Roma Tre

*Article*

# Explainable Instrument Classification: From MFCC Mean-Vector Models to CNNs on MFCC and Mel-Spectrograms with t-SNE and Grad-CAM Insights

**Tommaso Senatori [1] and Daniela Nardone [2]**

[1] Department of Civil, Computer Science and Aeronautical Technologies Engineering, Università degli Studi Roma Tre, Via V. Volterra 62, Roma, 00146, Italy
[2] Department of Civil, Computer Science and Aeronautical Technologies Engineering, Università degli Studi Roma Tre, Via V. Volterra 62, Roma, 00146, Italy

**Abstract:** This paper presents an automatic system for the classification of musical instruments from audio recordings. The project leverages Deep Learning (DL) techniques to achieve its objective, exploring three different classification approaches based on distinct input representations. The first method involves the extraction of Mel-Frequency Cepstral Coefficients (MFCCs) from the audio files, which are then fed into a two-dimensional convolutional neural network. The second approach makes use of Mel-Spectrogram images as input to a similar Conv2D architecture. The third method relies on traditional Machine Learning (ML) classifiers—namely Logistic Regression, K-Nearest Neighbors, and Random Forest—trained on MFCC feature vectors. To gain insight into the behavior of the DL model, explainability techniques were applied to the Conv2D model using Mel-Spectrograms, allowing for a better understanding of how the network interprets relevant features for classification. Additionally, t-distributed stochastic neighbor embedding (t-SNE) was employed on the MFCC vectors to visualize how instrument classes are organized in the feature space. One of the main challenges encountered was the class imbalance within the dataset, which was addressed by assigning class-specific weights during training. The results, in terms of classification accuracy, were very satisfactory across all approaches, with the convolutional models and Random Forest achieving around 97–98%, and Logistic Regression yielding slightly lower performance. In conclusion, the proposed methods proved effective for the selected dataset, and future work may focus on further improving class balance techniques.

## 1. Introduction

Automatic musical instrument classification is a central topic in Music Information Retrieval (MIR), with a wide range of applications in artificial intelligence, music education, and audio production. Recognizing the instrument from an audio signal is a challenging task, as the signal can vary significantly depending on factors such as performance style, recording environment, background noise, and overlapping sources.

Alongside traditional techniques, recent approaches leverage deep neural networks (particularly convolutional architectures) applied to time-frequency representations such as spectrograms or mel-spectrograms, which are more effective at capturing the distinctive characteristics of audio signals.

The structure of this paper is as follows:

- Section 2 reviews related works on musical instrument classification and the main techniques employed in the literature;

- Section 3 presents the materials and methods, including dataset preparation, feature extraction, and the architecture of the neural models used;
- Section 4 reports the experimental results and evaluation metrics, with particular focus on the explainability techniques adopted to interpret model behavior;
- Section 5, finally, provides conclusions and discusses possible future developments.

## 2. Related Works

One of the earliest contributions in the field of musical instrument recognition from audio files is the work by Haidar-Ahmad (2019) [1], which describes the implementation of a multi-class classifier based on a convolutional neural network (CNN) for identifying dominant instruments in audio tracks. The model takes Mel-spectrograms extracted from audio signals as input and classifies the dominant instrument into one of four categories: "Piano," "Drums," "Flute," or "Other." The reported results include a precision of 70%, recall of 65%, and an F1-score of 64%. The author suggests future work involving the implementation and comparison of more advanced deep learning architectures such as RNN, RCNN, and CRNN, as well as expanding the range of instrument classes.

The study by Solanki and Pandey (2022) [2] propose a model based on a deep convolutional neural network consisting of eight layers aimed at recognizing predominant instruments in real polyphonic music recordings. The model uses Mel-spectrograms as input and employs ReLU activation functions, Max Pooling for dimensionality reduction, and Dropout techniques to mitigate overfitting. Classification is performed using a Softmax function, and the model achieved an accuracy of 92.8% in identifying primary instruments.

More recently, Blaszke and Kostek (2022) [3] introduce an innovative approach that uses a separate convolutional neural network for each musical instrument, allowing for flexible system extension with new instruments. Their results demonstrate high precision in recognition, ranging from 0.86 for guitar to 0.99 for drums.

Finally, Chen et al. (2024) [4] present an interpretative analysis of CNN models applied to musical instrument recognition, based on heatmaps derived from various spectral representations. The authors classify ten instruments using CNNs trained on STFT, Log-Mel, MFCC, Chroma, Spectral Contrast, and Tonnetz spectrograms extracted from the NSynth database. To assess feature importance and model interpretability, visual analyses through heatmaps and statistical metrics, including Kullback-Leibler divergence, Jensen-Shannon divergence, Earth Mover's distance, and mean difference, were employed. The results highlight that MFCC and Log-Mel spectrograms generally yield superior performance, while other representations provide complementary information useful for model interpretation.

## 3. Materials and Methods

### *3.1. Dataset and Libraries*

#### 3.1.1. Dataset

The dataset used for this project, named Music Instrument Sounds for Classification, includes 28 musical instrument classes: Accordion, Acoustic Guitar, Banjo, Bass Guitar, Clarinet, Cowbell, Cymbals, Dobro, Drum Set, Electric Guitar, Floor Tom, Flute, Harmonica, Harmonium, Hi-Hats, Horn, Keyboard, Mandolin, Organ, Piano, Saxophone, Shakers, Tambourine, Trombone, Trumpet, Ukulele, Vibraphone, and Violin.

To ensure consistency and data quality, each audio file in the dataset has been carefully curated and preprocessed: all audio clips have a fixed duration of 3 seconds, were selected to clearly and distinctly represent the corresponding instrument, and any recordings containing large portions of silence were excluded. The audio files are provided in the .wav format, which is widely compatible with major audio processing libraries.

The main challenge posed by this dataset lies in its class imbalance: while some instrument

classes are represented by thousands of samples, others contain only a few hundred. This imbalance required the use of specific techniques during the model training phase to mitigate potential bias.

#### 3.1.2. Libraries

For the processing of audio data, extensive use was made of Python's scientific libraries. Among them, Librosa played a central role, proving to be an essential tool for audio analysis and feature extraction.
Librosa is an open-source Python library specifically designed for audio signal analysis and processing, with a strong focus on Machine Learning and Deep Learning applications. It supports various audio formats such as WAV, MP3, and FLAC, and provides a comprehensive set of tools for extracting audio features, including MFCCs, spectrograms, chromagrams, tempo, pitch, and more.
Beyond feature extraction, Librosa offers advanced functionality for audio visualization and manipulation, such as Fourier transforms, spectrogram generation, beat tracking, harmonic-percussive source separation, pitch shifting, time stretching, and structural analysis of audio tracks.
Thanks to its intuitive API and seamless integration with visualization libraries like Matplotlib, Librosa stands out as a highly versatile and indispensable tool in music classification, speech recognition, and audio signal processing projects.

In this project, extensive use was also made of Keras for implementing and training the neural network models. Keras is an open-source Python library for Deep Learning, designed to streamline the development, training, and evaluation of neural networks: it allows the integration of custom layers, activation functions, loss functions, and optimizers, making it adaptable to a wide range of projects. Keras is compatible with several computational backends, including TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK) and can run on CPUs, GPUs, and TPUs.

### *3.2. Data Analysis*

In the initial phase, an exploratory analysis of the dataset was carried out to understand its structure and to examine the characteristics of the instances with respect to the different musical instrument classes.
The distribution of audio files per class was then analyzed: as shown in Figure 1, the dataset is significantly unbalanced, with the number of samples varying considerably across classes, as previously discussed. This class imbalance issue will be addressed during the training phase of the Deep Learning model, using specific techniques aimed at mitigating the impact of unequal class representation.

Subsequently, the analysis was deepened through the visualization of waveforms. Waveforms provide a graphical representation of an audio signal in the time domain, enabling the analysis of amplitude variations over time. The horizontal axis represents time progression, while the vertical axis indicates the amplitude of the signal, which corresponds to sound intensity at a given moment. A higher amplitude reflects a louder sound, whereas a lower amplitude indicates a softer one. This visual representation serves as a fundamental tool for audio analysis: examining a waveform allows for the identification of pauses and silent segments, assessment of intensity fluctuations, and recognition of distinctive patterns.

### *3.3. Data Pre-Processing*

The data preprocessing procedure was divided into two distinct phases: feature extraction and standardization.

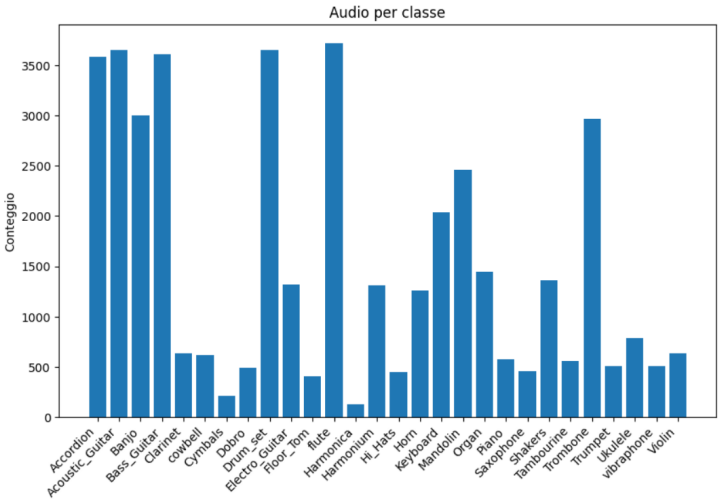



**Figure 1.** Distribution of audio files per class

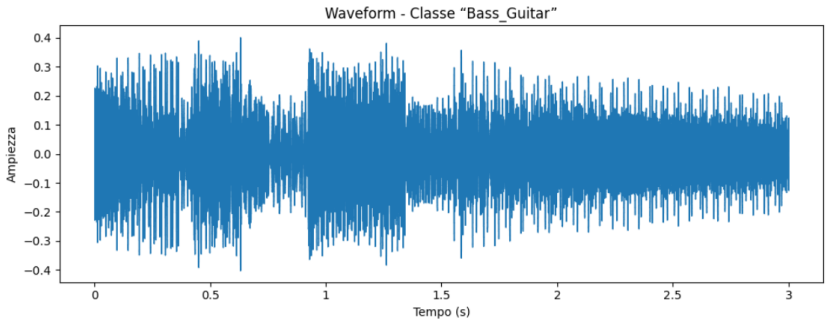



**Figure 2.** Example of the waveform for the class Bass Guitar

### 3.3.1. Features Extraction

In the first phase, audio data were initially loaded using the .load module from the Librosa library, followed by the extraction of relevant features. During this process, it was observed that performing loading and feature extraction simultaneously resulted in significant computational delays. Consequently, the two steps were separated: all audio files were pre-loaded using Librosa, and feature extraction was then executed in parallel, leveraging all available CPU cores (22 cores in this case). This modification led to a substantial improvement in processing time, reducing the combined loading and extraction time from approximately 10 minutes to roughly 4 minutes for loading and 1 minute for extraction.

Specifically, Mel-Frequency Cepstral Coefficients (MFCCs) and Mel spectrograms were extracted and used to train two separate models.

MFCCs capture the spectral energy of audio signals mapped onto a scale that mimics human auditory perception, effectively encoding timbral and tonal variations crucial for musical instrument classification.

The Mel spectrogram visually represents the audio signal’s energy distribution over time and frequency, with frequencies mapped on the Mel scale, a perceptual scale that emphasizes lower frequencies and compresses higher frequencies. Similar to traditional spectrograms, the horizontal axis corresponds to time, while the vertical axis indicates frequencies ordered by the Mel scale. The colour intensity at each time-frequency bin corresponds to the signal amplitude, forming a heatmap. This representation is particularly effective in capturing timbral and harmonic sound features, making it widely used in applications such as speech recognition and music classification. By closely aligning with

human auditory perception, the Mel spectrogram highlights the most relevant acoustic nuances for analysis.

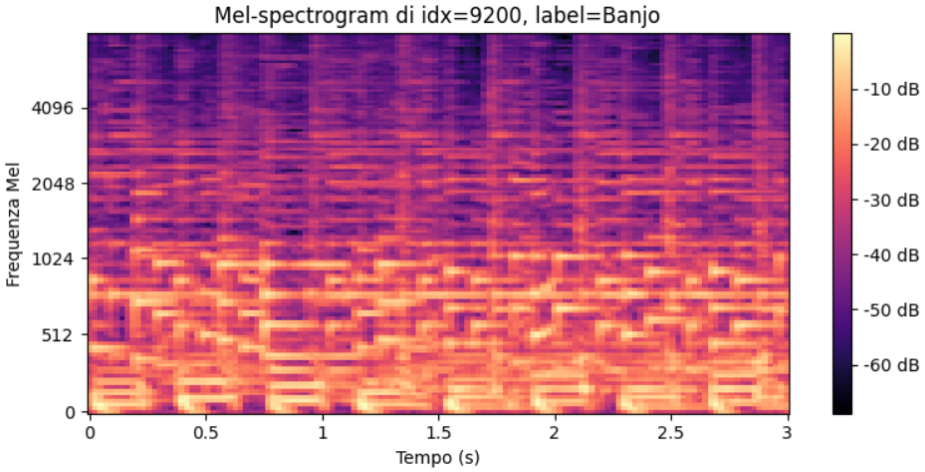



**Figure 3.** Example of the Mel-Spectogram for the class Banjo

#### 3.3.2. Standardization

The second phase involved standardizing the extracted features to bring their numerical values onto comparable scales, thereby enhancing the stability and efficiency of model training. StandardScaler from common libraries could not be used, as it operates on 2D arrays while the data were structured as 3D arrays. Therefore, a custom standardization function was implemented to normalize only the last two dimensions of the array by computing their mean and standard deviation.

### *3.4. CNN-based Models*

Before model construction, the preprocessed data were randomly shuffled and split into training, validation, and test sets with an 80-10-10 ratio. The label vector, initially categorical, was transformed into binary numerical vectors through One-Hot Encoding. Additionally, a new dimension was added to the data arrays to match the input shape required by the neural network.

#### 3.4.1. Model construction

In this study, a two-dimensional Convolutional Neural Network (2D CNN) was implemented for multi-class classification. The architecture follows a typical hierarchical CNN structure and incorporates Batch Normalization to enhance numerical stability and accelerate training convergence. The model consists of three convolutional blocks, each comprising a Conv2D layer with kernel size 3×3, ReLU activation, and "same" padding, followed by a BatchNormalization layer and a MaxPooling2D layer with pool size 2×2. These blocks use 32, 64, and 128 filters respectively, enabling the extraction of progressively more abstract and complex features.

Following the convolutional layers, a GlobalAveragePooling2D layer was added to reduce the feature maps into a one-dimensional vector. This step reduces the number of trainable parameters and helps mitigate overfitting.

A fully connected Dense layer with 128 ReLU-activated units was then applied, followed by a Dropout layer with a dropout rate of 0.5 to promote model generalization by introducing noise during training.

The output layer is a Dense layer with softmax activation and a number of units equal to the number of target classes, producing a probability distribution across the label space.

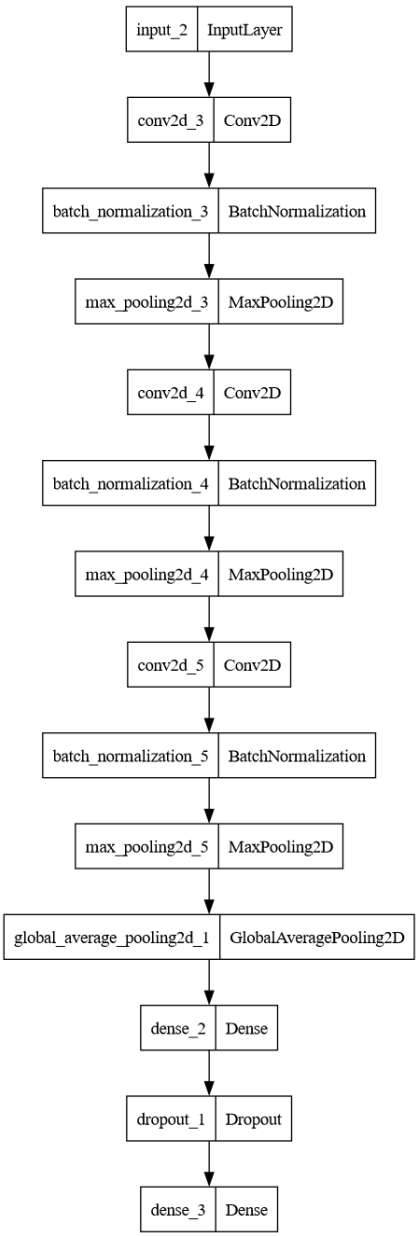



**Figure 4.** Model used for musical instrument classification

#### 3.4.2. Training

The model was compiled using the Adam optimizer with an initial learning rate of 0.001, and categorical cross-entropy was employed as the loss function, which is suitable for multi-class classification tasks with one-hot encoded labels. Accuracy was used as the evaluation metric during training.

To prevent overfitting, an Early Stopping mechanism was employed, monitoring the validation loss and halting training if no improvement was observed for 5 consecutive epochs. The model, then, automatically restores the weights corresponding to the best epoch.

#### 3.4.3. Class Imbalance

To address the class imbalance issue, class weights were computed based on the frequency of each label. The string labels were first encoded into integers using LabelEncoder, and the resulting weights were converted into a dictionary and passed to the model via the class_weight parameter during training. This strategy assigns greater weight to

underrepresented classes in the loss function, encouraging more balanced classification outcomes.

### *3.5. Machine Learning Models*

To compare the performance of the neural network with more traditional approaches, three Machine Learning models were trained and evaluated: Logistic Regression, K-Nearest Neighbors (KNN), and Random Forest.
Logistic Regression was configured with a maximum of 1000 iterations, and the parameter class_weight='balanced' was used to address class imbalance.
The KNN model employed 5 neighbours, with weights assigned based on distance in order to give more importance to closer neighbours.
The Random Forest classifier was implemented using 200 decision trees, and training was performed in parallel across all available CPU cores.
All models were trained using MFCC features extracted from audio signals. From these features were calculated the mean along the time axis, resulting in a fixed-length feature vector for each sample, with dimensionality equal to the number of MFCC coefficients.
Model performance was evaluated using accuracy as the primary metric.

## 4. Results and Evaluation

### *4.1. CNN-based Models*

The validation accuracy achieved by both convolutional neural networks (CNNs) reached approximately 98%.

The training progress was analyzed using accuracy and loss curves, plotted from the history object that stores the training checkpoints for each epoch.
The loss graphs show a smooth and consistent decrease, stabilizing below 0.1 around the 12th epoch, with no evidence of overfitting.
Similarly, the accuracy curves rise steadily up to 97–98%, with minimal fluctuations and a close match between training and validation trends.
These observations indicate a stable learning process and strong generalization capability, with high performance during both training and validation.

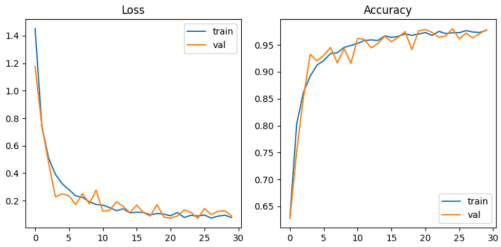



**Figure 5.** Loss and Accuracy curves for the Mel-Spectogram-based CNN model

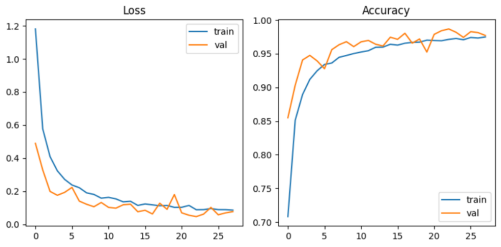



**Figure 6.** Loss and Accuracy curves for the MFCCs-based CNN model

To further support these findings, a classification report was generated, confirming the model's strong performance: just as the overall accuracy, the average F1-score was very high.
However, the model showed slight difficulty in correctly classifying instruments from underrepresented classes. For instance, the Cymbals class, which had only 20 test samples, achieved good recall and F1-score values but a precision of approximately 58%, meaning that barely more than half of the model's predictions for that class were correct. This highlights the persistent impact of class imbalance, which affects performance especially in categories with fewer training examples, despite the model's overall effectiveness.

The confusion matrices confirm the excellent classification results: most instances are correctly classified, aligning along the diagonal, which indicates robust learning and strong recognition of instrument-specific acoustic patterns.
In particular, instruments with similar timbral characteristics, such as bass guitar, accordion, harmonium, and banjo, were classified with near-perfect accuracy, suggesting the model's ability to capture discriminative features even among acoustically related classes.
Finally, greater variability was observed in the results for underrepresented classes, which remain more sensitive to the effects of data imbalance.

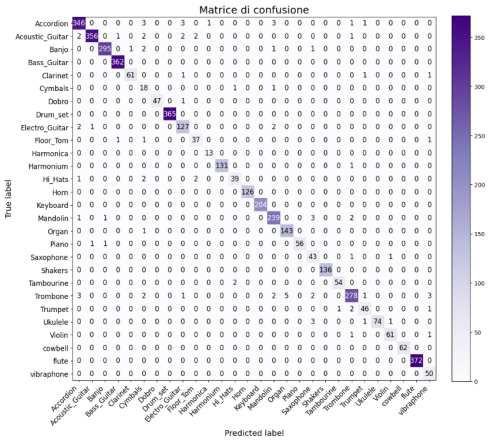



**Figure 7.** Confusion Matrix for the Mel-Spectogram-based CNN model

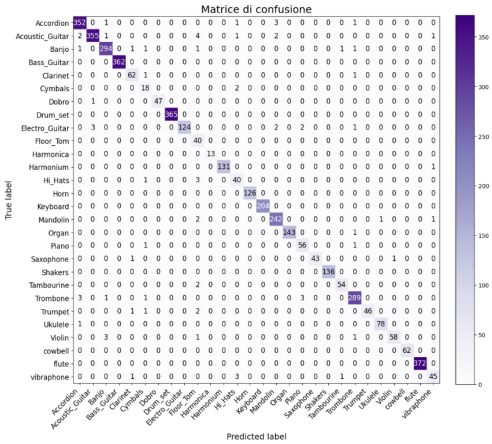
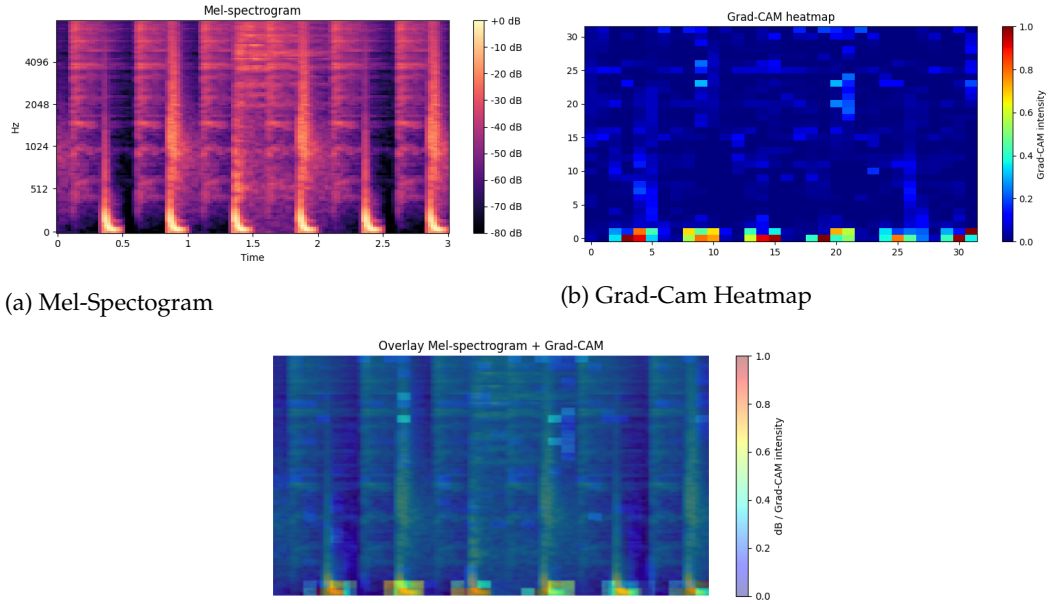



**Figure 8.** Confusion Matrix for the MFCCs-based CNN model

### *4.2. Explainability*

To enhance the interpretability of the audio classification model based on convolutional neural networks, the Grad-CAM (Gradient-weighted Class Activation Mapping) technique was applied to Mel-spectrograms. Grad-CAM enables the visualization of the input regions that the model focuses on to make a prediction by generating a heatmap highlighting the most relevant areas.
Specifically, it computes the gradient of the predicted class probability with respect to the activation maps of the model's last convolutional layer. A spatial average of these gradients is then calculated to obtain an importance weight for each feature map, which is linearly combined with the corresponding activation maps. The resulting localized heatmap, when overlaid on the original Mel-spectrogram, highlights the frequency bands and temporal segments that the network relied upon for classification.
The implementation involved configuring the model to output both the activations of the last convolutional layer and the classification prediction. Using the loss function corresponding to the predicted class, the required gradients were computed. The generated heatmap was then normalized and resized for overlay on the input Mel-spectrogram, enabling clear visualization of the discriminative features detected by the model.
This approach provided a visual explanation of the model's behavior, confirming that the highlighted regions correspond to meaningful audio patterns consistent with the performed classification.




(a) Mel-Spectogram

(b) Grad-Cam Heatmap

(c) Overlay of Grad-Cam Heatmap and Mel-Spectogram

**Figure 9.** Example of the steps for the explainability method Grad-Cam for the class Drum Set

### *4.3. Machine Learning Models*

Logistic Regression achieved an overall accuracy of 87.1%, with a macro-averaged F1-score of 0.82 and a weighted average of 0.87. Despite reasonable performance on several classes, the model exhibited notable difficulties in discriminating between instruments with similar timbral characteristics. The confusion matrix reveals frequent misclassifications among instruments belonging to the same family, indicating a limited ability to capture complex, non-linearly separable patterns.

The KNN model significantly improved classification performance, reaching an accuracy of 96.9%. Previously problematic classes now show satisfactory results. The confusion matrix reveals a clear reduction in inter-class errors, highlighting the model's improved robustness in identifying the distinctive acoustic features of each instrument.

The Random Forest classifier also delivered high performance, achieving an accuracy of 97%. Results across individual classes show high precision for commonly recognizable instruments, along with improvements on more challenging ones. While a slight drop in performance is observed for some classes, the model demonstrates better generalization capabilities compared to Logistic Regression, particularly for underrepresented classes.

Overall, these findings underscore the limitations of Logistic Regression in addressing classification tasks involving complex audio features. In contrast, KNN and Random Forest models prove to be considerably more effective, achieving accuracy levels above 96%. KNN excels in discriminating instruments with overlapping characteristics, while Random Forest offers greater stability and generalization.

### *4.4. t-SNE*

To gain a qualitative understanding of class distribution, the t-distributed Stochastic Neighbor Embedding (t-SNE) technique was applied to the entire dataset, combining training, validation, and test sets.

Each sample is represented by a mean MFCC vector of 40 coefficients, and t-SNE projects these 40-dimensional points into a two-dimensional space. This visualization provides insights into class separability and aids in interpreting the classification outcomes.

The resulting 2D map represents each audio sample as a point, colored according to its

corresponding instrument class. A custom colormap of 28 distinct colors was employed to ensure clear visual distinction between all classes.

The plot reveals well-separated clusters for some classes, suggesting high separability in the original feature space, while others exhibit partial overlap, reflecting timbral similarities among certain instruments.

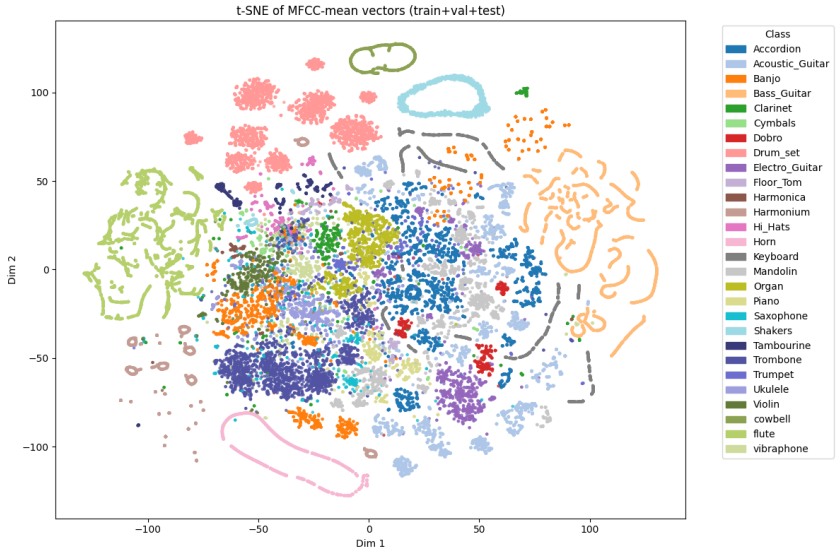



**Figure 10.** t-SNE of the entire dataset

## 5. Conclusions and Future Work

The work conducted enabled the classification of musical instruments from audio files in .wav format. The two 2D Convolutional Neural Network (2D-CNN) models, one trained on MFCC coefficients and the other on Mel-Spectrograms, demonstrated highly satisfactory performance, highlighting the model's ability to effectively recognize meaningful acoustic patterns for instrument classification.

Traditional Machine Learning models also performed well on this task, further confirming the relevance and quality of the extracted features. Based on the analysis of the results, it can be inferred that the dataset used is relatively simple and well-structured, allowing even simpler models to achieve good performance.

A potential enhancement of this work would be the adoption of a more complex and diverse dataset, possibly including greater timbral variability and more realistic recording conditions. An especially promising direction could involve the use of video datasets to enable multimodal classification, combining visual recognition of the instrument with its acoustic analysis.

Another possible extension would be to test the model's efficiency in a live setting, allowing users to record audio in real time and receive immediate classification results.

Ultimately, the project could be elevated by integrating the entire system into a mobile application capable of performing real-time classification directly on a portable device, thus making the tool practical and accessible for everyday use.

## References

1. Haidar-Ahmad, L. *Music and Instrument Classification using Deep Learning Techniques*. 2019. Available online: https://cs230.stanford.edu/projects_fall_2019/reports/26225883.pdf.
2. Solanki, A.; Pandey, S. Music instrument recognition using deep convolutional neural networks. *Int. J. Inf. Technol.* **2022**, *14*, 1659–1668. Available online: https://link.springer.com/article/10.1007/s41870-019-00285-y.

3. Blaszke, M.; Kostek, B. Musical Instrument Identification Using Deep Learning Approach. *Sensors* **2022**, *22*, 3033. Available online: https://www.mdpi.com/1424-8220/22/8/3033.
4. Chen, R.; Ghobakhlou, A.; Narayanan, A. Interpreting CNN models for musical instrument recognition using multi-spectrogram heatmap analysis: a preliminary study. *Front. Artif. Intell.* **2024**, *7*, 1499913. Available online: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1499913/full.