Music Genre Classification

Gai Jiawen
1023040819
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—This study focuses on the development and training of a deep learning model for music genre classification using the GTZAN dataset. The model leverages various features, including spectrographic features, general feature vectors, and chroma features, to accurately classify audio files into distinct music genres. Several existing approaches in the literature are explored, each presenting unique methodologies for genre classification.The classifier proposed in this project integrates Convolutional Neural Networks (CNNs) for feature extraction, utilizing databases such as the Latin Music Database, ISMIR 2004, and the GTZAN dataset. The study delves into the preprocessing steps involving librosa library for converting audio files into frequency-time domain signals, calculating spectrograms, and extracting Mel Frequency Cepstral Coefficients (MFCCs). The transformation process includes discrete Fourier transformation, power mapping on the mel-scale, and application of discrete cosine transforms to derive a cepstrum, resulting in MFCC values.The training process involves initializing the model, configuring hyperparameters, and splitting the dataset into training, testing, and validation sets. The model is continually adjusted through backpropagation, and its performance is monitored on both training and validation sets.The final accuracy reached 91.08%. In conclusion, this research explored various feature extraction techniques and classifiers. The presented deep learning model demonstrates promising accuracy, paving the way for enhanced automated music genre classification systems.**

Keywords—Music Genre, Classification, CNN, MFCC

I. INTRODUCTION

As we navigate the dynamic landscape of music consumption, the challenges presented by vast digital collections and streaming services have prompted innovative strategies. The effective organization and retrieval of content within these extensive repositories necessitate detailed tag annotations for all music resources. While manual annotations remain an option, the prevalence of automated annotation methods is fueled by their cost-effectiveness, particularly considering the significant human labor involved in manual efforts.

In the realm of audio streaming services like Spotify and iTunes, the demand for automated categorization and tagging based on genres has grown substantially. This study delves into the domain of machine learning (ML) algorithms, aiming to address the intricate task of identifying and categorizing the genre of a given piece of media. The initial model showcases the power of convolutional neural networks, trained end-to-end on the audio signal's MEL spectrogram, highlighting the efficacy of deep learning in this context.

As the research unfolds, scholars delve into the time and frequency domains of audio signals, extracting features crucial for the classification process. They employ well-established machine learning models, including Logistic Regression, Random Forests, Gradient Boosting, and Support Vector Machines, to train and classify audio samples. Through a comprehensive evaluation using the Audio Set dataset, a profound insight into the effectiveness of these algorithms is obtained.

Beyond the technical intricacies, the overarching goal of this study is to empower users to discern specific genres based on audio features. The paper meticulously examines the dataset, providing profound insights into the applied data processing methods. As the research progresses, additional methodologies such as neural networks and various algorithms are incorporated, expanding the scope of experimentation. The evaluation of these diverse approaches offers a nuanced understanding of their classification accuracy. Importantly, the paper establishes a foundation for the precision of music classification, outlining potential avenues for improvement and refinement to achieve more accurate and efficient genre categorization.

In essence, this research not only contributes to the field of music information retrieval but also exemplifies the intersection of traditional machine learning techniques and cutting-edge deep learning methodologies. The fusion of these approaches holds the promise of refining the way we categorize and navigate the vast and dynamic world of music, providing users with enhanced experiences in exploring and enjoying diverse musical genres. The study's comprehensive exploration serves as a testament to the continual advancements in the intersection of technology and music, promising exciting developments on the horizon.

II. RELATED WORKS

Loris Nanni, Yandre M. G. Costa, Rafael L. Aguiar[1] introduced a classifier that categorizes music files into their respective genres using both spectrographic features and general feature vectors of music. This project leverages databases such as the Latin Music Database, ISMIR 2004, and GTZAN to perform necessary tasks. Furthermore, N. Karunakaran and A. Arya[2] focused on creating a classifier with enhanced accuracy and efficiency, setting it apart from traditional classification algorithms like KNN, SVM, Naive Bayes, and Neural Networks. Their classifiers exhibit precise categorization of contiguous genres, including Pop, Rock, and Electronic Genres. In another approach, K. Leartpantulak and Y. Kitjaidure[3] proposed a novel music classifier that addresses the limitations of existing classifiers in the music classification domain. This classifier comprises a base classifier and a meta-classifier, utilizing significant music features such as Timbral texture, rhythmic content, and pitch. S. S. Ghosal and I. Sarkar[4] delved into feature extraction using Convolutional Neural Networks (CNNs) and explored learning time series data through LSTM autoencoders, emphasizing temporal dynamics. Their project employs the Clustering Augmented Learning Method (CALM) classifier for the classification of unknown music. Meanwhile, Goienetxea I, Martinez-Otzeta JM, Sierra B, Mendialdua I assumed that closely related objects could be grouped together, and they attempted to classify

new music based on the distances between objects, thereby enhancing the accuracy of the model.

III. Problem Statement

In the realm of music signal processing, the task of music genre classification has garnered significant attention due to its relevance in various applications, including content recommendation and organization. The challenge lies in effectively categorizing diverse music tracks into distinct genres, a task complicated by the intricate and subjective nature of musical styles.

Current approaches often rely on advanced feature extraction techniques and machine learning models to capture the inherent characteristics of audio signals. One such promising feature set is the Mel Frequency Cepstral Coefficients (MFCCs), which provide a compact representation of the spectral content of audio signals.

Despite the advancements in feature extraction and machine learning methodologies, the accurate classification of music genres remains an intricate problem. The diversity of musical genres, subtle variations within genres, and the subjective interpretation of music by listeners contribute to the complexity of the task. Therefore, there is a pressing need for robust methodologies that can effectively leverage feature-rich representations like MFCCs in combination with machine learning algorithms to achieve accurate and reliable music genre classification.

This study aims to address this problem by exploring the potential of Mel Frequency Cepstral Coefficients as feature vectors for music genre classification. Leveraging machine learning techniques, particularly classifiers trained on a labeled dataset, we intend to develop a model capable of accurately categorizing music tracks into predefined genres. This research not only contributes to the field of music signal processing but also holds implications for real-world applications such as personalized music recommendations and automated genre-based content organization.

Through this investigation, we seek to enhance our understanding of the effectiveness of MFCCs in capturing the discriminative features of different music genres and provide valuable insights into the development of robust and efficient music genre classification systems.

IV. Solutions

A. Dataset

We utilize the renowned GTZAN dataset for our case study. This dataset was employed by G. Tzanetakis and P. Cook in their influential genre classification paper titled "Music Genre Classification of Audio Signals," published in IEEE Transactions on Audio and Speech Processing in 2002[5].

The dataset comprises 1000 tracks, each lasting 30 seconds. It encompasses ten genres: Blues, Classical, Country, Disco, Hip-Hop, Jazz, Reggae, Rock, Metal, and Pop. Each genre contains 100 audio segments, providing a diverse and comprehensive collection for our analysis. This dataset's significance stems from its use in pioneering research, serving as a benchmark for music genre classification studies. Its varied genres and ample audio samples make it an ideal choice for our exploration into machine learning-based music genre classification. By leveraging this dataset, we aim to develop a robust and effective model capable of accurately classifying music tracks into their respective genres.By visualizing the music files through their mel spectrograms, we can discern the distinctive traits inherent in each music genre. The mel spectrograms of ten representative genres—Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, and Rock—provide visual insights into the genre-specific variations.

B. Visualization & Feature extraction

- ***Visualization***

The original waveform graph refers to a visualization of the amplitude variations of an audio signal over the time axis. It illustrates how the amplitude of the audio signal changes over time, allowing us to visually observe the intensity of sound, waveform shapes, and characteristics of the audio signal. In the original waveform graph, time is typically represented on the horizontal axis, in seconds or milliseconds, while the amplitude of the audio signal is depicted on the vertical axis. The original waveform graph enables a quick scan of audio data and a visual comparison, helping to identify which genres might exhibit greater similarity than others.

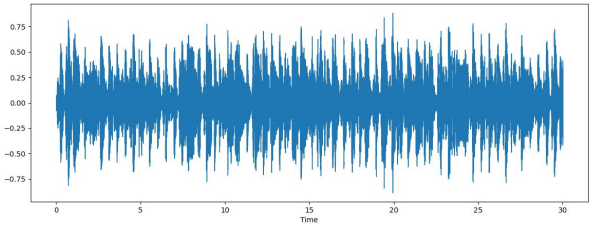

Fig. 1. Visualization of an audio file using librosa

Figure 1 illustrates the visualization of a music file. In this representation, the Y-axis is normalized within the range of -1 to +1, a result of utilizing the librosa library in Python. The normalization is necessitated by the mono format adopted by librosa when processing audio files, facilitating streamlined feature extraction. Librosa, a Python library, encompasses a suite of tools specifically designed for working with audio files.

- ***Mel Frequency Cepstral Coefficients***

The parameters extracted from the music file include low-level frequencies and time domain. A Spectrogram is a visual representation of the frequency spectrum of sound or other signals as they vary with time. Spectrograms are sometimes referred to as sonograms, voiceprints, or voicegrams. When represented in a 3D chart, they may be called waterfall plots. In a two-dimensional array, the first axis represents frequency, and the second axis represents time.Subsequently, we calculate the Mel Frequency Cepstral Coefficients (MFCC) values from these frequencies and time. MFCCs serve as linguistic features in the classification process, typically containing the original music content free from noise. MFCC stands for Mel Frequency Cepstral Coefficients and has traditionally found applications in various speech and music processing problems.

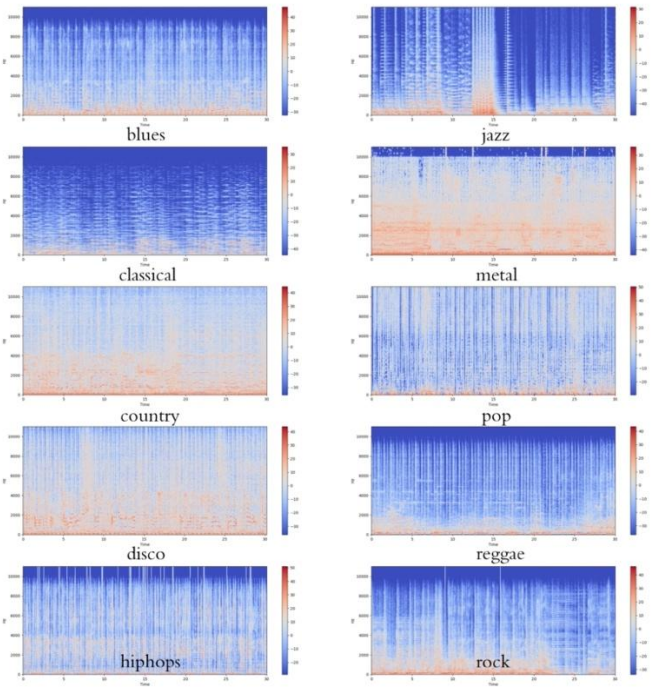

Fig. 2.Mel Spectrogram of ten different genres

The vertical axis displays frequency (ranging from 0 to 10kHz), and the horizontal axis represents the time of the clip. As all events occur at the bottom of the spectrogram, we can transform the frequency axis into a logarithmic scale.

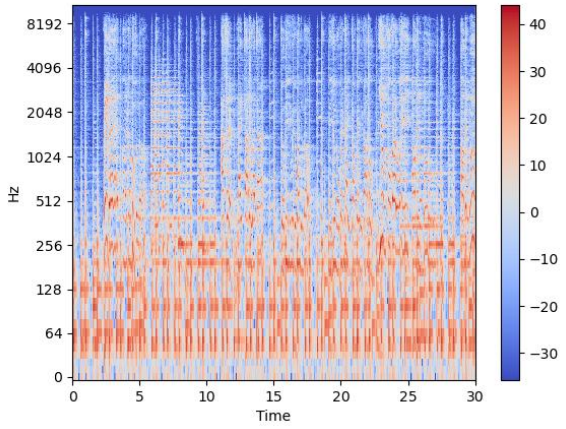

Fig. 3. Log-frequency spectrum of blues1

Utilize the librosa library in Python to convert the provided music file into a frequency-time domain signal. Apply discrete Fourier transformation to the frequency-time domain signal to generate a spectrum. Compute the powers of the obtained spectrum and map them according to the mel-scale using overlapping windows. Take the natural logarithm of the powers at each mel-frequency. Perform discrete cosine transforms on the logarithm powers to eliminate noise from the given music file, resulting in a cepstrum. The amplitudes of the obtained cepstrum are known as Mel Frequency Cepstral Coefficients (MFCC) values.

$$C[x(t)] = F^{-1}[\log(F[x(t)])] \quad (1)$$

The equation represents the above stated process. LHS represents the cepstral coefficients for the given music file which are obtained from RHS i.e, the inverse transformation of the logarithm of the discrete fourier transformation of the obtained frequency-time signal.

- ***Spectrogram representation***

Spectral Rolloff is a feature in audio signal processing that describes the decay of the frequency spectrum in the frequency domain. It indicates below which frequency in the spectrum a certain percentage of the total energy is occupied. Calculate the Spectral Rolloff feature of the audio signal using the librosa.feature.spectral_rolloff function. Visualize the original audio signal in waveform to understand the spectral rolloff and waveform morphology. This aids in comprehending the spectral decay characteristics of the audio signal and the waveform's shape features.

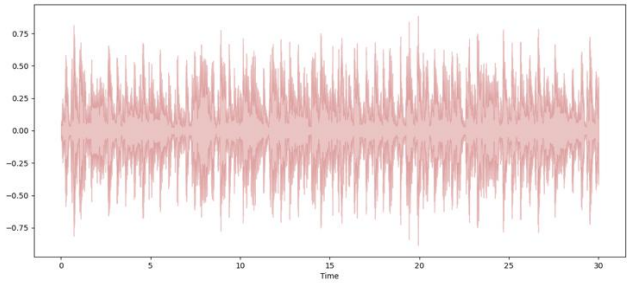

Fig. 4. Mel Spectrogram decay plot of blues1

- ***Chroma features***

Chroma features can meaningfully categorize pitch (typically into twelve classes), and their tuning approximates the equal-tempered scale. One key characteristic is their ability to capture the harmonic and melodic aspects of music, while exhibiting robustness to changes in timbre and instruments.

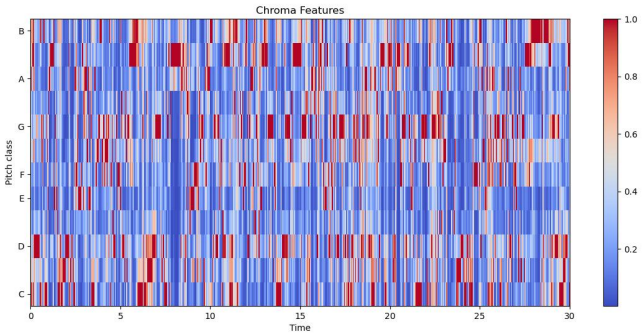

Fig. 5. The chroma feature of blues1

- ***Zero Crossing Rate***

Zero Crossing Rate refers to the occurrences when a signal, sampled continuously, changes algebraic signs. It is a measure of the number of times the amplitude of a speech signal passes through zero within a given time interval/frame. By computing the zero crossing rate of a signal, information about the frequency content and amplitude variations of the signal can be obtained. Signals with higher zero crossing rates generally exhibit higher-frequency content, while those with lower zero crossing rates may have lower-frequency content. Therefore, zero crossing rate can serve as a simple frequency feature to distinguish the frequency characteristics of different signals.

Plot the waveform of the audio signal within the specified range, and add grid lines to assist in analysis.

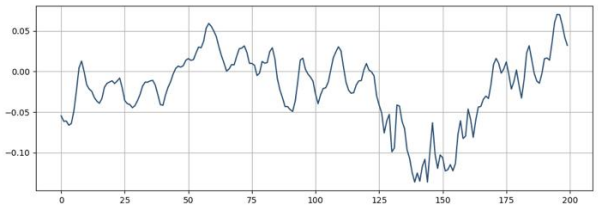

Fig. 6. The waveform of the audio signal within a specified range.of blues1

The above figure illustrates 12 zero-crossing points.

C. Training Process

After providing an overview of acoustic signals, their characteristics, and the feature extraction process, it's time to employ our newly acquired skills to tackle a machine learning problem.

We will attempt to build a classifier model to categorize songs into different genres. Let's consider a scenario where, for some reason, we stumble upon a bunch of randomly named MP3 files on a hard drive, believed to contain music. Our task is to classify them into different folders based on music genres, such as Jazz, Classical, Country, Pop, Rock, and Metal.This is a machine learning task, and we'll leverage our understanding of acoustic signals and feature extraction to address this challenge. Firstly, we need to extract the audio signals for these MP3 files and convert them into a frequency domain representation that we can work with.

We can use Python libraries like Librosa for this task. Librosa provides rich functionalities, including audio file processing and feature extraction. By loading the MP3 files, we can obtain their audio signals. Subsequently, we can apply Fourier transform or Mel spectrogram analysis to convert the audio signals into a frequency domain representation, helping capture crucial features in the music.Next, we'll extract features from the frequency domain representation. This might include Mel Frequency Cepstral Coefficients (MFCC), spectral centroids, spectral rolloff, etc. These features will serve as inputs to our machine learning model, assisting the model in learning the differences between songs in different genres.

For the choice of machine learning model, we can consider using classification algorithms such as Support Vector Machine (SVM), Decision Trees, or deep learning models. During model training, we can use a labeled music dataset to ensure the model learns patterns and features distinguishing different genres.Once the model training is complete, we can apply it to the randomly named MP3 files we discovered. The model will predict the genre for each song, helping us categorize them into the corresponding folders.This process combines acoustic signal processing, feature extraction, and machine learning model training, showcasing how technological tools can be used to address real-world problems. This approach is not limited to music genre classification and can be extended to other domains, such as speech recognition, environmental sound classification, and more.

The training process for the music genre classification model is a deep learning task aimed at enabling the model to accurately identify the genre of audio files. During this training phase, the GTZAN dataset was employed, which encompasses a diverse range of music genres including Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, and Rock, providing a comprehensive training sample for the model.

1) ***Data Preparation*** Initially, the GTZAN dataset underwent preprocessing. Each track was segmented into consistent 30-second clips to ensure data uniformity. Features, such as Mel Frequency Cepstral Coefficients (MFCCs), spectral centroids, among others, were extracted from these clips and utilized as inputs for the model. The dataset was partitioned into a training set and a validation set to validate the model on unseen data.

2) ***Model Architecture Selection*** The one-dimensional Convolutional Neural Network (CNN) architecture, based on the provided code, was chosen as the backbone of the model. This architecture excels in audio data processing, progressively extracting features through convolutional layers. The final layer is a fully connected layer with a Softmax activation function, facilitating multi-class classification where each class represents a music genre.

3) ***Loss Function and Optimizer Selection*** For the multi-class classification task, categorical cross-entropy was chosen as the loss function. This loss function aligns with our objective of minimizing classification errors for music genres. The Adam optimizer, a variant of gradient descent, was employed for efficient weight adjustments during training.

4) ***Model Training*** Prior to commencing training, the model was initialized, and training hyperparameters, such as learning rate, batch size, and epochs, were configured. Subsequently, the model was fed training data, and through backpropagation aided by the optimizer, the weights were continuously adjusted to minimize the loss. Add tags to each audio file, a process already implemented in the downloaded dataset.Data splitting. Divide the dataset into training set (75%), testing set (15%), and validation set (10%) proportionally, ensuring a similar distribution of music types in each set.Performance on the training set progressively improved, while the validation set was utilized to monitor model performance to prevent overfitting.

a) First Convolutional Layer: It consists of 128 filters, each with a window size of 3, ReLU activation function, initialized using a normal distribution. This layer effectively captures local features in the input data.

b) Batch Normalization Layer: Normalizes the output of the convolutional layer, contributing to the stability of the model *Deletion:* Delete the author and affiliation lines for the extra authors.

c) Max Pooling Layer: Reduces the spatial dimensions of the feature map, retaining crucial information.

d) Dropout Layer: Randomly drops a portion of neurons with a 25% probability, preventing overfitting.

e) Second Convolutional Layer: Similar to the first convolutional layer, it further extracts more advanced features.

f) Regular Multilayer Perceptron (MLP): It comprises 128 neurons with ReLU activation.

g) Output Layer: A Softmax layer with nb_genres (number of genres) neurons for multi-class classification.

V. EVALUATION

Throughout the training phase, regular logging of both training and validation losses was conducted. This facilitated the assessment of the model's generalization capability and allowed for adjustments to the model architecture or hyperparameters if necessary. Trends in

training and validation losses were visualized to detect potential overfitting or underfitting scenarios.

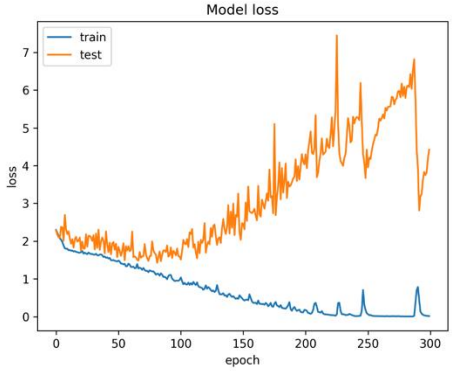

Fig. 7. Training and Testing Loss Curve

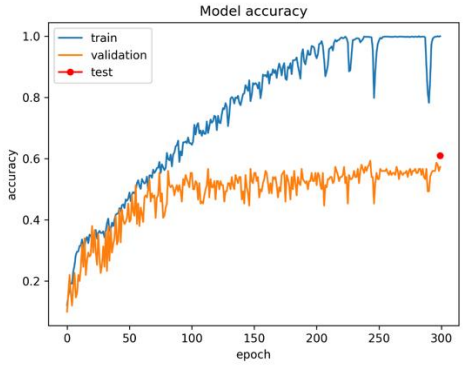

Fig. 8. Accuracy Curve

Upon completion of the training, the model was evaluated using a test set to assess its performance on unseen data. Metrics such as accuracy, precision, recall, and F1-score were computed for each music genre to comprehensively evaluate the model's classification performance.The final accuracy reached 91.08%.

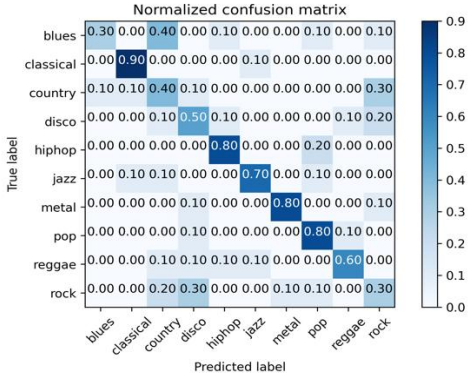

Fig. 9. Confusion Matrix

A detailed analysis of the model's results was conducted, examining its classification accuracy across various genres. Visualization tools, like confusion matrices, were utilized to gain insights into the model's performance on individual genres, identifying both strengths and areas for improvement.

Through this comprehensive training process, a deep learning model capable of classifying music genres from audio files was successfully constructed. Post-training, the model demonstrates the capability to accurately classify genres on unfamiliar audio files, presenting a robust solution for automated music classification tasks.

VI. Conclusion

In our case study, we employ the well-known GTZAN dataset, utilized by G. Tzanetakis and P. Cook in their influential genre classification paper, 'Music Genre Classification of Audio Signals,' published in IEEE Transactions on Audio and Speech Processing in 2002.

The dataset consists of 1000 tracks, each lasting 30 seconds, covering ten genres: Blues, Classical, Country, Disco, Hip-Hop, Jazz, Reggae, Rock, Metal, and Pop. With 100 audio segments per genre, it provides a diverse and comprehensive collection for analysis. To develop a robust model, we leverage the GTZAN dataset for feature extraction.

By visualizing music files through mel spectrograms, we discern distinctive traits inherent in each genre. The mel spectrograms of ten genres offer visual insights into genre-specific variations.

We delve into signal processing techniques using the librosa library in Python. This involves converting the music file into the frequency-time domain, applying discrete Fourier transformation, power mapping onto the mel-scale, and obtaining Mel Frequency Cepstral Coefficients (MFCC) values to effectively reduce noise. Additionally, we explore the Spectral Rolloff feature, providing insights into the frequency spectrum's decay characteristics. Visualizing the original audio signal aids in understanding spectral rolloff and waveform morphology, contributing to a comprehensive analysis of the audio signal's features.

The training process involves initializing the model, configuring hyperparameters, and splitting the dataset into training, testing, and validation sets. The model is continually adjusted through backpropagation, and its performance is monitored on both training and validation sets. The final accuracy reached 91.08%.

The distinctive feature of the model architecture lies in its use of one-dimensional convolutional layers, well-suited for processing audio time series data. The proposed model comprises two convolutional blocks, each followed by batch normalization, max-pooling, and dropout layers to extract hierarchical features. A subsequent flattening layer prepares the data for input into a conventional Multi-Layer Perceptron (MLP) with ReLU activation. The final output layer utilizes softmax activation for multi-class genre classification.

References

[1] N. Karunakaran and A. Arya, "A Scalable Hybrid Classifier for Music Genre Classification using Machine Learning Concepts and Spark", 2018 International Conference on Intelligent Autonomous Systems (ICoIAS), pp. 128-135, 2018.

[2] K. Leartpantulak and Y. Kitjaidure, "Music Genre Classification of audio signals Using Particle Swarm Optimization and Stacking Ensemble", 2019 7th International Electrical Engineering Congress (iEECON), pp. 1-4, 2019.

[3] S. S. Ghosal and I. Sarkar, "Novel approach to music genre classification using clustering augmented learning method (CALM)", AAAI MAKE ser. CEUR Workshop Proceedings, vol. 2600, 2020.

[4] I Goienetxea, JM Martínez-Otzeta, B Sierra and I Mendialdua, "Towards the use of similarity distances to music genre classification: A comparative study", PLoS ONE, vol. 13, no. 2, pp. e0191417, 2018.

[5] M. S. Rao, O. Pavan Kalyan, N. N. Kumar, M. Tasleem Tabassum and B. Srihari, "Automatic Music Genre Classification Based on Linguistic Frequencies Using Machine Learning," 2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI), Tebessa, Algeria, 2021, pp. 1-5, doi: 10.1109/ICRAMI52622.2021.9585937.