Towards Explainable Music Emotion Recognition for Guitar Improvisations

Michele Rossi*♮, Domenico Stefani*♮, Johan Pauwels†, Giovanni Iacca*, and Luca Turchet*
*Department of Information Engineering and Computer Science
University of Trento, Trento, Italy
Email: michele.rossi-2@unitn.it, domenico.stefani@unitn.it, giovanni.iacca@unitn.it, luca.turchet@unitn.it
†Queen Mary University of London, UK
Email: j.pauwels@qmul.ac.uk

***Abstract*—Explainability has gained significant attention across various domains, yet it remains relatively underexplored in the field of music, particularly in Music Emotion Recognition. This paper presents XMERApp; a web application designed to provide interpretability for a deep learning model that classifies classical and acoustic guitar into four emotional states. Our system employs a deep learning architecture trained on improvised musical performances to classify emotions, while providing comprehensive explainability through multiple complementary approaches. The application offers users three levels of interpretability: (1) detailed breakdowns of prediction probabilities across different emotion categories, enabling users to understand the confidence and uncertainty in model predictions; (2) temporal visualization of emotion evolution throughout the improvisation, revealing how the model's understanding of emotional content develops over time; and (3) LIME-based explanations that highlight specific spectrogram regions most influential to the model's decisions within focused time windows. Additionally, users can listen to the specific spectrogram regions identified as critical for the emotion classification, gaining insights into which parts of the performance and frequency ranges contributed the most to the model's output. The web-based nature of XMERApp enables deployment across many devices, including smart musical instruments, enhancing the interpretability of intelligent features embedded within them.**

***Index Terms*—Explainable Artificial Intelligence, Music Information Retrieval, Emotion Recognition**

I. Introduction

Over the past two decades, deep learning (DL) has profoundly advanced the field of Music Information Retrieval (MIR), but despite their strong performance, DL models tend to be complex and opaque, making it difficult to interpret their internal decision-making processes. Transparent models offer better interpretability but often lack the capacity to deliver high accuracy in complex tasks [1]. This trade-off has motivated the development of model-agnostic explainability techniques that seek to provide insights into black-box DL models without sacrificing performance [2].

While interpretability may be less critical in MIR than in domains like autonomous driving and medicine, it still offers valuable benefits. Model designers can gain a deeper understanding of model functioning, identify key input features, and develop more efficient models with fewer parameters and faster inference times. End users, including musicians, can benefit even more from explanations that clarify the reasons behind model outputs and enable interactive feedback.

One of the approaches to deal with explainability in the music domain is to borrow techniques from other domains (e.g., computer vision) and try to adapt them to MIR tasks. This was the case for Mishra et al. [3], who extended Local Interpretable Model-agnostic Explanations (LIME) [4] for singing voice detection across temporal, frequency, and time-frequency domains, demonstrating that accuracy does not guarantee trustworthiness. Then, Haunschmid et al. [5] improved upon LIME by generating listenable explanations through source separation perturbations. Other techniques include auralization-based explanation methods [6], to convert convolutional features into audio signals, and using a two-step approach for Music Emotion Recognition (MER) combining convolutional feature extraction with an interpretable linear model [7], other than the layer-wise relevance propagation [8]. However, existing research has largely overlooked the interpretability of DL models in MER.

Emotion recognition is a key task in MIR [9], and has been investigated in the context of smart musical instruments [10], hinting at how MER can be integrated into an Internet of Musical Things (IoMusT) ecosystem.

In this paper, we present XMERApp, a webapp for explainable MER for non-technically inclined users and musicians. XMERApp integrates a DL model for classification of four main emotions (i.e., aggressiveness, relaxation, happiness, and sadness, covering the four quadrants of the Arousal–Valence space [11]). It provides three levels of explainability features: (i) the primary emotion associated with the entire musical excerpt, along with the per-class confidence scores; (ii) the temporal evolution of the recognition results; and (iii) explanations generated using LIME [4], which highlight the specific regions of the spectrogram that were most influential in the model's decisions.

Notice that, although some attempts have been made to explain MER systems (e.g., [7]), to the best of our knowledge, no contributions have focused on explaining MER models by adapting LIME and by providing three distinct explanation perspectives.

The main page of the application is shown in Fig. 1. The

♮ Equal Contribution

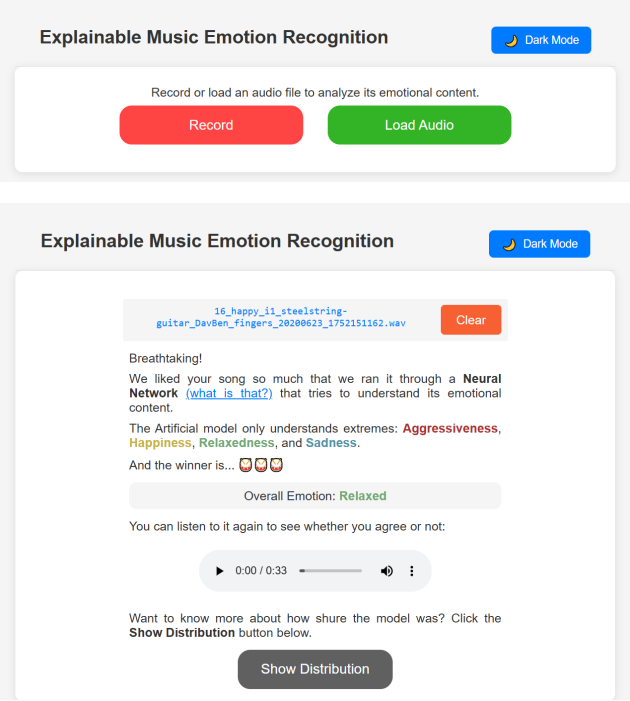

Fig. 1. First page of the explanation section of XMERApp. The app shows the overall classification and allows users to listen back to the recorded audio tune. The text appears gradually, paragraph by paragraph.

source code for XMERApp is made available as open-source in the project repository[1]

II. Methodology

The proposed explainable MER application enables musicians to record an improvisation or load a corresponding audio file; it processes the recorded signal with a DL MER classifier, and offers several levels of interactive explanations. The following subsection details the explainability features integrated in the webapp.

A. Explainability Integration

The explanations of the model's behavior integrated in XMERApp are organized into three levels. First, per-class confidence scores show the model's certainty for each emotion. Second, per-segment explanations reveal how emotional predictions evolve over time. Third, LIME-based interpretations highlight the spectrogram regions most influential to the model's decisions and enable the user to listen to them. The following subsections describe each level in detail.

1) Per-Class Confidence: While the concepts of classification techniques and MER may be well understood by technologists, XMERApp is intended for musicians who can be non-technically inclined, or may just not have been exposed to MER concepts. In particular, in previous user-studies with musicians (i.e., [10]), we found that many were asking questions about how a machine was understanding emotions, *"how sure it is"*, and what its constraints were (i.e., how many emotions it "knows", or how granular its understanding or emotional content was). Therefore, the first and highest-abstraction-level explanation–following the classification of the overall emotion of a recorded tune– consists of exposing the user to the following:

1) The classes of emotion understood by the model, and how it could lack fine granularity in describing the piece's emotional content.
2) Per-class confidence as percentages.

The domain of emotion categories is defined by the dataset, while the per-class confidence is represented by the softmax output of the model. These outputs are not calibrated probabilities but can be interpreted as relative indicators of the model's confidence across the different emotion classes. Future versions of the application could further refine these confidence estimates by introducing a calibration phase for the model output.

The first level of explanation can be seen in Figs. 1 and 2.

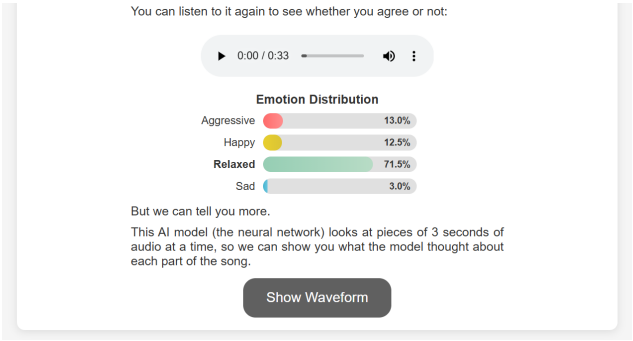

Fig. 2. A first level of explainability is provided by a bar chart displaying the pseudo probability values of each emotional state for the entire recording. The domain of emotions understood by the classifier is presented earlier (see Fig. 1).

2) Per-Segment Explanation: Following the introductory information about the model and its confidence, users are exposed to how the model classifies 3-second segments (also called slices) of audio, producing a "winning emotion" for each segment, as well as the confidence scores for the four categories. Per-segment winning categories are overlaid on the waveform, while confidence scores are displayed in a second diagram with connecting lines, showing the temporal evolution of the MER prediction (see Fig. 3).

Moreover, each segment in the waveform diagram can be clicked, opening a page with detailed information on the selected segment (see Fig. 4). Here, users can see the confidence score for each emotion category for the segment, listen to the isolated segment, and access the LIME-based explanation described in the next section.

Notice that this resolution provides interesting insights into the piece and resembles what a real listener might experience in practice: a musical piece can convey a specific emotion when heard in its entirety, or it can evoke different emotions when small segments are analyzed in isolation, without the context of the whole composition.

[1]https://github.com/cimil/explainable-music-emotion-recognition-app

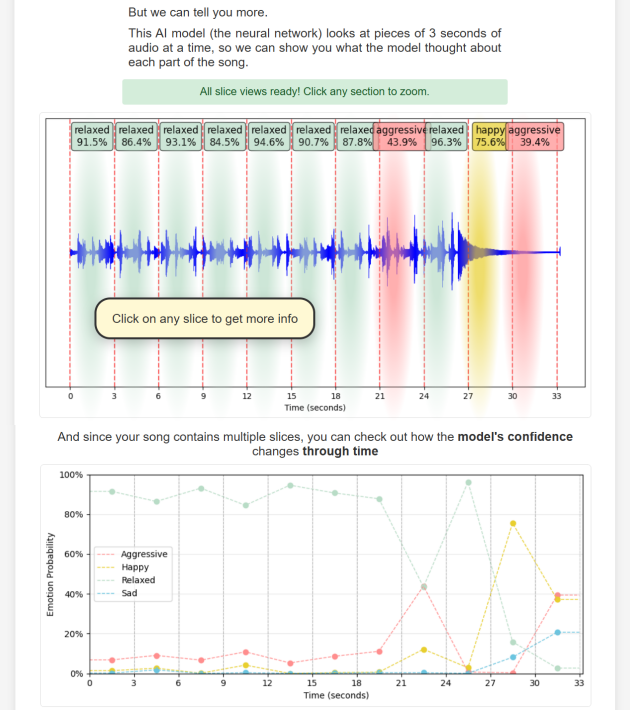

Fig. 3. Per-segment winning emotion and temporal evolution of the model's confidence with each category.

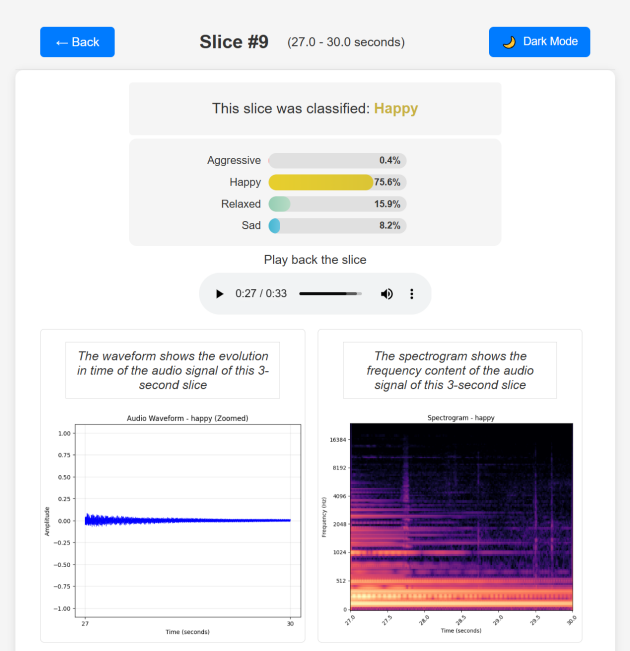

Fig. 4. Detailed view of a 3-second long slice of the improvisation.

3) Per-Segment LIME: LIME is a widely adopted explainability technique designed to provide local, human-understandable insights into the predictions made by complex machine learning models [4].

In essence, LIME explains which features are most relevant for the classification of a specific input sample by following these steps. First, it generates a set of perturbed samples that are similar to the original one, typically by selectively altering (e.g., turning on or off) some of its features. These perturbed samples are then labeled using the original model. Finally, an inherently interpretable model (such as a linear model or a decision tree) is trained on this newly generated dataset. By analyzing this interpretable model, it becomes possible to assess the contribution of each input feature to the original prediction. In the case of images, these features may correspond to specific regions or superpixels.

In the context of our work, we applied LIME to interpret the decisions made by the DL model when classifying emotions from musical excerpts.

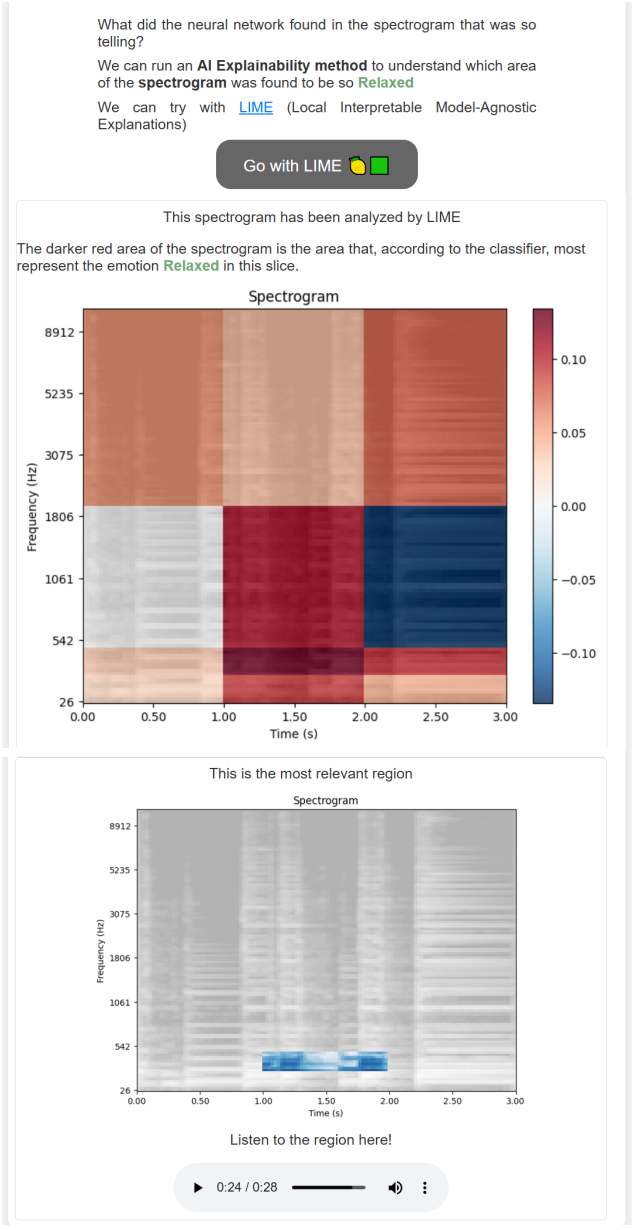

Fig. 5. View of the Interface for LIME explanations.

To adapt LIME for spectrogram-based audio classification,

we designed a custom segmentation scheme for the log mel spectrograms. Each spectrogram, corresponding to a 3-second audio segment, was divided into 12 non-overlapping regions by dividing the time axis into three equal parts and the frequency axis into four bands. The frequency bands were defined as follows: low (below 250 Hz), mid-low (250-500 Hz), mid-high (500-2000 Hz), and high (above 2000 Hz). These ranges align with commonly observed roles of frequency content in music, where lower frequencies are typically associated with rhythmic and bass elements, mid frequencies with melodic and harmonic content, and higher frequencies with timbral characteristics such as brightness and articulation. This segmentation enabled us to localize the model's attention across both the temporal and spectral dimensions. LIME returns a ranked list of the spectrogram regions based on their contribution to the model's output. This ranking provides interpretable insights into which portions of the spectrogram were most influential in the classification process, as inferred by the classifier. Furthermore, by isolating specific time segments and frequency bands, we were able to reconstruct and listen to the corresponding audio components. The original audio sample is preserved and then segmented and filtered according to the information provided by LIME regarding the most relevant regions. This enables a direct auditory analysis of the features driving the model's decision.

B. Dataset and Preprocessing

Our classifier was trained on the dataset described in [12], which consists of 391 original short guitar compositions obtained by musicians improvising on acoustic and classical guitars while trying to convey a specific emotion. The audio excerpts vary in length, ranging from 12.4 seconds to 75.5 seconds. The ground truth includes four possible emotion labels–aggressive, relaxed, happy, and sad.

In order to obtain the label for each piece, each track was rated by 16 listeners using a seven-point scale from -3 to $+3$ for each of the four emotions. For example, a rating of $(3, -3, 1, 0)$ indicates that a listener perceived the piece as strongly aggressive, not relaxed at all, slightly happy, and emotionally neutral in terms of sadness.

Therefore, as a first step, we averaged the 16 listener ratings for each piece and assigned the emotion with the highest average score as the label for classification, thus formulating a multi-class classification problem. Four compositions were excluded because they elicited multiple emotions with equal maximum scores, reflecting ambivalent emotional content [12].

For the preprocessing phase, each audio file was downsampled to 22,050 Hz and segmented into 3-second clips, as recommended in [12]. We then extracted log mel spectrograms from these segments using a short-time Fourier transform with a frame size of 2048 samples, a hop length of 512 samples, and 128 mel bands-parameters commonly used in Music Information Retrieval tasks (e.g., [13]).

C. Neural Network Architecture

The classification model is a Convolutional Neural Network (CNN) implemented using the Keras API with a TensorFlow backend. Its structure resembles models commonly used in MIR tasks [13], [14], and the main hyperparameters were optimized through multiple rounds of random search.

The network architecture comprises five convolutional layers with 16, 32, 32, 64, and 64 filters, respectively. Each convolutional layer uses 3×3 kernels with rectified linear unit (ReLU) activations, followed by a max-pooling layer with a 2×2 window and a stride of 2, which progressively reduces the spatial dimensions and computational complexity. Batch normalization is applied after each convolutional layer to stabilize and accelerate the training process. The convolutional stack is followed by a fully connected dense layer with 32 units, and a dropout rate of 0.5 is applied to reduce overfitting.

The model is trained using the Adam optimizer [15] with a learning rate of 1×10^{-5}. Sparse categorical cross-entropy is used as the loss function. Training is conducted over 50 epochs with a batch size of 64.

The trained model can be found in the same project repository referenced earlier in the footnote.

D. Web Application Architecture

XMERApp was developed as a webapp, employing a Python Flask backend, and an HTML/CSS/JavaScript frontend interface. The Python backend eases the process of inference for the MER model, which is a TensorFlow/Keras model that is loaded from an HDF5 file. Audio recording is offered through the JACK-Python integration when running locally, but loading of audio files is provided through the Javascript frontend, enabling deployment to a webserver.

III. Classification Results

We used 20% of the entire dataset as a test set, and 20% of the remaining samples for validation. We ensured that pieces performed by the same musicians were included only in either the training or test set, but not both, to avoid the so-called *artist effect*, where the model learns individual performers' unique styles, limiting its generalization [16], [17].

As previously mentioned, each audio file was divided into 3-second segments before preprocessing. Predictions for each segment were then aggregated using a soft-voting technique to compute the classification accuracy at the full piece level.

The resulting accuracy at the segment and song level (i.e., after soft-voting) is 53.12% and 58.33%, respectively. In Table I, we report the classification metrics associated with the proposed model. Note that the table refers to results computed at the segment level.

Table I shows that the model performs best on the *Aggressive* class, with high recall (82.3%) and F1-score (0.697). In contrast, the *Happy* class is the most challenging, with a recall of only 28.1%. The *Relaxed* and *Sad* classes show moderate performance. Overall, the segment-level accuracy is 53.1%,

TABLE I
SEGMENT-LEVEL CLASSIFICATION REPORT

Class	Precision	Recall	F1-score	Support
Aggressive	0.605	0.823	0.697	147
Relaxed	0.493	0.659	0.564	208
Happy	0.547	0.281	0.371	146
Sad	0.488	0.341	0.401	173
Accuracy			0.531	674
Macro Avg	0.533	0.526	0.508	674
Weighted Avg	0.528	0.531	0.509	674

reflecting the difficulty of the task and the limited separability between classes.

While the performance of the current model is modest, the proposed XMERApp allows users to easily replace the classification model, enabling the integration of higher-performing TensorFlow models. This also includes using different model types, which is possible because LIME treats the model as a black box and therefore does not require a specific model typology (i.e., the CNN in our case) to function.

IV. Conclusion and Future Work

This work presents a web application for explainable MER for guitar performances. The integration of various levels of explainability, including LIME, provides valuable insights into the model's decision-making process, aimed at making the system more understandable for both researchers and musicians. An early informal evaluation with musicians provided promising feedback on the gradual introduction to the explainability concepts of XMERApp granted by the interface. The main limitation of the current stage of this work is the lack of an in-depth evaluation of the app's functionality, along with the relatively low accuracy of the model employed, which can be attributed to the challenging nature of music emotion classification on a limited dataset. Future work will focus on a more in-depth model search, exploring techniques such as data augmentation and transfer learning to improve the model's accuracy. In parallel, we plan to conduct user studies to assess the app's explainability and improve usability. Finally, while current explainability features are limited to LIME, we aim to integrate additional methods such as SHAP and Grad-CAM, and to combine them into a robust ensemble framework.

Acknowledgment

This work has been supported by the MUSMET project funded by the European Innovation Council (Grant n. 101184379). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] C. Molnar, *Interpretable Machine Learning*, 3rd ed. Self-published, 2025. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[2] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[3] S. Mishra, B. L. Sturm, and S. Dixon, "Local interpretable model-agnostic explanations for music content analysis." in *ISMIR*, vol. 53, 2017, pp. 537–543.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you? explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[5] V. Haunschmid, E. Manilow, and G. Widmer, "audiolime: Listenable explanations using source separation," in *13th International Workshop on Machine Learning and Music (MML 2020), held with ECML-PKDD 2020*, 2020, p. 20.

[6] K. Choi, G. Fazekas, and M. Sandler, "Explaining deep convolutional neural networks on music classification," *arXiv preprint arXiv:1607.02444*, 2016.

[7] S. Chowdhury, A. V. Portabella, V. Haunschmid, and G. Widmer, "Towards explainable music emotion recognition: The route via mid-level features," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2019, pp. 237–243. [Online]. Available: https://doi.org/10.5281/zenodo.3527788

[8] C. Wang, V. Lostanlen, and M. Lagrange, "Explainable audio classification of playing techniques with layer-wise relevance propagation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[9] A. C. M. D. Silva, D. F. Silva, and R. M. Marcacini, "Heterogeneous graph neural network for music emotion recognition," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. ISMIR, Dec. 2022, pp. 667–674.

[10] L. Turchet, D. Stefani, and J. Pauwels, "Musician-ai partnership mediated by emotionally-aware smart musical instruments," *International Journal of Human-Computer Studies*, vol. 191, p. 103340, 2024.

[11] P. N. Juslin and J. A. Sloboda, *Music and emotion: Theory and research*. Oxford university press, 2001.

[12] L. Turchet and J. Pauwels, "Music emotion recognition: intention of composers-performers versus perception of musicians, non-musicians, and listening machines," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 305–316, 2021.

[13] M. Comunità, D. Stowell, and J. D. Reiss, "Guitar Effects Recognition and Parameter Estimation with Convolutional Neural Networks," *Journal of the Audio Engineering Society*, vol. 69, no. 7/8, pp. 594–604, 2020.

[14] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *19th International Society for Music Information Retrieval Conference (ISMIR2018)*, 2018.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, arXiv preprint arXiv:1412.6980.

[16] E. Pampalk, A. Flexer, G. Widmer *et al.*, "Improvements of audio-based music similarity and genre classificaton." in *Proc. 6th Int. Society for Music Information Retrieval Conference (ISMIR)*, vol. 5. London, UK, 2005, pp. 634–637.

[17] A. Flexer and D. Schnitzer, "Effects of album and artist filters in audio similarity computed for very large music databases," *Computer Music Journal*, vol. 34, no. 3, pp. 20–28, 2010.