

Translation of Bass Sound Frequencies with Artificial Intelligence

M. en C. Jaime Hugo Puebla Lomas

ingpuebla@gmail.com

Ing. Saúl De la O Torres

sdelaot@gmail.com

Escuela Superior de Cómputo I.P.N. México D.F.

Department of Computer Systems Engineering

Received 23 November 2024; Accepted 03 December 2024

Abstract - This terminal work introduces an artificial intelligence system for automatic identification of electric bass notes. Using digital signal processing and AI techniques, the system analyzes audio files in WAV format and recognizes musical notes. It's especially useful for music beginners, facilitating note identification, although it's limited to bass melodies in standard pitch (440 Hz). The system generates a PDF file with the detected notes, offering an effective tool for musical transcription and learning the electric bass.

Keywords - Neural Networks, Machine Learning, Automatic Generation, Musical Note Detection, Short Time Fourier Transform, Chromaticity.

I. INTRODUCTION

Audio signal processing and artificial intelligence have converged to transform a variety of applications, from speech recognition to music analysis. Specifically, note detection on instruments such as the electric bass represents an emerging field of study, crucial for music pedagogy and automatic transcription. Despite technological advances, there is a notable lack of accurate and efficient tools for the automatic identification of electric bass notes, which limits both performance and accurate music documentation.

In this context, our work focuses on the development of an artificial intelligence system that processes audio signals in WAV format to identify electric bass notes. This system integrates advanced signal processing and machine learning techniques, taking advantage of methodologies such as the Fourier Transform for feature extraction and the use of neural networks for accurate note classification. In addition, the project addresses the need for accessible and accurate music transcription by generating PDF files documenting detected compositions or improvisations. In the article "Automatic music transcription: challenges and future directions" [1] it is concluded that automatic music transcription still has significant challenges to overcome in order to improve the accuracy and performance of transcription systems, it also clarifies questions of the project "Why focus on a single instrument?", the article concludes that current transcription methods use general purpose models that fail to capture the diversity found in musical signals, the paper suggests that one way to improve the performance of transcription systems is to adapt the algorithms to specific use cases.

The relevance of this project lies in its potential to support self-taught learning and facilitate the teaching of the electric bass, as well as in its contribution to the field of artificial intelligence applied to music analysis. This terminal work not only offers a technical solution to a specific problem, but also explores new possibilities at the intersection of technology, music and education.

II. METHODOLOGY

The design of the architecture of our artificial intelligence system for electric bass note identification is a meticulously organized structure that is articulated in five sequential processes. Each process is critical and must be successfully completed before moving on to the next, thus ensuring the integrity and effectiveness of the system.

The first process focuses on data entry, where the user enters a WAV file containing the recording of the melody to be analyzed. Next, the system proceeds to preprocessing and feature extraction. Here, the Short Time Fourier Transform is the main technique used to decompose the audio signal and extract the distinctive features necessary for note identification. The third process is machine learning, where a previously selected and trained Machine Learning model classifies the extracted features to identify individual notes. This step is crucial, as the

accuracy of the system depends on the model's ability to correctly recognize and classify the audio signals. The fourth and final process is document generation. Here, the system produces a PDF document that documents the identified notes.

Each of these processes is essential to the holistic operation of the system and has been designed to ensure a smooth and logical transition from one step to the next. The effectiveness of this architecture has been verified through rigorous testing and detailed implementation, ensuring its reliability and robustness.

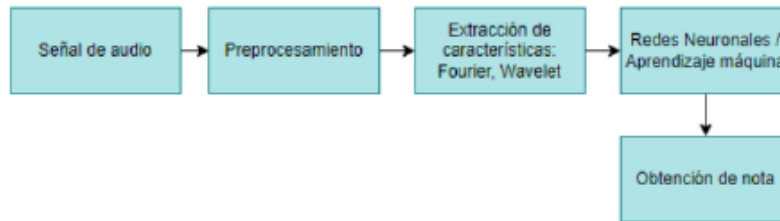


Fig. 1. System Architecture Diagram.

A. Design of the preprocessing and signal processing modules.

The design of the integral system for audio signal processing is structured in successive and correlated stages, guaranteeing the cohesion and methodological fluency necessary for the achievement of reliable and reproducible results. This process is based on the use of Python and the librosa library, the latter being a critical component for the manipulation and analysis of WAV format files.

1. WAV File Capture and Conversion: The first phase of the processing consists of capturing the audio file in WAV format. Using the book functionality, the file is read and transformed into a digital array, providing an accurate database for acoustic analysis. The audio signal is sampled at 44.1 kHz, an industry standard, ensuring a high-fidelity representation.

2. Note Segmentation and Isolation: Preprocessing focuses on identifying onsets within the audio signal, segmenting the recording into discrete fragments representing each individual note. The onset_detect function of librosa is crucial in this step, allowing accurate and effective segmentation for the independent treatment of each note.

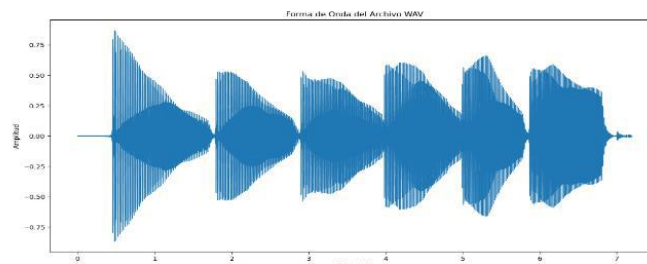


Fig. 2. Waveform of a melody (.wav).

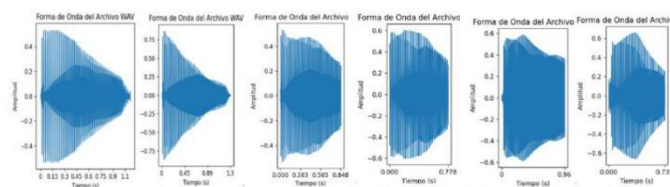


Fig. 3. Resulting segments after applying the onset_detect function.

3. Chromatic and Frequency Analysis: Next, we implement the Short Time Fourier Transform (1) to generate the chroma diagram, which is essential for the identification of the notes present in the audio. In conjunction, the PYIN algorithm is used to determine the fundamental frequency of each fragment, providing a detailed and accurate acoustic characterization.

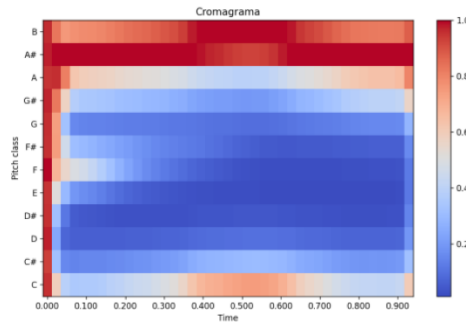


Fig. 4. Chroma key extracted from the WAV file of an example segment.

4. Detailed Feature Extraction: Beyond the basic identification of notes and fundamental frequencies, we proceed to the extraction of a broader set of acoustic features, including MFCCs, melspectrogram, spectral contrast, tonnetz and spectral rolloff. These additional parameters provide a wealth of information that is vital for classification and analysis by machine learning models.

B. Practical Implementation of the Processing Modules:

We developed a specific algorithm in Python to process the audio signal. This algorithm analyzes the signal and extracts a feature vector for each identified note, paving the way for interpretation and classification by advanced artificial intelligence techniques.

1. Audio Signal Preprocessing: This stage focuses on audio preparation, generating individual files for each note from onsets detection. This preparation is fundamental to aligning the training data with the AI model.

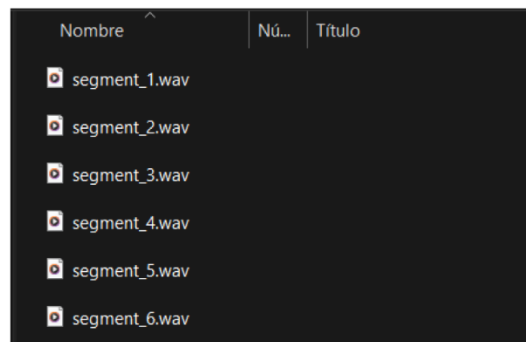


Fig. 5. How melody segments are saved for later use.

2. Advanced Signal Processing and Analysis: In this phase advanced signal processing techniques are applied to determine the specific note and fundamental frequency, calculate the corresponding octave and extract auxiliary features that provide a complete acoustic profile of each segment.

3. Comprehensive Feature Extraction: Finally, the librosa library is used to extract a comprehensive set of acoustic features, from MFCCs to the spectral rolloff, significantly enriching the dataset and enabling accurate classification and efficient machine learning.

Each of these stages is intrinsically connected, forming a cohesive workflow that facilitates the smooth transition from one step to the next, ensuring that each segment of the audio signal is analyzed with the utmost rigor and accuracy. This flow guarantees the integrity and quality of the data that will feed the machine learning model, meeting the exacting standards of academic research and practical applications in the field of computer systems engineering.

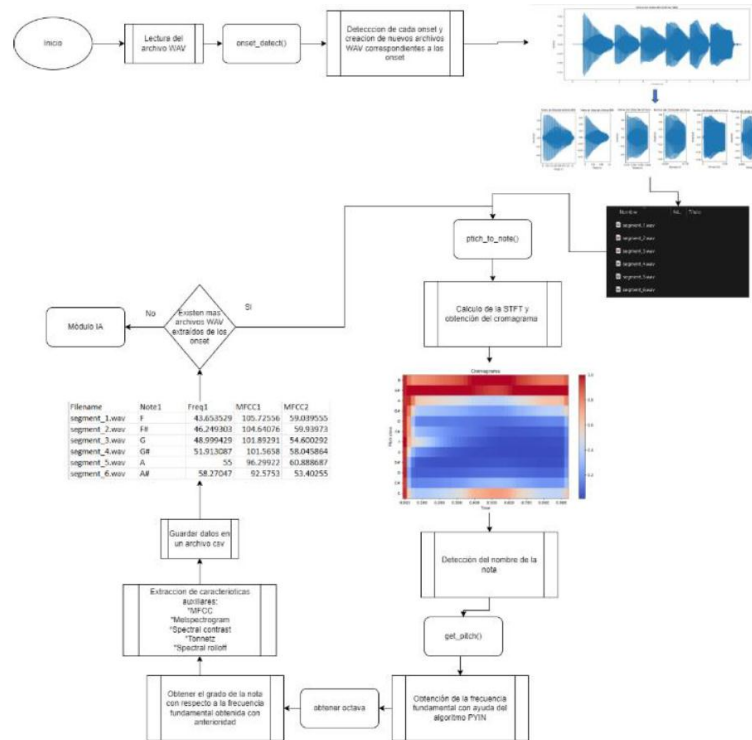


Fig. 6. Flow chart showing all the steps followed by the preprocessing and processing module before moving on to the AI module.

C. AI module design.

1. AI Model Selection: In the diversified ecosystem of machine learning models, Support Vector Machines (SVMs) stand out for their superlative performance in classification and regression in high-dimensionality contexts, such as audio signal analysis. These models can discern intricate patterns in nonlinear data, which makes them ideal tools for the identification of acoustic complexities inherent in the signal of an electric bass guitar.

2. Training and Validation: The data corpus was segregated into an 80/20 distribution for training and evaluation respectively, a standardized practice that mitigates overfitting and promotes generalization, as postulated by James et al. (2013) in “An Introduction to Statistical Learning” [2].

3. Optimization and Fine Tuning: Optimization of the SVM model was consummated using GridSearchCV in conjunction with K-Fold cross-validation, recommended by Murphy (2012) in “Machine Learning: A Probabilistic Perspective” [3]. This meticulous procedure ensures the selection of more efficient parameters and the robustness of the model by ensuring that every fraction of the data set is used for validation.

4. Integration with the Main System: After optimization and corroboration, the SVM model is amalgamated with the main system. This step is critical for systemic synergy, allowing the model to process the electrical bass audio signals accurately and generate the predictions required for the subsequent generation of the PDF documentation.

D. SVM implementation. We implemented a cascaded SVM system, selected for its ability to handle the classification of complex, nonlinear data. We leveraged the IDMT-SMT-Bass dataset for SVM model training, which comprises a vast collection of electric bass recordings in varied styles and techniques, providing fertile ground for acoustic feature extraction and model training with surgical precision.

1. Cascaded SVM Strategy: The cascaded SVM architecture, suggested by Abe (2010) in “Support Vector Machines for Pattern Classification” [4], enables an increase in the accuracy of audio signal feature identification. This strategy is segmented into distinct phases:

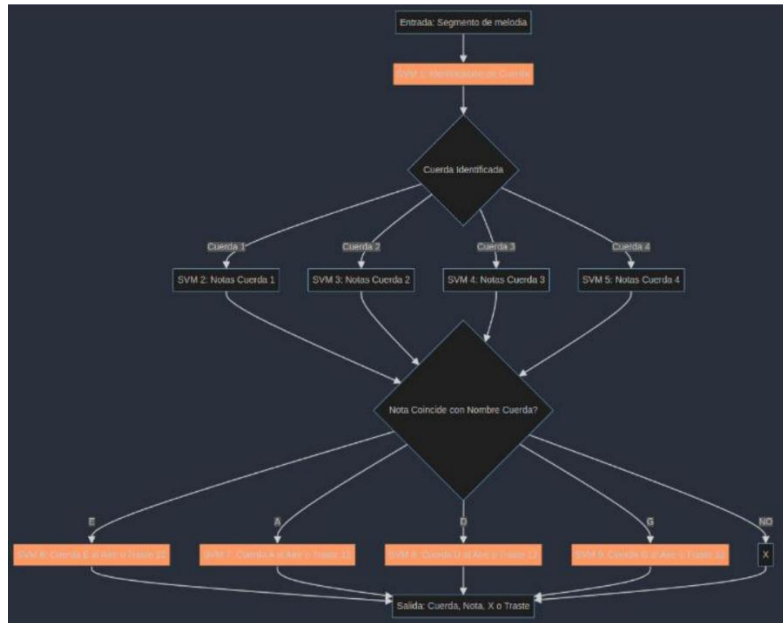


Fig. 7. Flow diagram of the cascade structure of the AI module.

2. Detailed Implementation: Data preparation and set partitioning was performed following modern preprocessing protocols, employing techniques such as one-hot coding and normalization, which facilitates efficient processing and accurate classification through Scikit-Learn pipelines, as documented in “Introduction to Machine Learning with Python” by Müller & Guido (2016) [5].

3. Training and Model Selection Process: SVM training was supported by parameter optimization using GridSearchCV and K-Fold cross-validation, with the objective of refining the models and verifying their robustness [6].

4. Integration and Workflow: The integration process ensures that each SVM classifies a particular aspect of the audio signals, and the amalgamated results provide accurate note identification, reinforcing the coherence and efficiency of the whole system [7].

E. Interface implementation.

The user interface constitutes the communication interface between the system and the end user. Its development, carried out in the Python programming language, integrates advanced libraries to guarantee an intuitive and functional experience.

1. PyPDF2: This library is instrumental in manipulation of PDF documents. It has been used for operations such as merging, segmenting and editing PDF document pages, as well as text extraction, positioning itself as an essential resource in file processing [8].

2. ReportLab: With ReportLab, we have orchestrated the creation of PDF documents from scratch, providing exhaustive control over document structure and layout. This tool has facilitated the inclusion of text, graphics and other visual elements, resulting in high quality, highly customized PDF documents [9].

3. Tkinter and the Construction of the GUI: Tkinter, the standard Python GUI library, has been the cornerstone for building the GUI. It has been chosen for its simplicity and efficiency in creating windows, buttons and other essential widgets. Its portability, as part of the Python standard library, ensures accessibility and compatibility across platforms [10].

Interface Visualization



Fig. 8. Program interface.

Notas Extraídas

Cuerda: 4, Nota: 2, Etiqueta2: X

Cuerda: 4, Nota: 9, Etiqueta2: X

Cuerda: 1, Nota: 7, Etiqueta2: X

Cuerda: 4, Nota: 2, Etiqueta2: X

Fig. 9. Example of system output printed in PDF.

III. RESULTS

At the culmination of our terminal work, we consolidated the evaluations of the SVM architecture in cascade designed for audio signal processing. Meticulously applied cross-validation revealed a 93.18% accuracy in string identification, while the classification of individual notes per string averaged an accuracy of 77%. Situational SVMs, dedicated to distinguishing cases, achieved an exemplary 100% accuracy. Collectively, these results culminated in an overall average accuracy of 71%, evidencing the effectiveness of the developed machine learning system and highlighting its potential in practical applications.

The user interface, designed to facilitate interaction with the system, proved intuitive and efficient, although we identified opportunities to improve accessibility and visual design. Usability was complemented by optimized computational performance, with response times reflecting a balance between accuracy and operational efficiency.

IV. CONCLUSIONS

From a theoretical perspective, the project advances the analytical understanding of music, offering new tools for music composition, education and production. Practically, the program is positioned to assist in music transcription and the creation of musical accessibility for the hearing impaired. Identified limitations, primarily in detecting low frequencies and interpreting advanced performance techniques, establish areas of focus for continued improvement.

Looking forward, exploration of more sophisticated techniques for note detection, employment of more advanced learning models, and collaboration with musical experts for practical validation are recommended. Optimizing processing time and expanding functionality to address diversity in interpretation and musical styles will also be crucial steps. This joint effort lays the groundwork for a music analysis tool that promises to revolutionize the intersection between technology and the art of music.

ACKNOWLEDGMENTS

The Authors would like to thank the Escuela Superior de Cómputo from Instituto Politécnico Nacional for the support received and the facilities granted for the development of this terminal work.