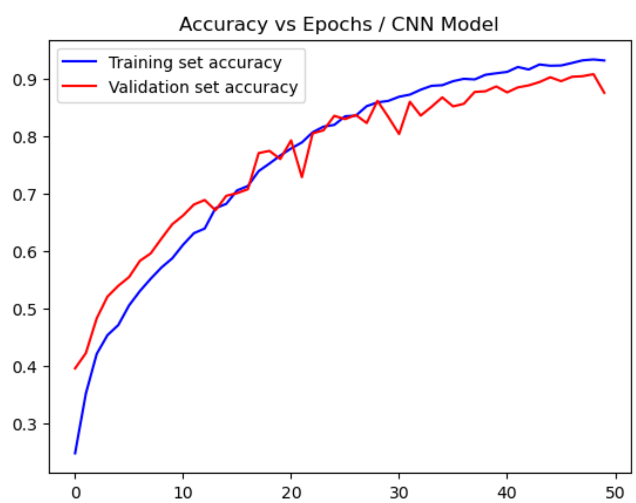

Figure 8: CNN Accuracy per Epoch Before Added Data

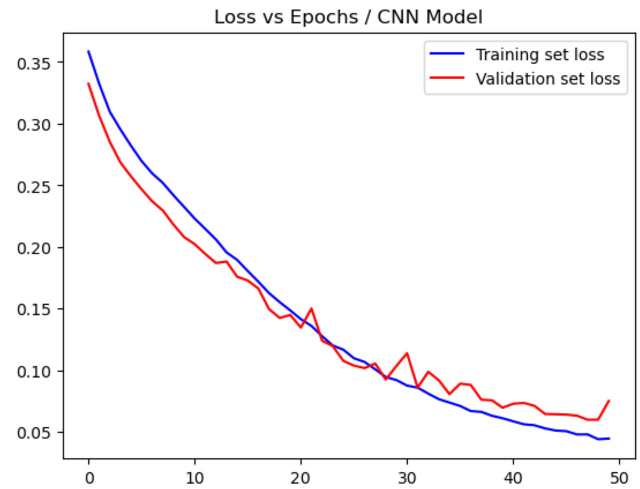

Figure 9: CNN Loss per Epoch Before Added Data

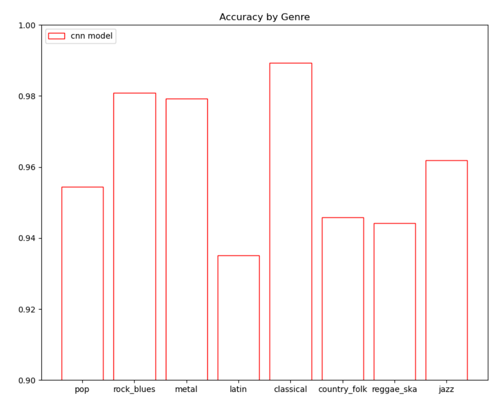

Figure 10: CNN Accuracy for each Genre Before Added Data

If the model were to perfectly accurately classify the guitar tracks, the precision, recall, and F1 score would all be 1. The first training of the model had a precision of 0.88, a recall of 0.89, and an F1 score of 0.88. The confusion matrix is plotted in Figure 11 to aid in finding which genres tended to get misinterpreted by each other. The bright values on the diagonal of the matrix indicate high true positives and high true negatives for the classifier. The model falsely predicted metal the most when it was analyzing reggae/ska data, likely due to the fast strumming speed and use of rhythmic syncopation. The model had the most difficulty classifying pop likely due to pop being a blend of styles that becomes popularized such as rock, classical, and hip hop in the modern day. Overall, this confusion matrix does not have any large values for false positives or false negatives further demonstrating that the model is able to distinguish unprocessed guitar tracks by genre/playing style.

However, this iteration of the model failed to accurately classify unprocessed guitar tracks from outside the model such as created guitar data from a Fender Stratocaster through a Scarlett 18i20.

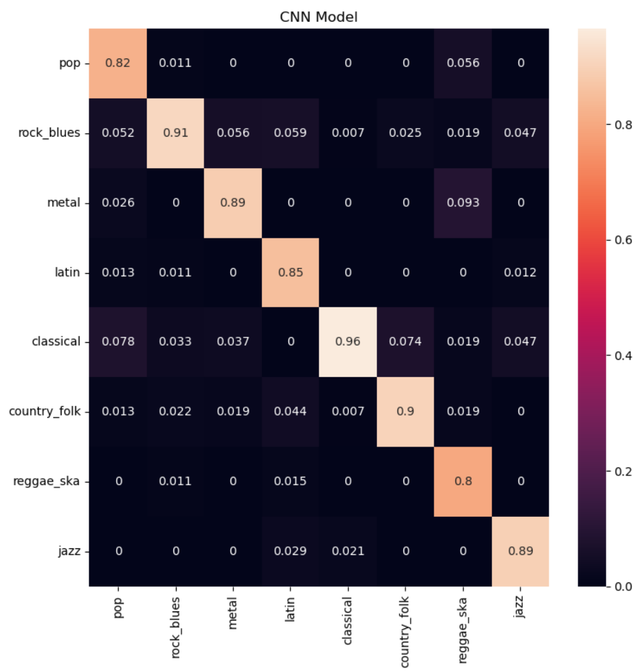

Figure 11: CNN Confusion Matrix Before Added Data

6.2 CNN Model Results After Data Creation

Following the addition of 64 unprocessed guitar tracks made for this project, the dataset was rebuilt and shuffled. The CNN was fit to this new dataset and the accuracy per epoch, loss per epoch, precision, recall, F1 score, and confusion matrix were calculated to verify that the model was predicting the guitar tracks as anticipated.

To prevent overfitting, the CNN model training after new data was added to the dataset decreased in total epochs from 50 to 40. This reduced the overall accuracy of the model relative to the dataset but improved the accuracy of predicting the genre of test data from outside the dataset. This reduction in accuracy for the dataset is shown in the decrease in overall accuracy for the training and validation set compared to the previously trained model. The training accuracy was 91.3% while the validation set accuracy was 85.6%.

The validation set loss and accuracy do not fluctuate as rapidly compared to the previous model training. This is likely due to the validation set being more representative of the whole dataset. The losses found at the final (40th) epoch of this training were 0.066 for the training set and 0.081 for the validation set. Although this demonstrates that the accuracy and loss are found to be decreasing with the addition of data and the decrease in epochs, the overall performance of the interactive system improves from the increased knowledge about different types of guitar progressions and phrases that could be used to test the system. The reduction in overfitting

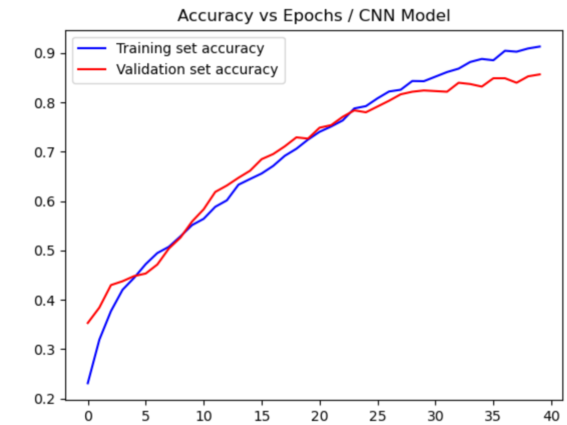

Figure 12: CNN Accuracy per Epoch After Added Data

to the dataset also ensures that the predictions are not based on abstractions of the training data that are arbitrary and only pertain to the training data.

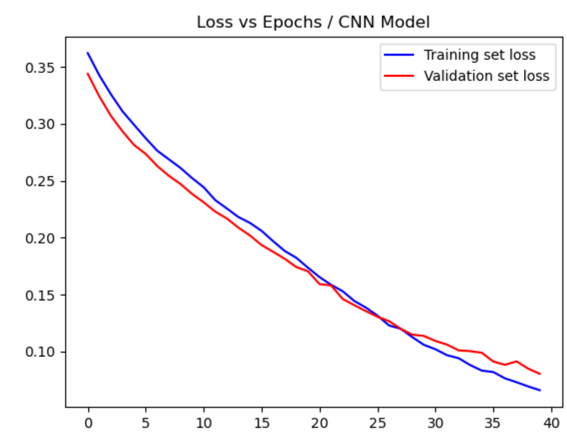

Figure 13: CNN Loss per Epoch After Added Data

Figure 14 shows that overall accuracy by genre decreases after new guitar tracks were added. This is especially true for pop and metal.

The training of the model with the additional guitar tracks performance metrics was a precision of 0.870, a recall of 0.876, and an F1 score of 0.873. While some true positives for the confusion matrix improved compared to the first training, the genre of metal had the largest decrease in accuracy with country/folk next. These genres likely decreased in accuracy due to their reliance on the overall volume of these genres being louder than the rest, an important feature for prediction. Since the additional data introduced more amplitude-related variance for each genre, these genres are harder to distinguish from the other genres than before. Rock/blues

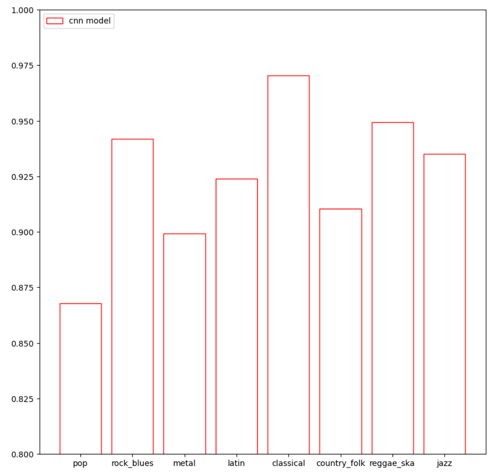

Figure 14: CNN Accuracy for each Genre After Added Data

was also found to be falsely predicted the most in this new training this could be due to the bias that the players asked to record for the additional data were primarily rock/blues guitarists.

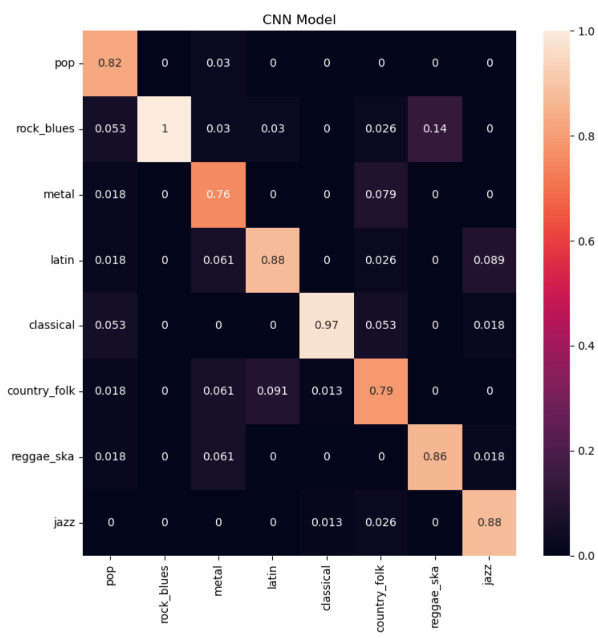

Figure 15: CNN Confusion Matrix after Added Data

6.3 Overall System Results

The overall interactive system was tested on its performance using the following tests, external data classification validation, external data interpolation validation, and audio to effect consistency. These tests focus on improving the human-computer interaction of the system and determining if the system functions as hypothesized.

6.3.1 External Data Classification. After the first training of the CNN model, the external data classification was tested. External data classification was done by recording a guitar track with a known genre that is not in the training data and sending it to the CNN classifier. The guitar used is a different model and different volume levels were also tested to ensure that the classification of the model is not volume dependent. The first CNN training was shown to falsely classify data from outside the dataset and for this reason, the additional guitar track data was included in the dataset. Once this additional data was added, further small tweaks to the classification model such as hyperparameters were made to improve the accuracy in classifying external guitar recordings. While the end performance of the classification is not perfect, the predictions of the system are of the correct genre or a genre that utilizes similar effects and effect parameters (i.e. jazz and latin). The model also is able to distinguish when a guitarist is using a certain pickup, either closer or farther from the guitar neck, and predicts a genre that uses this pickup more often. This test shows that the system is capable of making meaningful predictions from external guitar data.

6.3.2 External Data Interpolation. The external data interpolation validation test is done by recording a guitar track with multiple known genres and analyzing whether the classifier outputs percentages approximately or equal to the known genre split of the track. For example, smooth jazz was used to evaluate if the classifier would output a genre split of pop and jazz. This particular result was 27.4% jazz, 61.2% pop, and 11.4% rock/blues. When the effect parameters were interpolated in Max MSP, the overall effect and sound design matched expectations. Across a total of 6 examples, the interpolated effects were found to match the guitar tracks well. Minor alterations of parameter values were made for some examples due to personal preference, but these were minute compared to entire preset changes. The interpolation of genre effects allowed for a decrease in the time needed for the user to manipulate effect parameters compared to other current guitar effect systems.

6.3.3 Audio to Effect Consistency. During the creation of the presets for the guitar effects Max patch, popularized guitar track examples for each genre were used to tune preset templates. While certain genres have subgenres or notable artists that vary in effect choices, baseline effects and effect parameters were chosen. To maintain consistency and allow a variance for each genre, effect parameters were kept from going to extremes. While consistency for the effect template to blend with the guitar track is meaningful, this restriction of tone decreases the amount of difference in tone between genre presets and does not highlight the power of the system. However, if this system is trained based on particular guitar tracks that pair well with extreme or highly distinguishable guitar effects in future work, consistency validity would be more pronounced and the overall system would be more intriguing.

6.4 User Evaluation

The user experience questionnaire provided valuable insights into the user experience with the overall interactive system. The system was found to inspire playing in ways they normally would not and make practicing more fun. However, the system is complicated with limited user control for aspects such as the time between changes in effect and turning on and off the playing-style predictions.

Regarding creativity and expression, the system allowed the creation of certain effects layouts that they would take much longer by changing each parameter individually. There is a possibility of frustration with the unpredictability of the system's effects since continuously predicts even during silences.

In terms of enjoyment, the system was generally enjoyable to use and rarely distracted from playing. The interface could be improved to make it more intuitive and user-friendly. Some of the system's effect layouts were not very distinctive from each other making the interaction with the effects less exciting.

While self-evaluation limits the generalizability of the findings, the results do suggest some areas for improvement in the overall interactive system. Specifically, future iterations of the system could focus on enhancing the predictability and user-friendliness of the effects, while also ensuring that the effect layouts are more distinctive and personalized. These findings highlight the potential for automated style-based guitar effects to enhance the playing experience but also emphasize the importance of user-centered design in creating effective and enjoyable systems.

7 CONCLUSION

Overall, the results of the creation of this system show that Convolutional Neural Networks can classify unprocessed guitar playing styles by genre or playing style, without relying on the timbral cues of effects. This is shown by the accuracy, losses, and performance scores of the trained model along with the test results of external guitar data. The ability to determine genre or playing style from individual instruments is useful information for advancing tasks such as automated mixing and dynamic effects along with enhancing the complexity of automated music accompaniment. This research enables further development of interactive systems that react to dynamic playing style changes of individual instruments.

With the correct amount of precautions to prevent overfitting to the dataset, the deep learning model is even able to output interesting classification predictions for data that is outside the dataset and not constrained to individual classifications. The CNN along with the Max effects patch creates a combined interactive, dynamically changing guitar-effects system that was able to interpolate meaningful effect layouts. These effect layouts followed expectations of tone choices that meshed well with various guitar playing styles, were intuitive to the player, and were rarely distracted from the playing experience. Despite the promise of the project, there are areas of improvement for user-centered design such as user control of when and how the system predicts playing style. Automated style-based guitar effects can enhance the playing experience of improvisers, recording sessions, and even general guitar practice. This system demonstrates a need for further research on automated control of sound effects for instruments using music information retrieval methods and features from the said instrument.

8 DISCUSSION

The broader implications of these findings are that technology can enhance the playing experience of musicians, particularly guitarists, by creating dynamic and interactive effects systems. This system has the potential to increase creativity, facilitate learning, and provide new avenues for musical expression. Additionally, the research highlights the importance of music information retrieval methods and features from instruments in the development of automated control of sound effects. A possible improvement on the given project would be to design a dataset that works for particular effect presets rather than the generalization of the genre. Having a user append their own training data to allow the model to work with how they play would be an impressive feat in guitar effect personalization and workflow.

One gap that remains unknown or unsolved is how this system may impact different levels of guitar players, particularly beginners. It is possible that the system may be overwhelming or distracting for beginners, and future research should explore how to create a simplified version of the system that can be used by guitar players at different skill levels. This simplification could be implemented in a more commercially accessible format such as a VST or a desktop application. It is unknown how this type of system can be applied to other instruments beyond the guitar, and how different playing styles and techniques could affect the effectiveness of the system. Additionally, it is important to explore the potential limitations of the system, such as how it might struggle to adapt to non-standard playing techniques or unconventional instrument configurations.

While the study mentioned user evaluations, it is important to conduct further studies to gather more data on the effectiveness of the system in different contexts, such as in live performances or recording sessions. It would also be useful to gather feedback from a larger sample size to determine if the system is universally effective or if there are certain factors that may impact its effectiveness for different individuals.

Future work around this topic should focus on developing more advanced and sophisticated automated control of sound effects for instruments using music information retrieval methods and features. Additionally, further research should explore the potential of this technology to enhance the playing experience of other instruments beyond the guitar.

This project has been a novel investigation to develop and implement an automated style-controlled effects system and its potential impact on the playing experience of guitarists. This system demonstrates the potential for further research on automated control of sound effects for instruments using music information retrieval methods and features from the instrument. The findings from this research contribute to the broader field of music technology and sound engineering, providing insights into the design and implementation of neural networks for user interaction with music systems.

Overall, there is still much to be explored in the realm of automated control of sound effects for instruments, and this project provides a promising starting point for future research.

REFERENCES

[1] IBM . 2022. What is Deep Learning? https://www.ibm.com/cloud/learn/deep-learning publisher: IBM.

[2] Ethem Alpaydin. 2016. *Introduction to Machine Learning: Cognitive Representations as Graphical Models* (3rd ed ed.). MIT Press, The, Cumberland. OCLC: 1024244854.

[3] Babu Kaji Baniya, Deepak Ghimire, and Joonwhoan Lee. [n. d.]. Automatic music genre classification using timbral texture and rhythmic content features. In *2015 17th International Conference on Advanced Communication Technology (ICACT)* (2015-07). 434–443. https://doi.org/10.1109/ICACT.2015.7224907 ISSN: 1738-9445.

[4] Keith Bloemer. 2021. Neural Networks for Real-Time Audio: Raspberry-Pi Guitar Pedal. https://towardsdatascience.com/neural-networks-for-real-time-audio-raspberry-pi-guitar-pedal-bded4b6b7f31

[5] Jason Brownlee. [n. d.]. *How Do Convolutional Layers Work in Deep Learning Neural Networks?* https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/

[6] c3.ai. 2022. Artificial General Intelligence. https://c3.ai/glossary/data-science/artificial-general-intelligence/

[7] Pablo Samuel Castro. 2019. Performing Structured Improvisations with pre-trained Deep Learning Models. (2019). https://doi.org/10.48550/ARXIV.1904.13285

[8] Hasan Can Ceylan, Naciye Hardalaç, Ali Can Kara, and Fırat Hardalaç. [n. d.]. Automatic Music Genre Classification and Its Relation with Music Education. 11, 2 ([n. d.]), 36. https://doi.org/10.5430/wje.v11n2p36

[9] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. Transfer learning for music classification and regression tasks. (2017). https://doi.org/10.48550/ARXIV.1703.09179

[10] Darwin Grosse. 2021. Tutorial: Max 8 Guitar Processor, Part 1 | Cycling '74. https://cycling74.com/tutorials/max-8-guitar-processor-part-1

[11] MK Gurucharan. [n. d.]. *Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network*. https://www.upgrad.com/blog/basic-cnn-architecture/

[12] Reemt Hinrichs, Kevin Gerkens, Alexander Lange, and Jörn Ostermann. 2022. Convolutional neural networks for the classification of guitar effects and extraction of the parameter settings of single and multi-guitar effects from instrument mixes. *EURASIP Journal on Audio, Speech, and Music Processing* 2022, 1 (Oct. 2022), 28. https://doi.org/10.1186/s13636-022-00257-4

[13] Reemt Hinrichs, Kevin Gerkens, and Jörn Ostermann. 2022. Classification of Guitar Effects and Extraction of Their Parameter Settings from Instrument Mixes Using Convolutional Neural Networks. In *Artificial Intelligence in Music, Sound, Art and Design: 11th International Conference, EvoMUSART 2022, Held as Part of EvoStar 2022, Madrid, Spain, April 20–22, 2022, Proceedings*. Springer-Verlag, Berlin, Heidelberg, 101–116. https://doi.org/10.1007/978-3-031-03789-4_7

[14] Native Instruments. 2022. User Libraries : Guitar Rig | Community. https://www.native-instruments.com/en/community/user-libraries/guitar-rig/ publisher: Native Instruments.

[15] Christian Kehling, Jakob Abeßer, Christian Dittmar, and Gerald Schuller. 2014. Automatic Tablature Transcription of Electric Guitar Recordings by Estimation of Score- and Instrument-Related Parameters. In *International Conference on Digital Audio Effects*.

[16] George E. Lewis. 2000. Too Many Notes: Computers, Complexity and Culture in *Voyager*. *Leonardo Music Journal* 10 (Dec. 2000), 33–39. https://doi.org/10.1162/096112100570585

[17] Ed Lim. 2016. How to Guide: Signal Chain for Your Pedalboard. https://rolandcorp.com.au/blog/order-effects-chain-simple-guide

[18] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. [n. d.]. librosa: Audio and Music Signal Analysis in Python. 18–24. https://doi.org/10.25080/Majora-7b98e3ed-003

[19] John Mirisola. [n. d.]. *Guitar Playing Styles to Explore and Master*. https://college.berklee.edu/guitar/news/guitar-playing-styles-to-explore-and-master

[20] PACE. 2017. *Partnership for an Advanced Computing Environment (PACE)*. http://www.pace.gatech.edu

[21] Adrian Rosebrock. [n. d.]. *Convolutional Neural Networks (CNNs) and Layer Types*. https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/

[22] Tanveer Singh. 2019. MFCC's Made Easy. https://medium.com/@tanveer9812/mfccs-made-easy-7ef383006040

[23] John Sloan. 2014. Pedalboard - machine learning guitar pedal. https://www.youtube.com/watch?v=69r0a5oLbiY

[24] Benjamin Smith. 2022. ml.* - Machine Learning Toolkit in Max. https://www.benjamindaysmith.com/ml-machine-learning-toolkit-in-max

[25] Bob L. Sturm. 2013. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. (2013). https://doi.org/10.48550/ARXIV.1306.1461

[26] Magenta Team. [n. d.]. *MelodyRNN magenta/magenta*. https://github.com/magenta/magenta

[27] Alec Wright, Eero-Pekka Damskägg, Lauri Juvela, and Vesa Välimäki. 2020. Real-Time Guitar Amplifier Emulation with Deep Learning. *Applied Sciences* 10, 3 (Jan. 2020), 766. https://doi.org/10.3390/app10030766

[28] Chenyu Xi. 2021. Music-Genre-Classification. https://github.com/XiplusChenyu/Musical-Genre-Classification publisher: GitHub.