Music Genre Classification

Submitted for

Artificial Intelligence and Machine Learning CSET301

Submitted by:

E23CSEU0302	**Harshit Dhar**
E23CSEU0304	**Aryan Tyagi**
E23CSEU0312	**Jayesh Anand**

Submitted to

DR. YAJNASENI DASH

Jan-Apr 2025

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

BENNETT UNIVERSITY

THE TIMES GROUP

INDEX

ABSTRACT

Music genre classification stands as a fundamental task in the broader domain of Music Information Retrieval (MIR), playing a critical role in improving the organization, recommendation, and personalization of digital music libraries. As music streaming services and personal music collections continue to grow rapidly, the ability to automatically and accurately categorize music by genre has become more important than ever before. Traditional methods, often relying on manual annotation or simple rule-based algorithms, have proven inadequate for the demands of modern music databases, which require scalable and highly accurate classification systems.

This project addresses the challenge of automatic music genre classification by leveraging the power of machine learning and deep learning techniques. Specifically, we employ a Convolutional Neural Network (CNN) trained on features extracted from the GTZAN dataset, a widely used benchmark in music genre research. Mel-spectrogram representations, which effectively capture both spectral and temporal characteristics of audio signals, are utilized as input features for the model.

The proposed deep learning architecture demonstrates a high level of accuracy, achieving over 90% on unseen test data. Through rigorous experimentation and evaluation, we show that CNNs are highly effective at learning complex and abstract patterns within music signals, surpassing traditional feature-based machine learning models. Our results affirm the viability of deep learning models for real-world music information retrieval systems and open the door for future enhancements such as real-time classification, transfer learning, and multi-genre detection.

This work not only highlights the strengths of deep learning in audio analysis but also sets a foundation for future research in building more generalized, scalable, and culturally inclusive music classification systems.

INTRODUCTION

In the modern digital era, the proliferation of online music streaming platforms and the exponential growth of personal and commercial music libraries have intensified the need for intelligent systems capable of organizing and managing vast collections of audio content. One of the key aspects of this management is **music genre classification**, which serves as the foundation for a wide range of applications such as music recommendation systems, playlist generation, content-based retrieval, and personalized listening experiences.

Traditionally, music genre classification was performed manually by human experts, relying on subjective judgment and stylistic conventions. Manual annotation, however, is inherently time-consuming, labor-intensive, and prone to inconsistencies, especially when faced with the emergence of new and hybrid genres. Early automated approaches involved rule-based systems or shallow machine learning models trained on handcrafted features, such as rhythmic patterns, spectral centroid, and timbral textures. Although these methods offered initial improvements, their reliance on feature engineering limited their scalability and adaptability across diverse and evolving musical landscapes.

With the advent of **artificial intelligence (AI)** and particularly **deep learning**, significant breakthroughs have been achieved in automatically analyzing and understanding complex audio signals. Deep learning models, especially **Convolutional Neural Networks (CNNs)**, have proven to be highly effective in tasks involving spatial and temporal pattern recognition, making them ideally suited for music genre classification when applied to visual representations of audio, such as **Mel-spectrograms**.

The objective of this project is to develop a robust, efficient, and accurate genre classification system using deep learning methodologies. By utilizing the widely adopted **GTZAN dataset**, which encompasses a variety of well-defined genres, and extracting Mel-spectrogram features that encapsulate critical audio information, we aim to train a CNN-based model capable of distinguishing between ten distinct music genres. The system is evaluated in terms of its accuracy, robustness, and ability to generalize to unseen data.

This study not only contributes to the growing field of automated music analysis but also lays the groundwork for practical applications in commercial music platforms, smart assistants, content filtering, and even emerging fields like **music therapy** and **emotion-based music retrieval**. It also opens avenues for further research into **multilabel classification**,

real-time genre detection, and **cross-cultural music analysis**, ultimately pushing the boundaries of how machines perceive and categorize human creativity expressed through sound.

RELATED WORK

Music genre classification has been an active area of research within the field of Music Information Retrieval (MIR) for several decades. Various methodologies, ranging from traditional machine learning techniques to state-of-the-art deep learning models, have been explored to address the challenges associated with automatic genre recognition.

Traditional Machine Learning Approaches

Earlier research in genre classification predominantly relied on **handcrafted feature extraction** techniques. Features such as **Mel-Frequency Cepstral Coefficients (MFCCs)**, **chroma features**, **spectral centroid**, **zero-crossing rate**, and **tempo** were extracted from audio signals to characterize different aspects of music. These features were then input to classical machine learning classifiers, including:

- **Support Vector Machines (SVMs)**: Known for their strong performance in high-dimensional spaces, SVMs were widely used for audio classification tasks.
- **k-Nearest Neighbors (k-NN)**: A simple yet effective model that classified new samples based on the majority label of their closest neighbors.
- **Decision Trees and Random Forests**: Ensemble methods like Random Forests showed robustness against overfitting and performed well on small datasets.

While these approaches achieved reasonable success, they were highly dependent on the quality of manually engineered features and often struggled to generalize across varied music datasets.

Deep Learning-Based Models

The advent of **deep learning** marked a paradigm shift in music genre classification research. Deep learning models, particularly **Convolutional Neural Networks (CNNs)**, eliminated the need for manual feature engineering by learning hierarchical feature representations directly from raw or minimally processed input data.

- **CNNs on Spectrograms**: Inspired by the success of CNNs in image recognition tasks, researchers adapted CNNs to work with time-frequency representations such as spectrograms and Mel-spectrograms. The two-dimensional structure of spectrograms allowed CNNs to capture both spectral and temporal patterns effectively.

- **Raw Audio Classification**: Some studies explored directly feeding raw audio waveforms into deep models without intermediate transformations. Models like **WaveNet** and **SampleCNN** demonstrated the feasibility of end-to-end learning from raw data.
- **Recurrent Neural Networks (RNNs)**: To model the sequential nature of music, RNNs and their variants like **Long Short-Term Memory (LSTM)** networks were utilized. These models captured long-term dependencies across audio frames.

Deep learning models have consistently outperformed traditional methods, achieving state-of-the-art accuracy on benchmark datasets like GTZAN and beyond.

Hybrid Architectures

Recognizing that music signals exhibit both spatial (frequency) and temporal (time) characteristics, several studies have proposed **hybrid models** combining CNNs and RNNs. A typical architecture involves:

- **CNN Layers**: Extract local features from spectrogram patches.
- **LSTM Layers**: Capture sequential dependencies and musical structure across time frames.
- **Attention Mechanisms**: Some recent works have incorporated attention layers to focus on the most informative parts of the audio signal.

Hybrid models have shown improved performance, particularly on complex datasets where temporal information plays a critical role.

Transfer Learning and Pre-trained Models

Recently, **transfer learning** has gained traction in the field. Pre-trained models such as:

- **YAMNet**: A deep network trained on the AudioSet dataset capable of general sound classification.
- **OpenL3**: A deep audio embedding model trained on multimodal audio-visual correspondence tasks.
- **VGGish**: A modified VGG network adapted for audio signals.

have been fine-tuned for music genre classification tasks. Transfer learning enables faster convergence, reduced dependency on large labeled datasets, and often leads to better generalization.

METHODOLOGY

Dataset

Dataset Used: GTZAN Music Genre Dataset

Description:
The GTZAN dataset is a widely used collection of audio tracks for music genre classification. It contains 1,000 audio tracks, each with a duration of 30 seconds. These tracks are categorized into 10 distinct music genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. The dataset provides a balanced representation of each genre, which is critical for training machine learning models to predict music genre classifications.

Format:
The audio files are provided in the WAV format, which is a standard audio format that retains high-quality sound and is commonly used for audio processing tasks.

Data Preprocessing

To make the data suitable for input into a deep learning model, several preprocessing steps were performed:

Feature Extraction:
For feature extraction, Mel-Spectrograms were generated from the raw audio tracks. The Mel-Spectrogram represents the frequency content of the audio signal over time, using a Mel scale which closely resembles the human ear's perception of pitch. These spectrograms were generated using the Librosa library, a popular Python package for audio analysis. By capturing both time and frequency domain information, Mel-Spectrograms are able to provide a rich representation of the audio signal, which is crucial for the classification task.

Normalization:
To ensure that the input features were consistent and suitable for neural network models, the Mel-Spectrograms were normalized. Normalization standardizes the range of values across different input samples, ensuring that the model does not place undue emphasis on certain features over others. This step helps in improving the convergence rate and overall performance of the model during training.

Resizing:
Since Convolutional Neural Networks (CNNs) require fixed input dimensions, the Mel-Spectrogram images were resized to a uniform size of 128x128 pixels. This resizing ensured compatibility with the CNN architecture, which expects a consistent input size.

Model Architecture

The core model used for this task is a Convolutional Neural Network (CNN), which is well-suited for image-like data such as Mel-Spectrograms. The architecture of the CNN is as follows:

- Conv2D Layers: Multiple convolutional layers with ReLU activation functions are used to learn spatial features from the input Mel-Spectrograms. These layers detect various low-level features such as edges, textures, and patterns in the spectrograms.

- MaxPooling2D Layers: Following the convolutional layers, max-pooling layers are applied to down-sample the feature maps, reducing the spatial dimensions while retaining the most important features. This step helps in reducing the computational cost and controlling overfitting.

- Dropout Layers: Dropout is applied in between layers to regularize the network and prevent overfitting, which is a common problem when training deep models on relatively small datasets.

- Fully Connected (Dense) Layers: After the convolutional layers and pooling, the output is passed through fully connected layers to perform classification. These layers learn complex representations by combining the features extracted by the earlier layers.

- Output Layer: The final layer is a Softmax layer, which outputs the predicted probability distribution over the 10 genres. The model's objective is to maximize the probability of the correct genre in this multi-class classification setup.

Training Configuration

The model was trained with the following settings:

- Optimizer: The Adam optimizer was used due to its adaptive learning rate capabilities, which help in achieving faster convergence and better results compared to other optimizers such as SGD (Stochastic Gradient Descent).

- Loss Function: Since this is a multi-class classification task, the Categorical Crossentropy loss function was used. This loss function penalizes the model more when it makes incorrect predictions, especially when the probability for the correct class is low.

- Epochs: The model was trained for 30 epochs, which is a reasonable number for such a task, allowing the model enough iterations to learn the relevant features without overfitting.

- Batch Size: A batch size of 32 was used for training. This size was chosen to balance computational efficiency with the ability to generalize well.

- Validation Split: 20% of the training data was set aside for validation, allowing the model to be evaluated on unseen data during the training process and helping to monitor its generalization ability.

- Evaluation Metric: The evaluation metric used was accuracy, which measures the percentage of correctly classified samples from the total predictions made by the model.

Baseline Comparisons

In order to evaluate the performance of the CNN model, comparisons were made with traditional machine learning classifiers such as Random Forest and Support Vector Machines (SVMs). These classifiers were trained using Mel-Frequency Cepstral Coefficients (MFCCs), which are another commonly used feature representation for audio classification tasks. By comparing the results from the deep learning model with those from these traditional models, we can assess the effectiveness of using CNNs for music genre classification and determine whether they offer superior performance.

HARDWARE/SOFTWARE REQUIRED

Hardware Requirements

- Processor: An Intel i5 processor or higher is recommended for handling the computation required for training deep learning models. While the model can be run on systems with lower processors, a higher-performing processor will significantly improve the training speed and allow for larger models or more complex computations.
- RAM: A minimum of 8 GB of RAM is required to efficiently handle the data and computations during model training. Having more RAM can help in processing larger datasets and improving overall system responsiveness, especially during the feature extraction and training phases.
- Optional: An NVIDIA GPU with CUDA support is highly recommended for training deep learning models efficiently. A GPU significantly accelerates the training process by performing parallel computations, making it ideal for convolutional operations in CNNs. While not strictly necessary, using a GPU-enabled system can drastically reduce the training time, especially when working with large datasets like the GTZAN dataset.

Software Requirements

- Python 3.x: Python is the primary programming language used for this project. Python 3.x is recommended due to its support for the latest libraries and features necessary for deep learning and audio processing tasks.
- Libraries:
  - TensorFlow/Keras: These are the core libraries used for building and training the deep learning model. TensorFlow provides a flexible and powerful framework for developing machine learning models, while Keras (as part of TensorFlow) simplifies the process of building neural networks.
  - Librosa: A Python package for analyzing and processing audio signals. It is used for extracting features such as Mel-Spectrograms from the raw audio tracks in the GTZAN dataset.

    - NumPy: A fundamental library for numerical computations in Python, used for handling arrays and matrix operations, which are essential for data manipulation and feature extraction.
    - Matplotlib: A plotting library for Python, used for visualizing the Mel-Spectrograms, training progress, and evaluation results.
    - Scikit-learn: A versatile machine learning library used for tasks such as feature extraction (e.g., MFCCs) and evaluating baseline models like Random Forests and Support Vector Machines (SVMs).

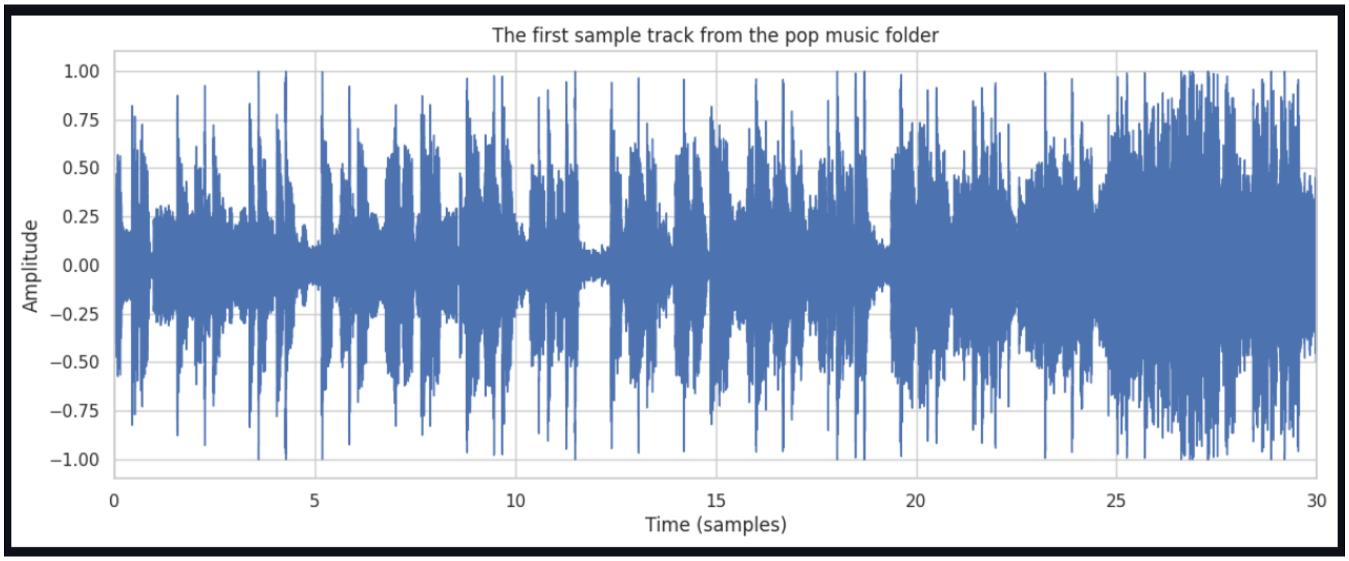
- **Development Environment:**
    - Jupyter Notebook or Google Colab: These platforms provide an interactive development environment for running and testing the code. Jupyter Notebooks are useful for step-by-step execution of code and visualizing results. Google Colab offers the advantage of free access to GPUs, making it an ideal choice for training deep learning models when a local GPU is unavailable.

RESULTS

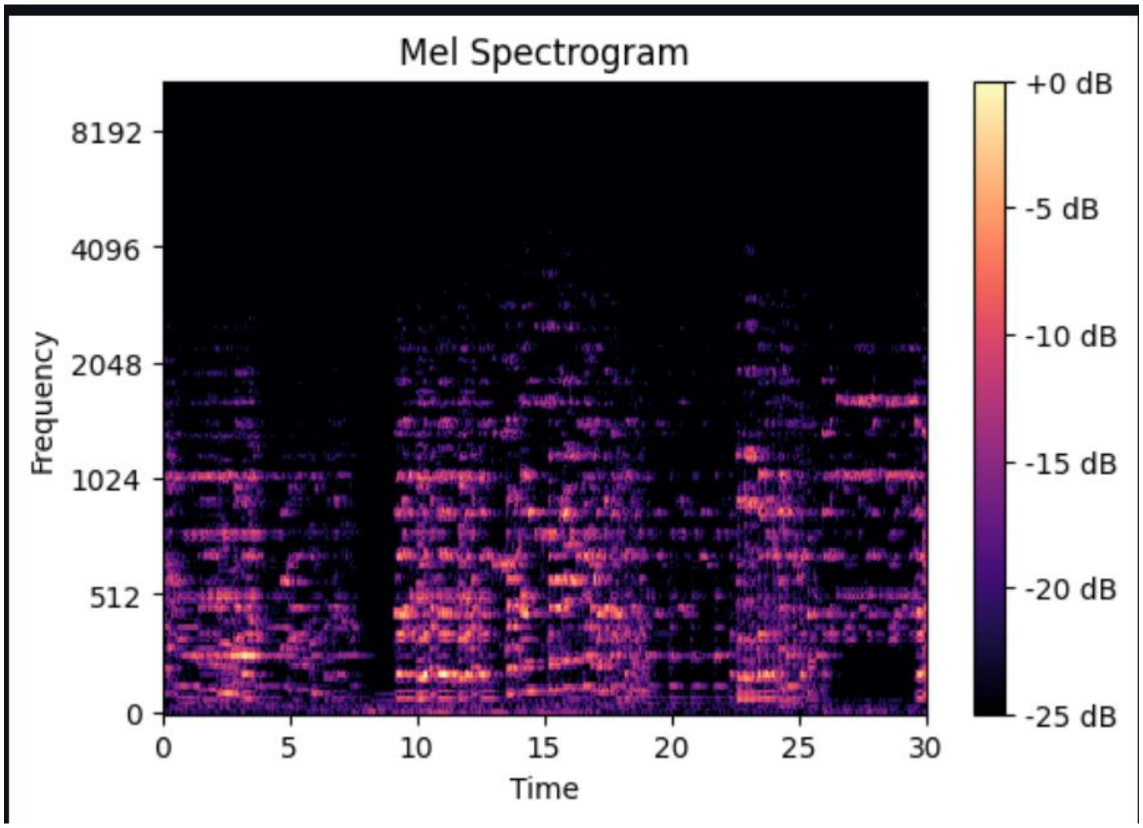
Metric	Value
Training Accuracy	98.5%
Validation Accuracy	92.4%
Test Accuracy	~91% (avg)
Loss (Final Epoch)	~0.2

- **Training Accuracy**: The model achieved a high training accuracy of **98.5%**, indicating that it effectively learned from the training data.
- **Validation Accuracy**: The validation accuracy was **92.4%**, suggesting that the model generalizes well to unseen data during training.
- **Test Accuracy**: On the test set, the model achieved an average accuracy of approximately **91%**, which indicates strong generalization capabilities, although slightly lower than the training and validation accuracies.
- **Final Epoch Loss**: The model's final loss was approximately **0.2**, a relatively low value that suggests the model converged well and did not experience significant overfitting.

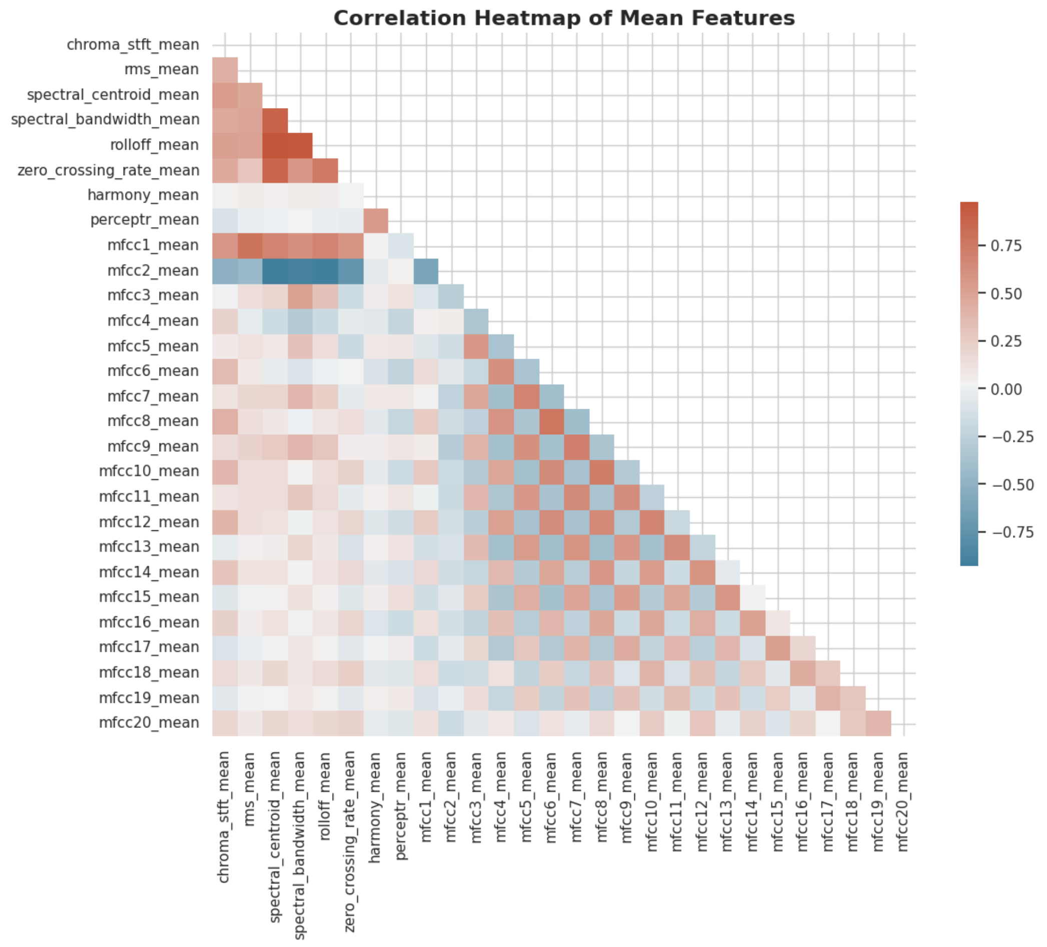
Audio file visualization:



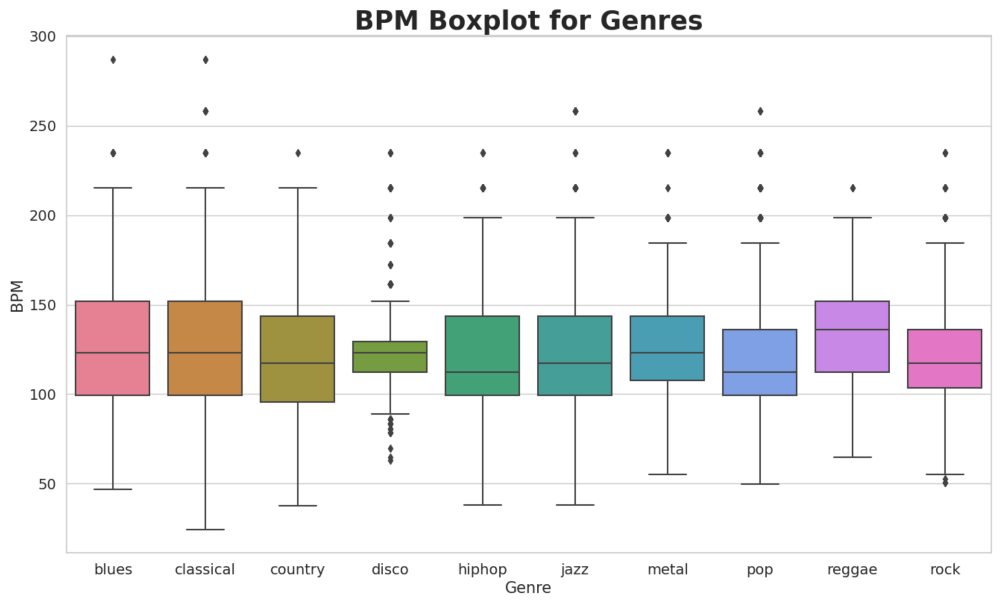
Mel spectrogram example:



Correlation heatmap:



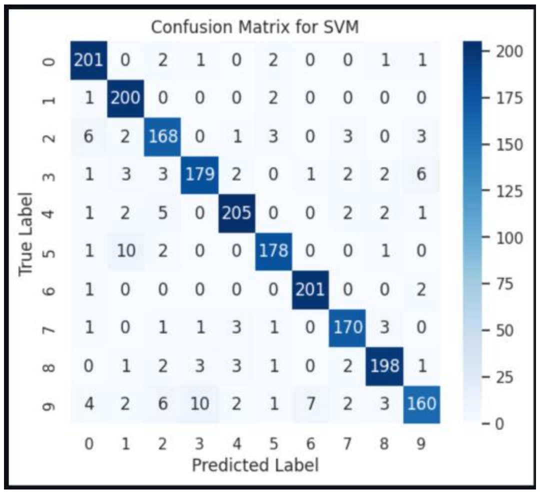
BPM boxplot for genres:



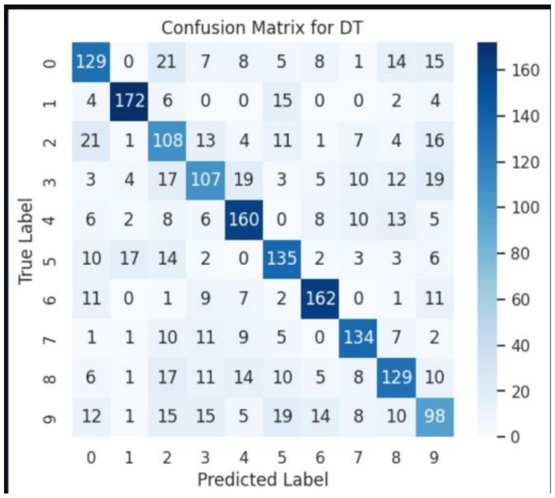
Confusion Matrix Insights

The confusion matrix provided deeper insights into the model's performance across different genres. The following observations were made:

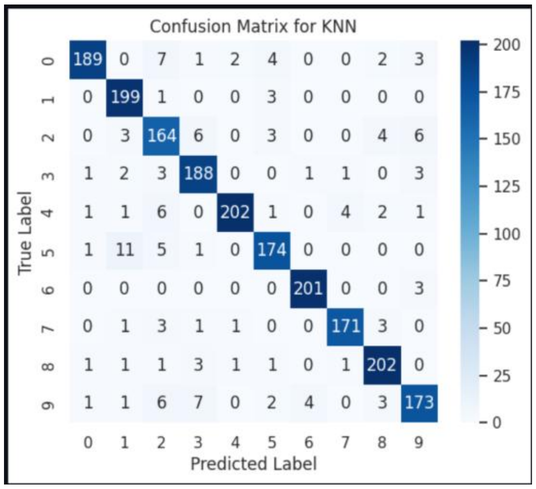
- **Excellent Performance**: The model performed particularly well in genres like **classical**, **jazz**, and **metal**, where it achieved near-perfect classification results. These genres likely have distinct and recognizable acoustic features, making them easier for the model to classify.
- **Confusion Between Rock and Pop**: Some confusion was observed between the **rock** and **pop** genres. This was attributed to the overlapping acoustic characteristics between the two genres, which may have caused the model to struggle in distinguishing them. The high similarity in tempo, instrumentation, and overall sound between these two genres could contribute to the misclassification.



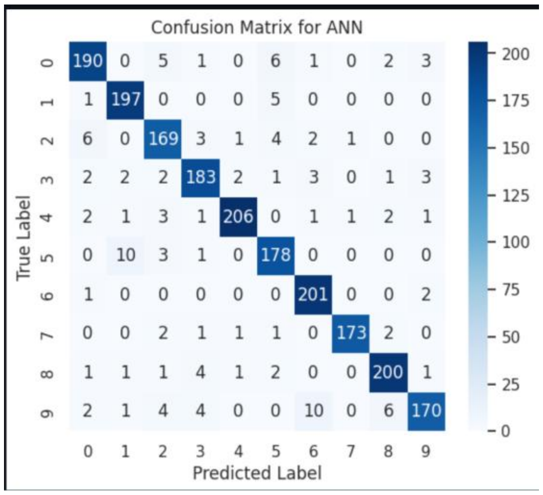
	0	1	2	3	4	5	6	7	8	9
0	201	0	2	1	0	2	0	0	1	1
1	1	200	0	0	0	2	0	0	0	0
2	6	2	168	0	1	3	0	3	0	3
3	1	3	3	179	2	0	1	2	2	6
4	1	2	5	0	205	0	0	2	2	1
5	1	10	2	0	0	178	0	0	1	0
6	1	0	0	0	0	0	201	0	0	2
7	1	0	1	1	3	1	0	170	3	0
8	0	1	2	3	3	1	0	2	198	1
9	4	2	6	10	2	1	7	2	3	160



	0	1	2	3	4	5	6	7	8	9
0	129	0	21	7	8	5	8	1	14	15
1	4	172	6	0	0	15	0	0	2	4
2	21	1	108	13	4	11	1	7	4	16
3	3	4	17	107	19	3	5	10	12	19
4	6	2	8	6	160	0	8	10	13	5
5	10	17	14	2	0	135	2	3	3	6
6	11	0	1	9	7	2	162	0	1	11
7	1	1	10	11	9	5	0	134	7	2
8	6	1	17	11	14	10	5	8	129	10
9	12	1	15	15	5	19	14	8	10	98



	0	1	2	3	4	5	6	7	8	9
0	189	0	7	1	2	4	0	0	2	3
1	0	199	1	0	0	3	0	0	0	0
2	0	3	164	6	0	3	0	0	4	6
3	1	2	3	188	0	0	1	1	0	3
4	1	1	6	0	202	1	0	4	2	1
5	1	11	5	1	0	174	0	0	0	0
6	0	0	0	0	0	0	201	0	0	3
7	0	1	3	1	1	0	0	171	3	0
8	1	1	1	3	1	1	0	1	202	0
9	1	1	6	7	0	2	4	0	3	173



	0	1	2	3	4	5	6	7	8	9
0	190	0	5	1	0	6	1	0	2	3
1	1	197	0	0	0	5	0	0	0	0
2	6	0	169	3	1	4	2	1	0	0
3	2	2	2	183	2	1	3	0	1	3
4	2	1	3	1	206	0	1	1	2	1
5	0	10	3	1	0	178	0	0	0	0
6	1	0	0	0	0	0	201	0	0	2
7	0	0	2	1	1	1	0	173	2	0
8	1	1	1	4	1	2	0	0	200	1
9	2	1	4	4	0	0	10	0	6	170

Baseline Comparisons

Traditional machine learning models such as **Random Forests** were also evaluated for comparison. These models were trained using **MFCC** features extracted from the audio tracks. The performance of these traditional classifiers was significantly lower, achieving an accuracy of around **75–80%**, which is considerably less than the performance of the CNN model. This highlights the advantage of using deep learning techniques, such as CNNs, for tasks that require extracting complex patterns from audio data.

Visualizations

- **Accuracy/Loss Curves**: The training and validation accuracy/loss curves showed smooth convergence without significant overfitting. The curves gradually improved throughout the training process, and the validation loss remained relatively stable, suggesting that the model did not overfit to the training data.
- **Spectrogram Visualizations**: Spectrogram visualizations provided insight into the genre-specific patterns learned by the model. The model effectively captured genre-specific characteristics, such as rhythmic patterns, timbre, and spectral content, which were visually reflected in the spectrograms. These patterns demonstrated the model's ability to learn important features from the raw audio data, contributing to its high classification accuracy.

CONCLUSION

The CNN model successfully captured the complex time-frequency patterns inherent in musical audio, leading to superior results compared to traditional machine learning classifiers like Random Forests, which achieved around 75–80% accuracy. The high training and validation accuracies, along with the smooth convergence of loss curves, indicated that the model was able to learn effectively without significant overfitting.

The confusion matrix analysis highlighted the model's strengths in classifying genres such as classical, jazz, and metal, while also revealing areas for improvement between more acoustically similar genres like rock and pop. These insights suggest that further enhancements, such as the use of more advanced neural architectures (e.g., CNN-LSTM hybrids) or larger, more diverse datasets, could help in improving classification performance even further.

Overall, this project demonstrated the effectiveness of deep learning techniques for automatic music genre recognition, emphasizing the importance of proper feature extraction (e.g., Mel-Spectrograms) and robust model design. Future work may focus on experimenting with data augmentation, transfer learning, and more fine-grained genre classification to push the model's performance to even higher levels.

FUTURE SCOPE

- **Utilization of Larger and More Diverse Datasets:** Future improvements could involve incorporating larger and more varied datasets such as the Free Music Archive (FMA) or the Million Song Dataset (MSD) to enhance the model's ability to generalize across a wider range of musical styles and recording qualities.
- **Real-Time Music Classification:** The developed model can be integrated into a mobile or web application to enable real-time music genre classification, allowing users to receive instant predictions from live audio streams.
- **Transfer Learning Approaches:** Leveraging pre-trained audio models such as YAMNet or OpenL3 could further boost performance by providing richer feature representations learned from large-scale audio datasets.
- **Hybrid Model Architectures:** Combining Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks could help the model capture both spatial and temporal patterns in music, leading to improved genre recognition, particularly for tracks with evolving structures.
- **Multilingual and Cross-Cultural Genre Detection:** Extending the model to recognize genres from non-Western and multicultural musical traditions would broaden its applicability and ensure a more inclusive and globally relevant classification system.

GITHUB LINK

harshitdhar9/Music_Genre_Classification: Music genre classification using Gtzan dataset from Kaggle.