Audio Engineering Society

Convention Express Paper

Presented at the AES 158th Convention
2025 May 22–24, Warsaw, Poland

Guitar Audio Dataset for AI

Monika Zarudzka[1], Cyprian Zasada[1], Piotr Krzyżanowski[1], Klaudiusz Muszyński[1], Maciej Trocha[1], and Szymon Zaporowski[2]

[1]*Gdańsk University of Technology, Faculty of Electronics, Telecommunication and Informatics*
[2]*Gdansk University of Technology, Faculty of Electronics, Telecommunication and Informatics, Department of Multimedia Systems*

Correspondence should be addressed to Szymon Zaporowski, Cyprian Zasada or Monika Zarudzka
(`szymon.zaporowski@pg.edu.pl, s193222@student.pg.edu.pl, s193276@student.pg.edu.pl`)

ABSTRACT

Classifying musical instruments within the same family presents a significant challenge that remains relatively underexplored. To address this issue, we introduce G.A.D.A. (Guitar Audio Dataset for AI), a novel open-source dataset curated to support research in guitar audio analysis, signal processing, and machine learning applications. This comprehensive corpus consists of recordings from three primary guitar categories—electric, acoustic, and bass guitars—with multiple instruments represented within each category to ensure diversity and robustness.

To evaluate the utility of G.A.D.A. in machine learning contexts, we propose a testing framework incorporating established algorithms, including k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Feed-Forward Neural Networks (FFNN). Our experiments explore both traditional audio feature-based classification and deep learning approaches, achieving classification accuracies of up to 99%.

1 Introduction

The guitar, owing to its adaptability across diverse musical genres, has become one of the most widely used and versatile instruments. The vast variety of guitar models and manufacturers often fuels debates among musicians and enthusiasts regarding the superiority of specific instruments or brands. These discussions naturally lead to the question: Are the differences among guitar models objectively distinguishable based on their audio characteristics?

In this context, we present G.A.D.A., a standardized and publicly available dataset tailored for the classification and analysis of guitar audio. The dataset aims to bridge the gap between subjective perceptions of guitar tone and objective, data-driven insights.

G.A.D.A. distinguishes itself through a comprehensive approach to various guitar types, encompassing three principal categories: electric guitars, acoustic guitars, and bass guitars. This multi-instrument taxonomy represents a significant departure from previous datasets, which typically focus on a specific instrument category. For instance, Guitar-TECHS [1] exclusively contains electric guitar recordings, GuitarSet [2] primarily concentrates on acoustic guitar samples, while specialized collections such as the Bass Guitar Dataset [3] are dedicated solely to bass guitar audio.

Provided dataset implements a differentiated approach to audio acquisition based on instrument classification like employing direct recording methodology utilizing DI (Direct Injection) boxes for electric and bass guitars, yielding clean, unprocessed signals optimal for subsequent digital signal processing. This methodological framework contrasts with approaches employed in other datasets, such as Guitar-TECHS[1], which utilizes four distinct recording configurations for all instruments: an egocentric microphone positioned on the performer's head, an exocentric microphone placed in front of the performer, direct computer input, and an amplifier-proximate microphone placement.

G.A.D.A. has been specifically engineered for musical instrument classification tasks within the same instrumental family (guitars). This targeted design philosophy differentiates it from alternative datasets that predominantly focus on guitar transcription (Guitar-TECHS[1], GuitarSet[2]), analysis of guitar-specific techniques, or guitar solo detection algorithms. The dataset's architecture facilitates the development of computational models capable of discriminating between subtle timbral variations across different guitar types and models.

As stated above, electric and bass guitars were recorded using direct recording techniques via DI boxes, providing clean, unprocessed signals ideal for further digital processing and manipulation. For acoustic guitars, where direct recording was not feasible, we utilized multiple microphone configurations at various positions to capture the complete acoustic properties of the instruments. After performing sound checks, we chose a condenser microphone - Neumann TLM 103 as it was recording sound the closest to real-life hearing experience. Both recording approaches prioritize signal quality while maintaining maximum flexibility for subsequent processing and analysis.

The dataset includes standardized recordings of major and minor chords played in multiple positions and voicings across all electric and acoustic instruments. For the bass guitars only single tones were recorded as recording chords is impractical for this type of guitar. Each recording is accompanied by detailed metadata, including instrument specifications, tone and volume potentiometer configurations as well as transducer setting sand chord information. The clean signals from electric instruments enable various post-processing applications, including virtual amplifier modeling, effects processing, impulse response convolution, and room acoustics simulation.

G.A.D.A. will be freely available for academic and research purposes, complete with documentation, preprocessing scripts, example code, and usage guidelines. This resource aims to facilitate research in musical instrument classification, audio signal processing, deep learning applications in music technology, computer-aided music education, and automated music transcription systems.

The combination of standardized recording methodologies, comprehensive metadata, and the inclusion of both direct-recorded and microphone-captured audio makes G.A.D.A. a valuable resource for comparative studies and reproducible research in music information retrieval and audio processing.

Our ML-classifiers' implementations and guitar recordings are open-source, and can be found in the following GitHub repository: https://github.com/cyptrix12/Guitar-Recording-Classifier

2 Methods

2.1 Dataset Description

The recorded electric guitar models are as following:

1. Arirang SG - an iconic model and shape of electric guitar designed and introduced to the market by Gibson. Appreciated for its versatility and lightness.

2. Harley Benton TE-52 NA - a guitar with an American ash body with a warm and balanced sound. It has characteristic bright and glassy tones. Control over the sound is provided by tone and volume potentiometers and a 3-position switch.

3. Epiphone SG - 2 different copies were recorded - a guitar that stands out for its classic shape, which gained popularity thanks to iconic rock artists. It is valued for its ease of playability, as well as versatility, working well in both rock music and more diverse genres.

4. Harley Benton Les Paul - offering a characteristic, full sound with a clear emphasis on low and mid tones. Tone and volume potentiometers and a 3-way switch allow for precise control over the sound, allowing for a wide palette of tones.

5. Harley Benton Telecaster - a guitar inspired by one of the most iconic models in music history, known for its bright, cutting sound with a pronounced attack. Thanks to its simple yet effective construction, it delivers a sharp tone ideal for country, rock, blues, and indie styles. Its two single-coil pickups and 3-way switch offer a wide range of classic, expressive sounds.

6. Harley Benton Stratocaster - a versatile instrument modeled after the legendary Stratocaster design, offering a balanced tone with sparkling highs, clear mids, and tight bass. Equipped with three single-coil pickups and a 5-way selector, it provides a wide tonal palette perfect for genres ranging from blues and rock to funk and pop. Its comfortable body shape and smooth playability make it a favorite among guitarists of all levels.

Bass guitars:

1. Quintus Bass SG - a bass guitar inspired by the iconic SG design, offering a lightweight, double-cutaway body for comfortable playability. Known for its affordability, it serves well in practice settings and casual performances.

2. Epiphone SG - a bass guitar that pay tribute to the classic Gibson SG models.

The recorded acoustic guitar models are as following:

1. Vintage V300VSB - an acoustic guitar known for its compact folk body shape, offering a comfortable playing experience and a bright sound.

2. T. Burton J-0 BS - an acoustic guitar featuring a smaller, parlor-style body designed for easy handling and a delicate, balanced sound.

For each electric and acoustic guitar, a complete set of diatonic chords was recorded, considering various transducer settings, while tone and volume knobs were set to maximum. The specific settings for each recording are reflected in the file names. To enhance the accuracy and reliability of the dataset, each chord and setting combination was recorded 10 times.

For each bass guitar, only single tones were recorded (no chords). Each string was recorded up to the 4th fret, resulting in 20 distinct tones per instrument. The file names indicate the string used for each recording. As mentioned above, each tone and setting combination was also recorded 10 times.

2.2 Recordings

The electric and bass guitar recordings were conducted using a Focusrite Scarlett Solo audio interface connected to a computer via USB. Recording were made with 32-bit resolution and 48 kHz sampling frequency to balance disk space requirements with audio fidelity. The 32-bit depth ensures precise signal representation, which is especially advantageous. Instruments were connected via mono jack cables directly to the interface, without the use of amplifiers or other signal-modifying devices, to ensure clean DI recordings.

Recording software first included DAW Reaper and Audacity, both of which were then replaced by a custom-written Python script enabling automatization and sequential recording. The recordings were saved in lossless WAV format. In order to standardize the naming of recordings, the following name format was established: GuitarModel_ChordName_Position _SettingsKnob_PlayerID_SampleNumber.wav

Acoustic recordings were conducted in the studio facilities of the Gdańsk University of Technology using a variety of microphone placements to ensure accurate spatial representation and tone capture. After testing 4 possible microphone devices and performing sound checks, we chose Neumann TLM 103 as the one representing the most real sound of the guitars. This microphone is a renowned large-diaphragm cardioid condenser microphone, known for its pristine sound quality, exceptionally low self-noise, and versatility in various recording applications. It covers a broad frequency range from 20 Hz to 20 kHz, ensuring accurate and detailed sound reproduction for acoustic guitars.

2.3 Feature Extraction and ML-based model classification

To complement the G.A.D.A. dataset and establish baseline performance metrics for guitar model classification, we have performed classification tests using various Machine Learning models, which are k-Nearest-Neighbours, Support Vector Machines, Feed-forward neural network and Convolutional Neural Network. Feature extraction for performed using the Mel-Frequency Cepstral Coefficients (MFCCs) feature-extraction. For the Convolutional Neural Network we not only used MCFFs but also chromagrams and mel-spectrograms.

2.3.1 MFCC Feature Extraction

This task, handled by extract-features-and-save.py, processes all .wav files located in the organized folder structure. For each recording, it computes a set of 13 Mel-Frequency Cepstral Coefficients (MFCCs) using a fixed sample rate of 48 kHz. The MFCCs are averaged over time to a single 13-dimensional feature vector per file.

The output includes:

- `features.npy`: NumPy array of shape (nsamples,13)(nsamples,13)
- `labels.npy`: Corresponding array of string-based class labels (e.g., `"Gretsch"`, `"HBLP"`)
- `filenames.npy`: Array containing filenames (including relative paths) associated with each feature vector, enabling later error analysis and tracing of misclassified samples.

2.3.2 KNN

The component of `train-and-classify-knn.py`, performs the classification task. It loads the previously saved features and labels, encodes the labels using `LabelEncoder`, and applies `StandardScaler` to normalize the features. The dataset is split into training and test sets using a stratified sampling strategy to preserve class balance.

A k-Nearest Neighbors (kNN) classifier is then trained and evaluated. Model performance is visualized using a confusion matrix, highlighting correct and incorrect predictions across guitar model classes. This provides an interpretable, entry-level benchmark for instrument classification tasks using shallow learning methods.

2.3.3 Support Vector Machine (SVM)

The script `train-and-classify-svm.py` handles the classification using a Support Vector Machine (SVM) model. It loads the previously extracted MFCC feature vectors, class labels, and associated filenames.

Before training, the class labels are encoded into integers using `LabelEncoder`, and feature vectors are normalized using `StandardScaler` to ensure that all features contribute equally to the model's decision boundary.

The dataset is split into training and test sets, preserving the overall class distribution. An SVM classifier with an RBF (Radial Basis Function) kernel is then trained on the training set. A high regularization parameter (C=100) is used to minimize classification errors, and class balancing is applied to address potential class imbalance issues.

After training, the model's predictions on the test set are evaluated. The achieved accuracy is reported, and a confusion matrix is plotted to visualize the classification performance across different guitar models. Misclassified samples are identified and detailed, including the original filename, true label, and predicted label, allowing a deeper inspection of the classifier's errors.

2.3.4 Feed-forward network

The Feed-forward network (FFN) model consists of an input layer matching the feature size, two hidden layers (128 and 64 neurons) with ReLU activations, and dropout layers (50%) to prevent overfitting. The final layer uses a softmax activation to output probabilities across all guitar model classes.

The network is compiled with the Adam optimizer and trained using sparse categorical cross-entropy loss. After training, its performance is evaluated on the test set by measuring accuracy and visualizing a confusion matrix. Misclassified examples are identified and reported along with their corresponding filenames and labels, enabling further analysis of classification errors.

2.3.5 Convolutional network

The proposed model employs a Convolutional Neural Network (CNN) architecture specifically designed for acoustic feature analysis of guitar recordings. This network has been optimized to process and differentiate between spectral and timbral characteristics unique to different guitar models. The architecture accommodates the aforementioned MFCCs, as well as other two distinct input feature representations: mel-spectrograms and chromagrams, each requiring slight modifications to the network structure.

Mel-spectrogram

For the mel-spectrogram representation, the network consists of three sequential convolutional blocks of increasing filter complexity. The initial block employs 32 filters with 3×3 kernels, followed by an intermediate block with 64 filters, and culminating in a deep block with 128 filters. Each convolutional layer utilizes ReLU activation functions and is followed by max pooling operations for dimensionality reduction and dropout mechanisms to enhance generalization. These hierarchical convolutional layers progressively extract increasingly abstract representations of spectral patterns characteristic of different guitar models.

Chromagram

The chromagram-based architecture follows a similar but simplified design with two convolutional blocks instead of three, accounting for the reduced dimensionality of chromagram inputs. This design choice was made considering that chromagrams already represent abstracted pitch-class information, requiring less hierarchical processing than raw spectral data.

Training Protocol

The network is trained using a robust cross-validation methodology to ensure generalizability across varying datasets. The data is initially partitioned into stratified training (80%) and testing (20%) sets to maintain class balance. A 5-fold cross-validation strategy is implemented on the training set, wherein the data is further divided into five equally sized subsets. During each of the five iterations, four subsets are used for training while the remaining subset serves as validation data.

2.3.6 Random forest

Random Forest was used to identify and eliminate irrelevant features, as well as to determine which MFCC features are the most important for distinguishing between guitar models. Thanks to its built-in feature importance evaluation, Random Forest enabled us to better understand the contribution of each feature to the classification task and to focus the analysis on the most informative components.

2.4 Visualization

To further explore the structure of the MFCC feature space, we applied Principal Component Analysis (PCA) and generated a 3D scatter plot of the data projected onto the top three principal components. The rotating animation (1Single frame of 3D PCA visualization.figure.caption.1) visually demonstrates the natural clustering of different guitar models, suggesting that even simple, low-dimensional descriptors such as MFCCs carry meaningful information for classification.

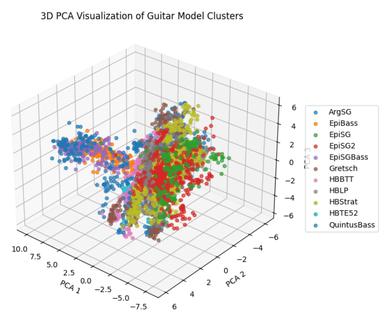

Fig. 1: Single frame of 3D PCA visualization.

3 Results

The aim of the project was to create a unique corpus of recordings from different guitar models and develop an initial system for recognizing the guitar model based on the analysis of audio recordings using machine learning methods. An analysis of the impact of the number of MFCC parameters and different data normalization methods on classification performance was carried out. Three types of classifiers were tested: KNN (k-Nearest Neighbors), SVM (Support Vector Machine), and RF (Random Forest). The experiments were conducted for a range of MFCC coefficients from 5 to 60. The results are presented as classification accuracy for different data normalization methods: StandardScaler, MinMaxScaler, and RobustScaler. Below is a summary of the key observations:

3.1 KNN

1. The classification accuracy increased with the number of MFCC coefficients.
2. The highest accuracy was achieved with around 47–55 MFCC (approximately 99.0%).
3. The differences between the normalization methods were small, with StandardScaler and MinMaxScaler giving slightly better results than RobustScaler.

3.2 SVM

1. The SVM model achieved the highest results starting from 20 MFCC coefficients, exceeding 95% accuracy.
2. The best results appeared with 40–60 MFCC, reaching accuracy up to 99.6%.
3. In this case, the StandardScaler normalization method also gave the best results.

3.3 Random Forest

We have conducted tests of feature importance test using Random Forest. We checked the influence of the numbers of features and also the impact of specific features on the predictions. Then we have performed the Random Forest classification itself.

3.3.1 Number of features

1. **General Accuracy Trend**:
   For both the kNN and SVM classifiers, accuracy increases with the number of MFCC features. This trend is visible up to around 40 features, after which accuracy stabilizes and gains are marginal.
2. **Classifier Comparison:**
   The SVM classifier generally outperforms kNN at lower MFCC dimensions, but both achieve near-perfect accuracy (close to 1.00) when more than 40 features are used.
3. **Scaler Impact:**
   The difference between the three scalers is minimal. All of them perform similarly across the entire MFCC range, with only slight variations in certain regions. This suggests that the model is robust to the choice of scaling method.
4. **Optimal Feature Range:**
   The best performance (99% accuracy or higher) is achieved when the number of MFCC features is between 40 and 60, indicating this is an optimal range for both classifiers in this task.

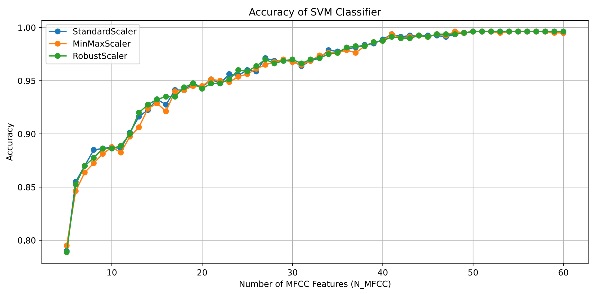

Fig. 2: Accuracy of the SVM classifier dependent on number of MFCC features.

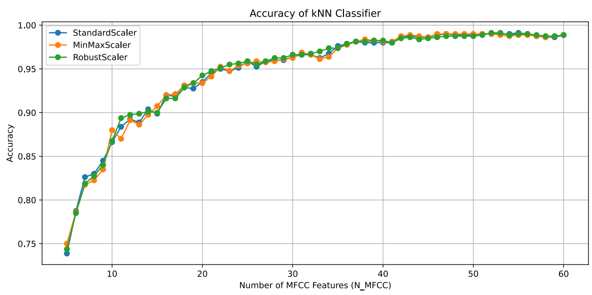

Fig. 3: Accuracy of kNN classifier dependent on number of MFCC features.

3.3.2 Feature importance

SVM

The first few MFCC features are clearly the most important for the SVM classifier. These low-index coefficients (especially features 0 to 5) show the highest permutation importance values. This makes sense because the first MFCCs represent the overall spectral shape of the signal, which is closely related to the timbre or color of the sound — a key factor in identifying instruments or their models.

Feature 41 stands out significantly compared to its neighbors. Despite being a high-order coefficient (which usually captures finer, more noisy spectral details), it shows an unusually high importance. This anomaly suggests that feature 41 might capture some unique characteristic or subtle resonance specific to certain guitar models or recording conditions. It's worth investigating further what exactly this coefficient encodes — possibly by analyzing the corresponding MFCC band or listening to audio samples where this feature differs.

Most higher-order MFCCs (after index 10) contribute much less to classification, showing very low or near-zero importance. This supports the idea that dimensionality reduction or feature selection could improve model efficiency without sacrificing accuracy.

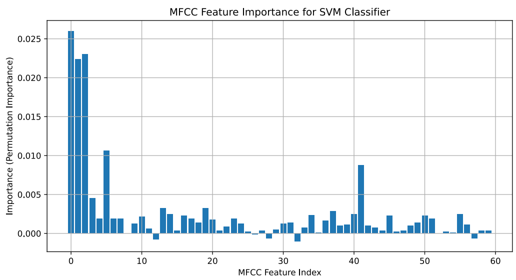

Fig. 4: Accuracy of kNN and SVM classifiers dependent on number of MFCC features.

KNN

In this experiment, a k-Nearest Neighbors (kNN) classifier was used with n_neighbors set to 3 (5MFCC feature importance for KNN Classifier (n_neighbors = 3).figure.caption.5).

The permutation importance plot for MFCC features reveals that many features have negative importance values. This occurs when randomly permuting a given feature actually improves the classifier's performance on the test set, suggesting that the feature may introduce noise or confusion into the decision process.

This phenomenon is particularly relevant for distance-based algorithms like kNN, where irrelevant or misleading features can distort the distance metric, especially

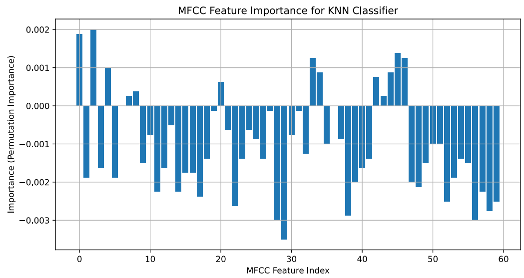

Fig. 5: MFCC feature importance for KNN Classifier (n_neighbors = 3).

in higher-dimensional spaces. In this case, several high-index MFCC features may downgrade the classifier's effectiveness rather than contributing to accurate predictions.

To explore this further, the number of neighbors was increased to n_neighbors = 5 (6MFCC feature importance for KNN Classifier (n_neighbors = 5).figure.caption.6).

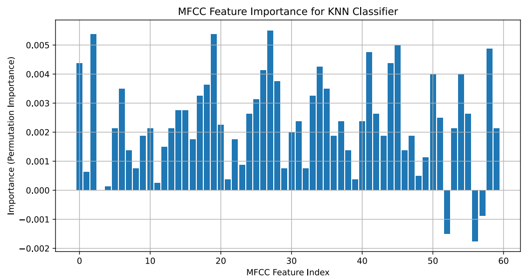

Fig. 6: MFCC feature importance for KNN Classifier (n_neighbors = 5).

With this adjustment, the feature importance distribution became more balanced and almost all features had positive importance values, indicating more stable behavior and reduced sensitivity to noisy or less informative features.

However, despite this change, the overall classification accuracy remained nearly the same, meaning the increase in n_neighbors did not significantly improve the model's performance. This suggests that while adjusting k can improve the interpretability or robustness of the model, it does not necessarily lead to better accuracy in this particular setup.

A possible next step would be to perform feature selection to remove unhelpful MFCC coefficients or to experiment with dimensionality reduction methods such as PCA to simplify the feature space.

3.3.3 Classification with Random Forest Classifier

1. **General Accuracy Trend:** For the RandomForest classifier, accuracy increases sharply with the number of MFCC features, rising from approximately 40% at 1 feature to 90% at around 15 features. Beyond this point, accuracy stabilizes between 90% and 98%, with minor fluctuations.

2. **Scaler Comparison:** The three scalers (StandardScaler, MinMaxScaler, and RobustScaler) perform similarly overall. RobustScaler slightly outperforms at lower feature counts (e.g., 73% at 5 features vs. 70% for StandardScaler). At higher feature counts, MinMaxScaler shows more variability (e.g., a dip to 95% at 50 features), while RobustScaler achieves the highest accuracy of 0.9814 at 60 features.

3. **Stability of Performance:** After 15 features, accuracy for all scalers stabilizes between 90% and 98%, with occasional peaks above 98% (e.g., 98.48% for StandardScaler at 56 features). This suggests that adding more features beyond 15 provides diminishing returns.

4. **Optimal Feature Range:** The best performance (consistently above 98%) is achieved when the number of MFCC features is between 53 and 60, particularly for RobustScaler (98.14% at 60 features) and MinMaxScaler (98.14% at 53 features), indicating this as an optimal range for the RandomForest classifier in this task.

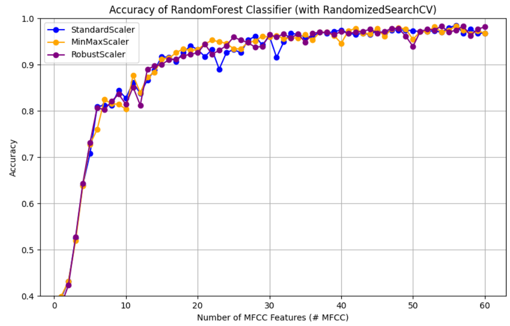

Fig. 7: Accuracy of Random Forest Classifier for different numbers of features.

3.3.4 Feature Importance for Random Forest Classifier

1. **Dominance of Low-Order MFCC Features:** The first few MFCC features (indices 0 to 3) are the most important for the RandomForest classifier, with permutation importance values ranging from 0.01 to 0.05. Feature 1 (index 1) stands out with the highest importance at 0.05. This makes sense as lower-order MFCCs capture the overall spectral envelope and energy distribution of the audio signal, which are key factors in distinguishing between audio classes, such as music genres or instrument types.

2. **Negative Importance of Certain Features:** Several features, such as those at indices 10, 29, 53, and 58, exhibit negative permutation importance values (e.g., around -0.005). This indicates that permuting these features improves the model's performance, suggesting they may introduce noise or irrelevant information. These features could be capturing artifacts or irrelevant spectral details, and their removal might enhance model accuracy. Further analysis—possibly by examining the corresponding MFCC bands or audio samples—could clarify their detrimental impact.

3. **Sparse Contribution of Higher-Order MFCCs:** Beyond index 3, most MFCC features contribute minimally, with permutation importance values near zero or slightly negative. A few features, such as feature 22 (0.01), show small positive contributions. This sparsity in importance suggests that higher-order MFCCs, which capture finer spectral details, are generally less critical for classification in this context. It highlights the potential for dimensionality reduction by focusing on the most impactful features.

4. **Implications for Feature Selection and Model Optimization:** The concentration of importance in the first few features, combined with the presence of negative importance for several higher-order features, indicates that a subset of MFCC

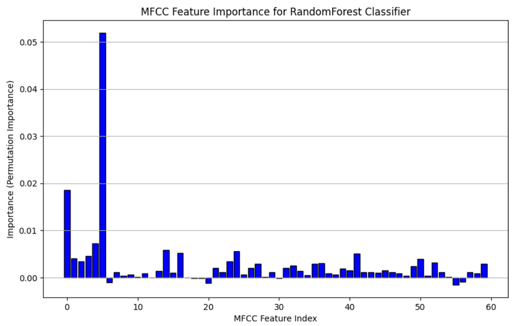

Fig. 8: Accuracy of Random Forest Classifier for different numbers of features.

features could improve model performance. Selecting the top 5–10 most important features (e.g., indices 0–3 and 22) and excluding those with negative importance (e.g., indices 10, 29, 53, 58) may reduce noise, enhance model efficiency, and potentially improve classification accuracy.

3.4 Feed-forward Neural Network

First test using neural networks was performed for 13 MFCC coefficients, reached best results to 87% of correctness. Below we present the confusion matrix showing classification results:

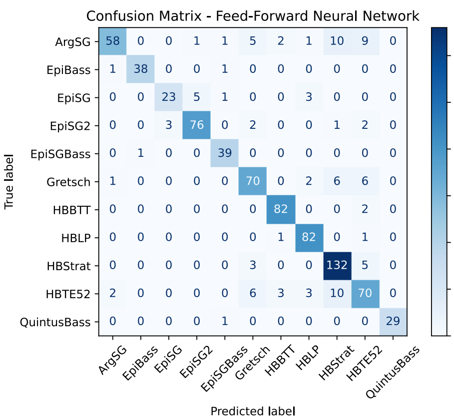

Fig. 9: Confusion matrix for feed forward classification for electric and bass guitars.

The second test, similarly to the previous tests, was conducted to examine how the accuracy changes with varying numbers of MFCC features. The plot (10Accuracy of FNN and CNN classifiers depen- dent on number of MFCC features.figure.caption.10) below presents a comparison of these changes for both the FNN and CNN models.

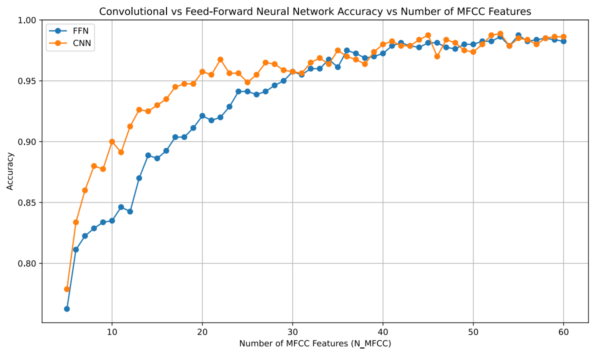

Fig. 10: Accuracy of FNN and CNN classifiers dependent on number of MFCC features.

1. **General Accuracy Trend:**
   For both the FNN and CNN models, accuracy increases steadily as the number of MFCC features grows. The most significant improvement occurs up to about 30 features, after which accuracy continues to rise but more gradually.

2. **Model Comparison:**
   The CNN model achieves slightly higher accuracy than the FNN model for most MFCC dimensions. Particularly between 10 and 35 features, CNN consistently outperforms FNN. For higher numbers of features, the performance of both models becomes very similar.

3. **Stability and Saturation:**
   After approximately 35–40 MFCC features, the accuracy of both models stabilizes above 96%. Minor fluctuations are present, but they are small and do not significantly affect the overall trend.

4. **Optimal Feature Range:**
   The best performance (accuracy above 97–98%) is achieved when the number of MFCC features is between **35 and 60**, making this the recommended range for both the CNN and FNN models in this task.

3.5 Convolutional Neural Network

Below we present confusion matrices for chromagrams and mel-spectrograms. (11Confusion matrix for Convolutional Neural Network using chromagram representation.figure.caption.11, 12Confusion matrix for Convolutional Neural Network using mel-spectrogram representation.figure.caption.12).

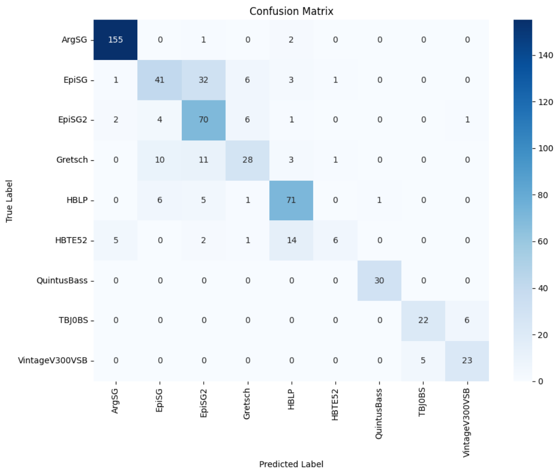

Fig. 11: Confusion matrix for Convolutional Neural Network using chromagram representation.

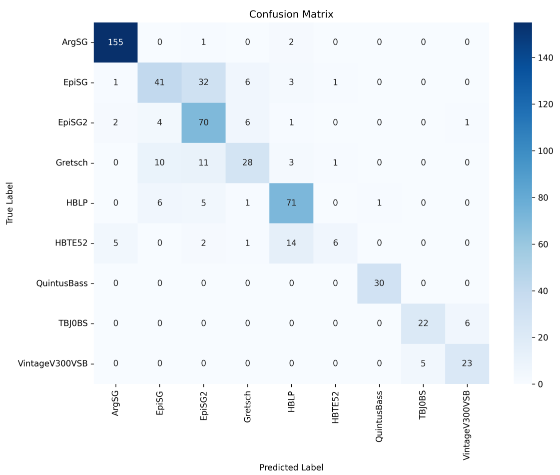

Fig. 12: Confusion matrix for Convolutional Neural Network using mel-spectrogram representation.

We also present training results for both representations. (13Training results for chromagram representation.figure.caption.13, 14Training results for mel-spectrogram representation.figure.caption.14).

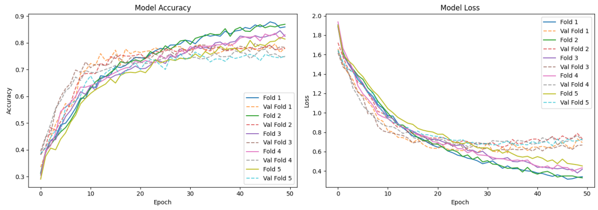

Fig. 13: Training results for chromagram representation.

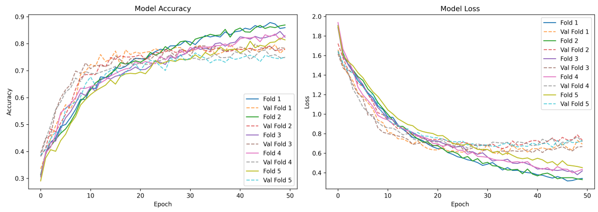

Fig. 14: Training results for mel-spectrogram representation.

Model performance is rigorously evaluated using a comprehensive set of metrics. Beyond simple accuracy, we calculate precision (the ratio of true positives to predicted positives), recall (the ratio of true positives to actual positives), and F1-score (the harmonic mean of precision and recall) for each class. These metrics provide a more nuanced view of classification performance, especially in the presence of class imbalance.

Visual analysis complements numerical metrics through confusion matrices that highlight misclassification patterns between guitar models. Learning curves depicting training and validation metrics across epochs offer insights into the dynamics of the learning process and potential overfitting. The per-class performance metrics visualization provides a detailed breakdown of how well the model performs across the different guitar models in the dataset.

This CNN-based approach demonstrates effective capability for discriminating between different guitar models based solely on their acoustic characteristics, with particular efficacy when using mel-spectrogram representations due to their richer spectral content and alignment with human auditory perception.

3.6 Final accuracy achieved

Here we present the final accuracy of the classifications:

Classifier	Accuracy
k-Nearest Neighbors	99%
Support Vector Machine	99%
Feed-Forward NN	98%
Convolutional NN - spectrogram	87%
Convolutional NN - mel-chromagram	87%
Convolutional NN - MFCC	98%
Random Forest	98%

Table 1: Classification accuracy for different models.

4 Discussion

A comprehensive evaluation of multiple machine learning approaches for guitar model classification using the G.A.D.A. dataset reveals several important observations. The results show that guitar models within the same family can be effectively distinguished based on their audio features with high accuracy, answering the fundamental question posed in our introduction.

Performance Comparison Across Classifiers

Traditional machine learning algorithms, in particular k-NN and SVM, achieved classification accuracy of the order of 99% when equipped with an appropriate number of MFCC features. The Random Forest classifier also performed with high accuracy (on the order of 98%). The high accuracy rates suggest that the sonic differences between guitar models are significant enough to be effectively captured using spectral features, even with relatively simple classification algorithms.

Neural network-based approaches did not outperform traditional methods in this particular case. Feed-Forward neural networks and CNN variants using chromagrams and mel-spectrograms achieved noticeably lower accuracy (87%), while CNNs with MFCC features performed comparable to Random Forest (98%). This suggests that for this particular classification problem, the increased complexity of deep learning architectures may not provide significant benefits over well-tuned conventional classifiers.

Feature Analysis and Optimization

Feature importance analysis revealed key observations for many classifiers. Low-order MFCC coefficients (indices 0-5, specifically) consistently showed the highest importance, confirming their importance in capturing the fundamental sonic features that differentiate guitar models. This is consistent with acoustic theory, as these coefficients represent the overall spectral envelope most relevant to instrumental timbre.

It was observed that some higher-order MFCCs contributed negligibly or even negatively to classification accuracy, suggesting the potential for dimensionality reduction without compromising performance. This knowledge has practical implications for developing more efficient classification systems that can operate with reduced computational requirements while maintaining high accuracy.

Experiments with varying numbers of MFCC features consistently showed performance improvements up to about 35–40 features, after which accuracy plateaued with only marginal gains. This pattern was observed across all classifier types, indicating an inherent limitation in the discriminative information available in spectral features for this particular task.

Implications for Audio Analysis and Instrument Recognition

The exceptional classification accuracy achieved with multiple algorithms suggests that the acoustic signatures of different guitar models are indeed objectively distinguishable, providing an empirical basis for discussing tonal differences between instruments. This finding has potential applications beyond academic research, including quality control in instrument manufacturing, automated inventory management for musical instrument retailers, and plug-ins for digital audio workstations for guitar tone identification or emulation.

The small differences in performance between different feature representations (MFCCs, chromagrams, and mel-spectrograms) underscore the importance of selecting the right acoustic features for specific audio classification tasks. The superior performance of MFCC-based approaches in this context highlights their continued utility for timbre-related classification problems, despite the growing popularity of deep learning on raw spectrograms.

Limitations and Future Work

Although our results demonstrate high classification accuracy, it is important to acknowledge several limitations. The current dataset consists of a specific set of guitar models played in controlled recording environments. Future work should test the robustness of these classification approaches under more diverse playing techniques, different chord voicings, variable dynamics, and different recording conditions.

Furthermore, expanding the dataset to include more guitar models from different manufacturers would further confirm the generalizability of our findings. Including recordings made with amplifiers and effects processing would also better reflect real-world scenarios where guitar sounds are rarely heard in their direct, unprocessed form.

Future research directions may include investigating the applicability of transfer learning approaches, exploring unsupervised learning techniques to discover natural guitar sound clusters, and developing explainable AI methods to better understand which specific acoustic properties contribute most to the characteristic sound of different instruments.

Possible directions of system development

- **Dataset extension** - the current database of recordings may be limited in terms of the variety of artists, recording conditions, or playing techniques. In the future, an "open" corpus of recordings could be created, where each user could add their own sound sample. Additionally, this would allow the corpus to be expanded to include other guitars. Furthermore, the corpus could be expanded to include recordings of more chords in different configurations as well as feature other settings of tone and volume knobs (not only the maximum).
- **Online system or mobile app** - the created system can be adapted to the form of an interactive application – for example, to identify a guitar model "live" based on a microphone recording. Such a system could find application in music education, instrument production, or among collectors and manufacturers. It would also be possible to provide functionality for adding sound samples, thus implementing the first development proposal.
- **Classification of other instrument features** - the system could be extended beyond just recognizing the guitar model, but also the type of soundboard, type of wood, strings, and even the guitarist's playing technique. However, such detailed classifications would require precisely described input data and advanced analysis.
- **Exploring more ML and AI solutions** - future work could involve investigating more advanced machine learning or artificial intelligence methods. Additionally, it would be beneficial to implement approaches that could help better understand the specific audio features driving model decisions.

5 Summary

This paper introduced G.A.D.A. (Guitar Audio Dataset for AI), a novel open-source dataset designed to support research in guitar audio analysis, signal processing, and machine learning application. The dataset encompasses recordings from electric, acoustic, and bass guitars, with multiple instruments represented within each category to ensure diversity and robustness. Evaluated the utility of G.A.D.A. using a comprehensive testing framework incorporating established machine learning algorithms, including k-Nearest Neighbors, Support Vector Machines, Random Forest, Feed-Forward Neural Networks, and Convolutional Neural Networks. The highest classification accuracy (99%) was achieved by k-NN and SVM classifiers when provided with an optimized number of MFCC features, demonstrating that guitar models can be effectively distinguished based on their audio characteristics. Feature importance analysis revealed that low-order MFCC coefficients carry the most discriminative information for this task, providing insights for future feature optimization. The observations confirm that the timbral differences between guitar models are objectively distinguishable through computational analysis. The G.A.D.A. dataset and our baseline classification results establish a foundation for future research in musical instrument classification, audio signal processing, and computer-aided music applications.

References

[1] Hegel Pedroza, R. M. C. I. R. R., Wallaca Abreu, "Guitar-TECHS: An Electric Guitar Dataset Covering Techniques, Musical Excerpts, Chords and Scales Using a Diverse Array of Hardware," *arXiv*, pp. 1–5, 2025.

[2] Q. Xi, J. P. X. Y. J. B., Rachel M. Bittner, "GuitarSet: A Dataset for Guitar Transcription," *International Society for Music Information Retrieval Conference*, pp. 1–8, 2018.

[3] Abeßer, D.-I. J., "IDMT-SMT-Guitar," *Fraunhofer IDMT*, p. 1, 2023.