International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 05 | May 2025 www.irjet.net p-ISSN: 2395-0072

High-Dimensional Audio Genre Classification Using MFCC-Driven Multi-Layer Perceptrons

Chiranjeevi Vantaku

Department of Computer Science & Electronics and Communication Engineering
Maharaj Vijayaram Gajapathi Raj College of Engineering (MVGRCE),
Vizianagaram, Andhra Pradesh, India

Abstract - *Audio data, as a form of unstructured information, presents both opportunities and challenges in the field of machine learning. Classifying such data effectively requires thorough preprocessing to transform raw signals into meaningful features. This project focuses on building a music genre classifier using artificial neural networks, specifically a multi-class model capable of predicting one of ten genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. The goal is to develop a probabilistic model that not only identifies the most likely genre but also provides insight into the confidence of its predictions.*

Key Words: **Audio Classification**, **MFCC**, **Multilayer Perceptron**, **Deep Learning**, **Music Genre Recognition**, **Neural Networks**, **Signal Processing**, **Feature Extraction**, **PCA**, **Overfitting Mitigation**

1.INTRODUCTION

Audio classification has become an increasingly important area within machine learning, with applications spanning diverse industries. It sits at the intersection of signal processing, machine learning, and acoustics, enabling systems to interpret and categorize sound data. From detecting speech and analyzing environmental noise to recognizing emotional tone in human voices, audio classification supports a wide range of real-world use cases and continues to grow in relevance as audio data becomes more prevalent.

1.1 The Dataset

A critical foundation for effective audio classification lies in selecting a dataset that accurately represents the problem space. For music genre classification, an ideal dataset should maintain consistent audio quality, standardized file properties, and well-defined genre labels. The GTZAN Genre Collection, curated by George Tzanetakis, meets these requirements and serves as the primary dataset for this project. It consists of 1,000 audio tracks, evenly distributed across 10 distinct genres, with 100 tracks per genre. Each track is a 30-second .WAV file in lossless format, sampled uniformly at 22,050 Hz. This standardized sampling rate ensures reliable feature extraction across all tracks, minimizing discrepancies due to over- or under-sampling.

1.2 Preprocessing Methodology

Machine learning algorithms cannot directly interpret raw audio data due to its unstructured nature. To bridge this gap, the audio signals must first be transformed into structured numerical representations. In this project, the raw audio files were processed using signal transformation techniques to extract Mel Frequency Cepstral Coefficients (MFCCs) — a widely used feature set for audio classification tasks. MFCCs are particularly suited for music genre recognition because they reflect how humans perceive sound frequencies, emphasizing lower frequencies where the human ear is most sensitive. This perceptual alignment makes MFCCs highly effective for applications where subjective interpretation of sound plays a role.

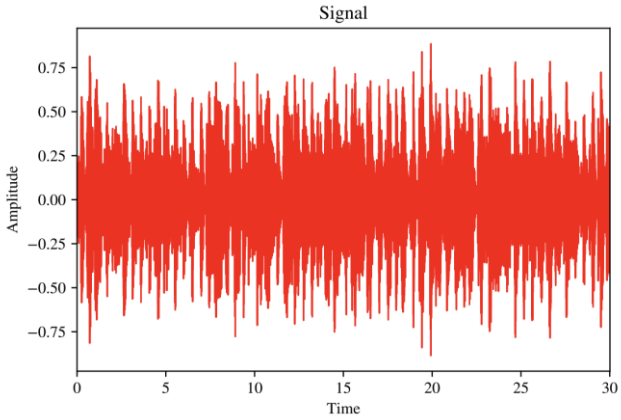

Figure 1: Visual Representation of Audio Signal

2. Methodology

2.1 Technologies Used

A range of Python-based libraries were employed to build and evaluate the music genre classification model. NumPy and Pandas served as the core numerical engines, enabling efficient handling of linear algebra operations and high-dimensional data structures. For audio preprocessing, the Librosa library was utilized to extract relevant audio features, including MFCCs. Visualization and performance tracking were conducted using Matplotlib, providing clarity on model behavior and data trends. Scikit-learn

facilitated the splitting of the dataset into training and testing sets, ensuring a structured pipeline. Finally, the classification model was constructed using Keras with a TensorFlow backend, offering a high-level API for building and training deep learning models.

2.2 Preprocessing Implementation

The initial step in building an effective music genre classifier involves standardizing the dataset to ensure consistent input for the machine learning model. This requires systematically extracting meaningful features from each audio file across all genre categories. For each segmented window of an audio track, Mel Frequency Cepstral Coefficients (MFCCs) were derived using a sequence of signal processing techniques, including Fast Fourier Transforms (FFT), Short-Time Fourier Transforms (STFT), and log-scaled spectrogram analysis. These methods collectively transform raw audio signals into structured representations that are suitable for training the model.

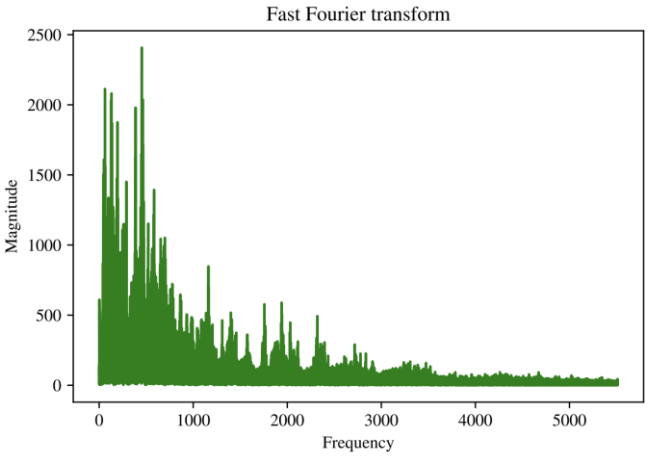

Figure 2: Fast Fourier Transform of Audio Signal (Librosa)

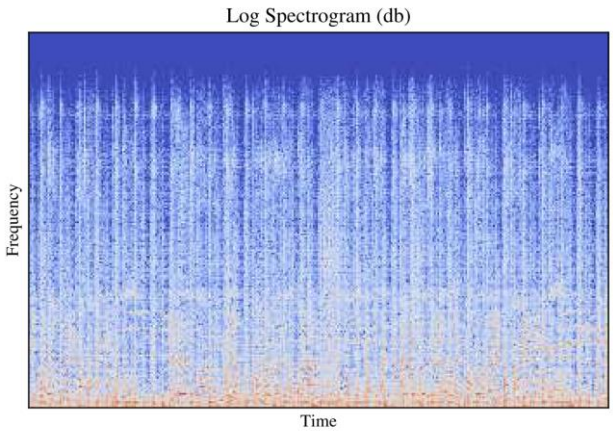

Figure 3: Example of Librosa Log Spectrogram

MFCCs capture key characteristics of an audio signal across three dimensions: frequency, magnitude, and time. In this project, each MFCC was calculated using 13 distinct coefficients, a standard choice in audio analysis tasks due to its balance between complexity and representational power [9]. These coefficients are stored in a 13-dimensional array, where the array's length corresponds to the number of MFCC windows derived from each audio track. The total number of samples per track is determined by multiplying the audio sample rate by the track's duration, ensuring consistent input dimensions across the dataset.

(22, 050Hz × 30sec) = Samples / track (1)

An additional benefit of using MFCCs over log spectrograms lies in their reduced dimensionality. While log spectrograms can produce thousands of features per audio sample, MFCCs are confined to just 13 predefined coefficients. This significant reduction in feature space makes MFCCs far more efficient for computationally demanding tasks like model training. Lower dimensionality not only decreases memory usage but also helps in reducing training time and mitigating the risk of overfitting, thereby improving overall model performance and scalability.

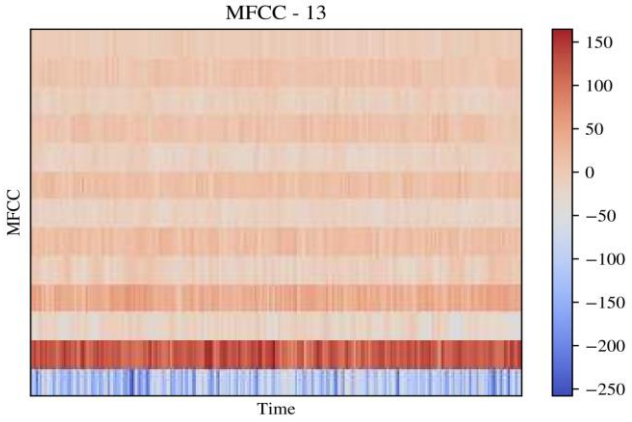

Figure 4: Example of Librosa MFCC Output

2.3 Exploratory Data Analysis

After converting the raw audio signals into structured numerical features, the dataset becomes suitable for modeling and pattern recognition. Prior to training, the data was divided into training and testing sets using Scikit-learn to ensure unbiased evaluation. Given the high-dimensional nature of the extracted MFCCs – each sample forming a matrix of size 130 × 13 – dimensionality reduction was applied to facilitate exploratory data analysis. Principal Component Analysis (PCA) was used to project the data into a lower-dimensional space. The resulting scatter plots revealed that the data is not linearly separable, which influenced the decision to adopt more

complex, non-linear classification models such as neural networks.

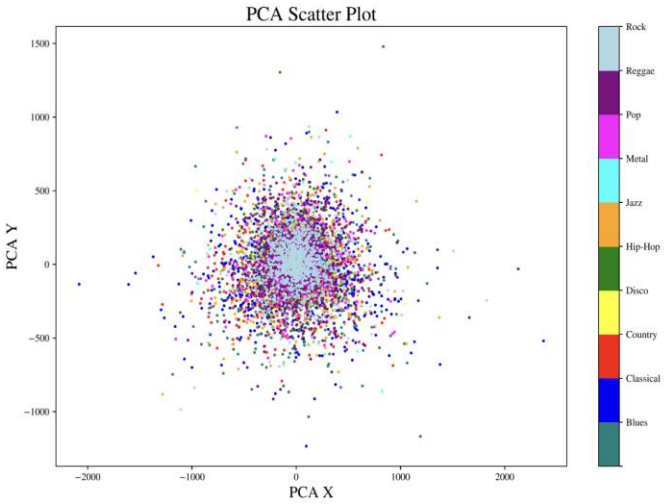

Figure 5: PCA 2D Decomposition

2.4 Model Selection

Due to the high dimensionality of the input features and the need to generate a multiclass probability distribution, a neural network architecture was chosen for this classification task. Specifically, a Multilayer Perceptron (MLP) was implemented using Keras, taking advantage of the flexibility and learning capacity of artificial neural networks (ANNs). The model architecture included four hidden layers, trained using the backpropagation algorithm to optimize weight adjustments based on error feedback. The layer configuration consisted of an input layer with 1690 units, followed by hidden layers of 1024, 512, 256, and 64 neurons respectively, and a final output layer of 10 nodes representing the target music genres. Model performance was evaluated using sparse categorical cross-entropy as the loss function, along with accuracy metrics on both training and validation datasets. These measures were instrumental in diagnosing overfitting and fine-tuning model parameters for improved generalization.

3 Machine Learning Techniques

3.1 Computational Complexity

To minimize computational complexity and reduce the training time during parameter tuning, a simplified version of the dataset was employed. This reduced set retained the structure of the original dataset but included only one representative song per genre instead of 100. By limiting the data in this way, the overall training time was decreased by a factor of 100, allowing for quicker experimentation and model iteration during the optimization phase.

3.2 Parameter Configuration

To streamline parameter optimization and reduce training time, a scaled-down version of the dataset was utilized. While structurally identical to the full dataset, this reduced set contained only one audio track per genre instead of 100. This strategic reduction significantly lowered computational demands, cutting training time by approximately 100-fold, and allowed for faster testing of model configurations.

Following the initial evaluation of model performance on both the training and validation sets, several adjustments were made to enhance learning efficiency. One key modification involved tuning the learning rate, as the model was converging too rapidly, potentially overlooking optimal solutions. To address this, the default Keras learning rate of 0.01 was reduced to 0.0005 through empirical testing. This lower learning rate slowed down convergence, allowing for more gradual and stable training. Additionally, increasing the number of epochs provided more opportunities for weight updates, leading to smoother and more interpretable learning curves. This adjustment helped mitigate the issue of overly steep loss and accuracy trends, making the training process easier to monitor and refine.

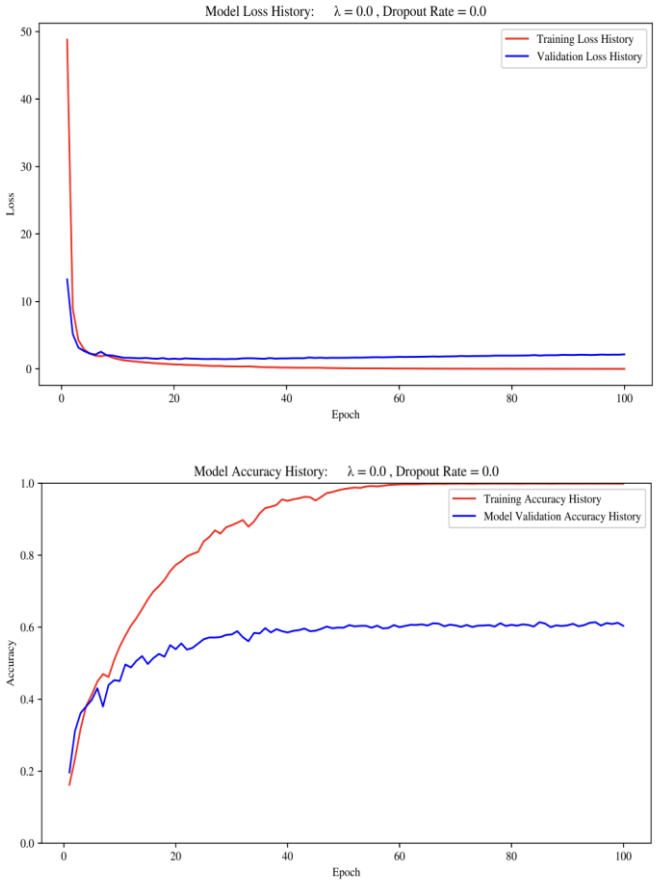

Figure 6: Standard Learning Functions

3.3 MLP Optimization

3.3.1 Hyper Parameterization

With the learning rate appropriately tuned, the next focus was on addressing the gap between training and validation performance, particularly to assess the model's ability to generalize to unseen data. A noticeable discrepancy, where validation loss exceeds training loss, is a clear indicator of overfitting. To mitigate this, regularization techniques were applied. Specifically, L2 regularization (also known as ridge regression) was introduced using a kernel regularizer, which penalizes large weight values to encourage simpler models. The regularization strength, denoted by λ (lambda), was initially set to 0.01. However, after evaluating convergence patterns using loss and accuracy plots, the optimal lambda value was found to be 0.001, effectively balancing model complexity and generalization

$$L_2 Regularization = \lambda \sum_{i=1}^{n} \theta_i^2$$

3.3.2 Dropout Layers

Beyond L2 regularization, additional measures were taken to further reduce overfitting. Dropout layers were introduced after each of the three dense layers to randomly deactivate a portion of neurons during training, promoting robustness in the network. The default Keras dropout rate of 0.3 was reduced to 0.2, which yielded better performance when used alongside the L2 regularization term. To complement these efforts, early stopping was also implemented as a final safeguard. With the model set to train for up to 100 epochs, early stopping allowed training to halt automatically if the validation loss ceased improving, thereby preventing unnecessary weight updates and reducing the risk of overfitting.

3.3.3 Activation Functions

Given the structure of artificial neural networks (ANNs), one critical challenge that arises during training is the vanishing gradient problem, particularly when using the backpropagation algorithm. This issue is common with traditional activation functions like sigmoid or hyperbolic tangent, whose gradients approach zero at extreme input values, leading to stalled learning in deeper layers. To address this, the Rectified Linear Unit (ReLU) activation function was applied to all hidden layers, as it maintains a stronger and more consistent gradient during training. The softmax function was reserved solely for the output layer, where it was used to produce a 10-class probability distribution corresponding to the target music genres.

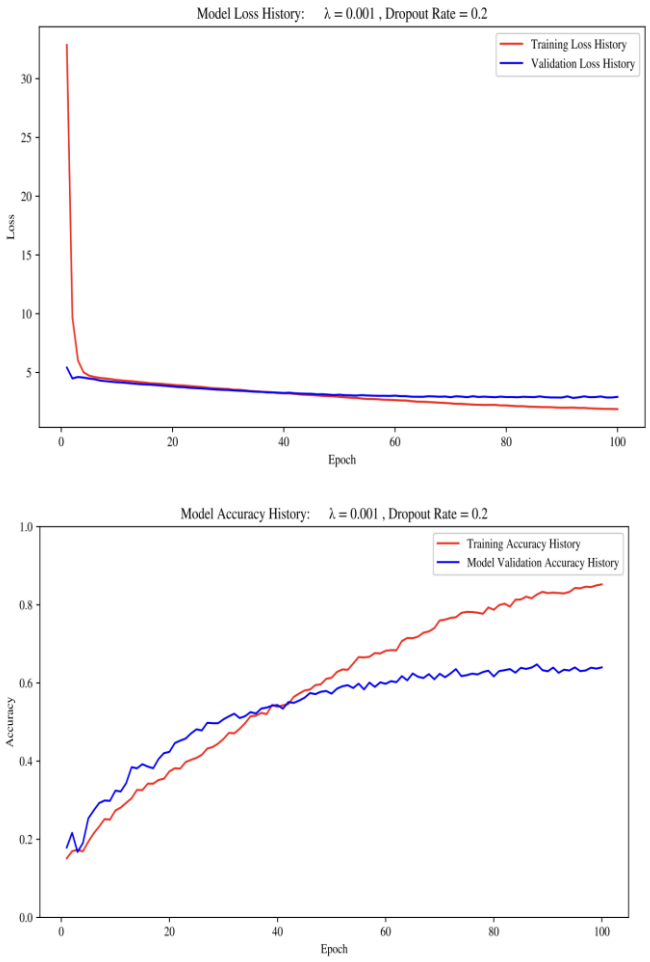

Figure 7: Regularized Learning Functions

4 Results

4.1 Classifications

After applying various optimization techniques to minimize error and overfitting, the model achieved a 64% accuracy on the validation dataset. This result is considered promising, especially given the model's probabilistic output design. Unlike single-label predictions, a multiclass probability distribution offers deeper insights by indicating the likelihood of each genre. For instance, if a song is misclassified, it is still valuable to observe that it shows high probabilities across related genres like pop, country, and rock, rather than receiving only the top predicted label. This richer output aids in understanding the model's behavior and the inherent overlap between music genres.

4.2 Cross Class Confusion Matrix

The confusion matrix shown in Figure 9 supports this observation by visually illustrating patterns in the model's predictions. A strong diagonal trend indicates high accuracy in correctly identifying individual genres,

suggesting solid single-class performance. In contrast, the lighter, off-diagonal regions reveal where misclassifications commonly occur—highlighting genre pairs that the model tends to confuse. This visualization provides valuable insights into both the strengths and limitations of the classifier.

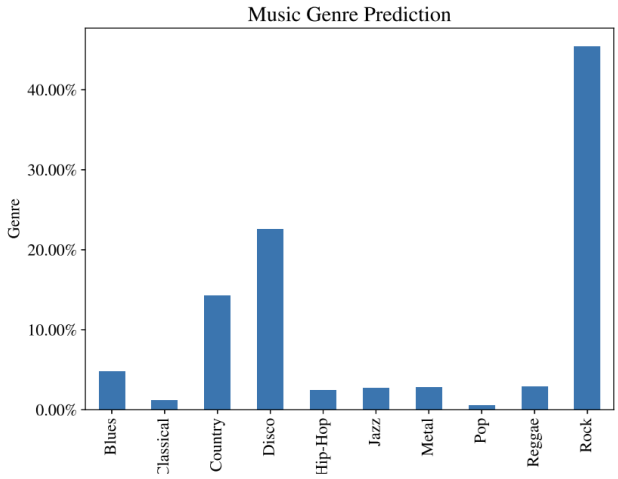

Figure 8: Multiclass Probability Distribution Output

The confusion matrix also sheds light on the genre-specific performance of the model. It clearly indicates that metal and classical music genres are classified with the highest accuracy, reflecting strong feature separation and consistent patterns within these classes. On the other hand, genres such as country, disco, hip hop, and rock exhibit comparatively weaker performance, likely due to overlapping audio characteristics and broader variability within those categories. This insight is valuable for identifying which genres may benefit from additional preprocessing or feature refinement.

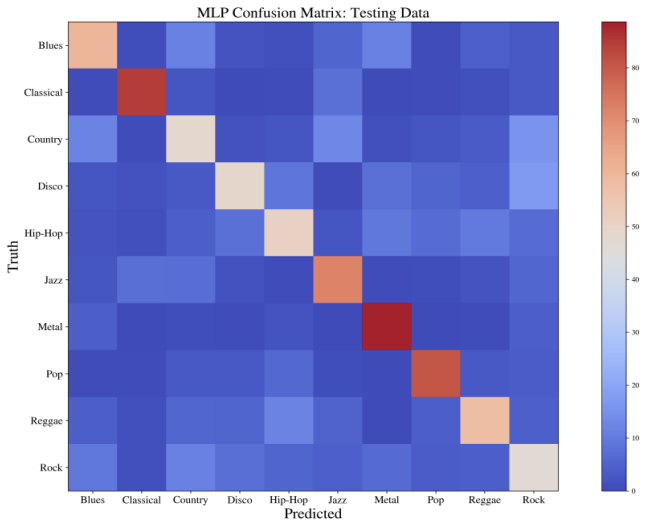

Figure 9: Validation Confusion Matrix

Conclusion

This study demonstrates the effectiveness of using MFCCs as structured features and MLPs as predictive models for audio genre classification. By focusing on dimensionality reduction, regularization techniques, and probabilistic output interpretation, we achieved a meaningful classification accuracy of 64% on the validation set. While certain genre pairs posed classification challenges due to overlapping acoustic features, the model showed strong performance on more distinct genres such as classical and metal. Future improvements involving deeper architectures, ensemble learning, and metadata integration could enhance performance further. The findings underscore the potential of deep learning in extracting and modeling patterns from high-dimensional unstructured audio data.

REFERENCES

1. George Tzanetakis and Perry Cook. Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing, 10(5), 2002.

2. Logan, B. Mel Frequency Cepstral Coefficients for Music Modeling. Proceedings of the International Symposium on Music Information Retrieval (ISMIR), 2000.

3. Deng, L., & Yu, D. Deep Learning: Methods and Applications. Foundations and Trends® in Signal Processing, 7(3–4), 197–387, 2014.

4. McFee, B., Raffel, C., Liang, D., Ellis, D.P.W., et al. Librosa: Audio and Music Signal Analysis in Python. Proceedings of the 14th Python in Science Conference (SciPy), 2015, https://librosa.org/doc/latest/index.html

5. Chollet, F. Keras: Deep Learning for Humans, https://github.com/keras-team/keras