Music Genre Classification System

By:
Julissa Mijares and Selina Manua

Table of Contents

Introduction

Classifying music by genre is a fundamental task in music information retrieval, with applications in recommendation systems, automated library organization, and audio research. The rapid growth of digital music libraries and streaming platforms has made efficient genre classification an essential component for enhancing user experience and enabling personalized discovery. While traditional methods have relied on manual feature extraction and statistical models, these approaches often fail to capture the nuanced and overlapping characteristics present in musical styles. As a result, more sophisticated methods are required to achieve accurate classification.

Recent advances in deep learning have enabled automatic feature extraction from raw data, leading to significant improvements in tasks such as image recognition, natural language processing, and audio analysis. Convolutional Neural Networks (CNNs) have demonstrated strong performance in processing 2D data, such as images, and can be adapted to analyze Mel spectrograms, a 2D time-frequency representation of audio signals. Mel spectrograms allow CNNs to extract both temporal and frequency-based features directly from the data, enabling a richer and more representative feature space for genre classification.

This paper demonstrates that a Convolutional Neural Network can classify music into distinct genres using Mel spectrograms as input. The model is trained to identify 10 distinct genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. As modern music often blends elements from multiple genres, there is a growing need for multi-genre classification. To address this challenge, the model is extended to classify tracks that reflect these genre overlaps, better capturing the complexity of real-world music classification.

This study aims to build CNN models that classify both single-genre and multi-genre songs, addressing the evolving nature of modern music. The models' performance is evaluated on both tasks. The results demonstrate the potential of deep learning to enhance genre classification and support personalized music discovery.

Data Collection

Our study utilizes the GTZAN dataset[1], a benchmark for music genre classification containing ten balanced genres (blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock), with 100 tracks per genre, each lasting 30 seconds. This dataset provides a foundation for training and evaluating our model.

[1] https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

For our multi-genre classification, we selected 50 songs that each contained two genres, ensuring that all 10 genres were equally represented across the dataset. We downloaded the MP3 versions of these tracks and then extracted a 30-second segment from each song to maintain consistency in the audio sample length. Finally, we converted these selected segments from MP3 to WAV format for use in the model. This dataset was used to calculate the accuracy of our multi-genre classification model.

Data Processing

We processed the audio data by converting each audio file into a corresponding Mel spectrogram image and organizing the output into genre-specific directories. For each audio file with a *.wav* extension in a genre-specific input folder, the script uses Librosa to load the audio file and compute a Mel spectrogram with 128 frequency bins, a maximum frequency of 8000 Hz, and normalized amplitude values between 0 and 1. The spectrogram is converted to a decibel scale for better visualization. Using Matplotlib, the script saves the spectrogram as a *.png* image in the corresponding genre output folder, ensuring that the plot has no axes, color bars, or titles for a clean look. The images are resized to a uniform dimension of 128x128 pixels and converted to grayscale to standardize the input for a machine-learning model. Each image is then converted into a NumPy array for numerical processing. If an error occurs during processing, the script logs the issue and skips the problematic file. Each successfully processed file is logged with its output path for tracking. Our dataset only had one issue with the corrupted file *jazz.00054*, so that file was skipped.

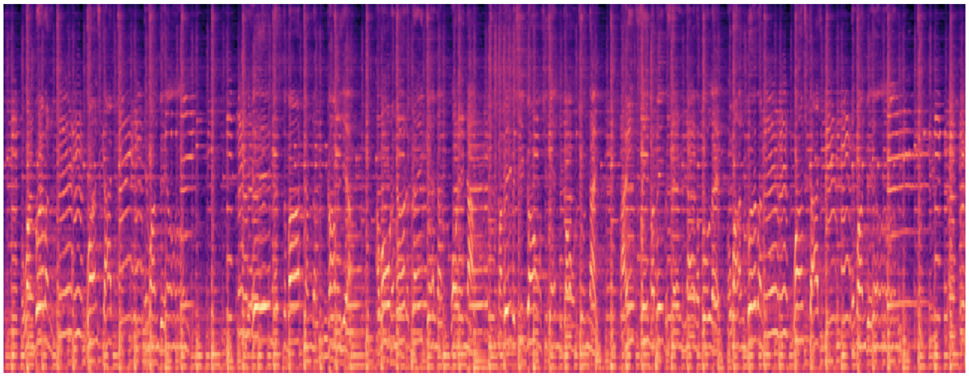
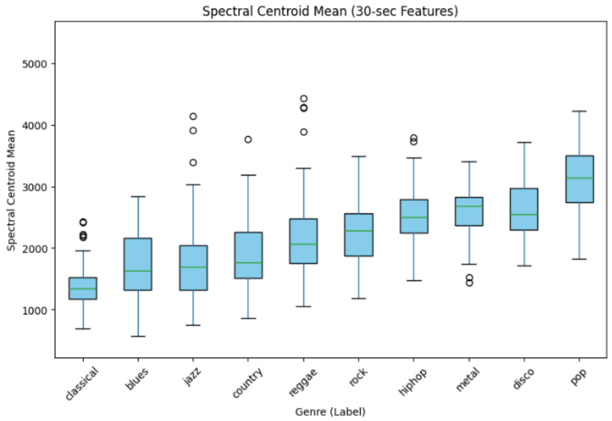


Figure 1: Mel spectrogram of the audio file "*blues.00000.wav*"

Preliminary Analysis



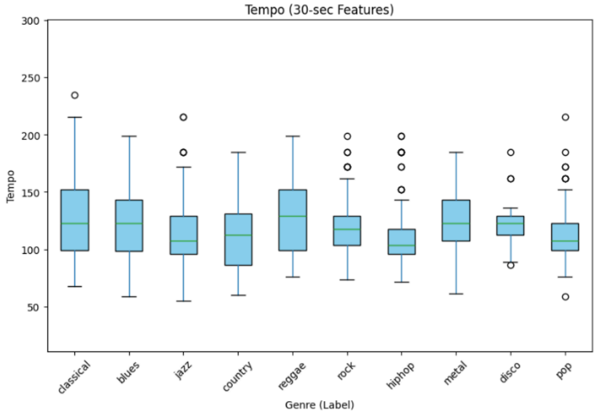


Figure 2: Boxplot of the Spectral Centroid Mean by Genre

Figure 3: Boxplot of Tempo by Genre

In Figure 2, the spectral centroid mean varies significantly between genres, with pop exhibiting the highest values, indicating a stronger presence of high-frequency content. In contrast, classical and blues genres show lower spectral centroids, reflecting their emphasis on lower-frequency sounds. Figure 3 highlights the tempo distribution, where classical tends to have higher tempos, while jazz generally displays slower tempos, reflecting their characteristic musical structures. These observations reveal genre-specific traits in both tempo and rhythm, which are crucial for distinguishing between genres in music classification models.

Methodology

After data preprocessing, the dataset was split into 70% training, 10% validation, and 20% testing, with genre labels one-hot encoded into 10-dimensional categorical vectors. Data augmentation techniques were applied to the training set using TensorFlow's ImageDataGenerator to enhance generalization.

Base Model

The custom CNN was designed in TensorFlow with three convolutional layers, each followed by max-pooling to reduce spatial dimensions and dropout layers to mitigate overfitting. The starting model layers included a fully connected layer with 128 units and a softmax output layer for multi-class classification. The Adam optimizer was used with a learning rate of 0.001 and a categorical cross-entropy loss function. The model was trained for 50 epochs with a batch size of 32. Early stopping was implemented with a patience of 10 epochs to prevent overfitting, and a ReduceLROnPlateau callback adjusted the learning rate by a factor of 0.1 when validation loss plateaued.

Additional Models

To enhance model performance, we explored other CNN architectures.

Hyperparameter Tuning

Hyperparameter optimization was performed using GridSearch, varying key parameters such as the number of filters, kernel sizes, dropout rates, and learning rates. 551 trials were conducted, and the best-performing model was selected based on validation accuracy. Initially, Random Search was considered as an alternative because it selects a random subset of possible combinations, offering a computationally efficient approach. However, since Random Search did not improve the base model's performance, GridSearch was chosen instead. GridSearch exhaustively evaluates all possible combinations of hyperparameters, ensuring a thorough search for the optimal configuration. Despite the higher computational cost of training deep learning models, GridSearch was preferred as it ultimately provided better results with the more exhaustive exploration of hyperparameter space.

Transfer Learning: VGG16

Additionally, transfer learning was explored by fine-tuning the VGG16 architecture. Since VGG16 is a visual model trained on RGB images, the inputs were adapted by converting grayscale spectrograms into 3-channel RGB images by repeating the single grayscale channel three times. All convolutional layers in the network were frozen to retain the original learned weights, ensuring that only the newly added layers were trained. A fully connected layer with 128 units, a dropout rate of 0.5, and a softmax output layer were then added to classify the spectrograms into 10 genres.

Multi-Genre Classification

For our multi-genre classification, we modified the VGG-16 model by changing the softmax function in the output layer to a sigmoid function. The sigmoid function was applied to each genre individually, transforming each raw output logit into an independent probability between 0 and 1. This allows the model to treat the presence of each genre as a separate binary classification problem, making it possible to assign multiple genres to a single song. To determine which genres are predicted for a given song, we applied a threshold to the sigmoid output. Genres with a probability higher than 0.2 were considered part of the song's genre set. We chose this threshold based on experimentation, where higher thresholds often led to the model predicting only a single genre, even for songs that fit multiple genres. The 0.2 threshold allowed for more flexible predictions, enabling the model to classify songs into multiple genres while avoiding the issue of overly conservative predictions.

Results

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Base Model	89.54%	68.00%	58.50%
Tuned Model	97.40%	70.00%	62.00%
VGG-16	70.96%	62.00%	66.50%
Multi-Genre Classification with VGG-16 Model	-	-	40.00%

Performance was assessed using test and validation accuracy as well as precision. Confusion matrices and classification reports provided further insight into the model's ability to classify genres accurately.

The hyperparameter-tuned model achieved a validation accuracy of 70.00% and a test accuracy of 62.00%, improving from the base model. The confusion matrix in Figure 4 shows the performance of the model across the ten genres, with some genres being classified more accurately than others. Metal demonstrates high precision (0.95) and high recall (0.83), reflecting the distinctive features that make it easier to classify. Classical has perfect recall but moderate precision (0.71), which suggests that the model accurately identifies all classical tracks but occasionally misclassifies other genres as classical. Rock shows the lowest precision (0.39) and low recall (0.45), highlighting the model's difficulty distinguishing it from other genres, possibly due to the diversity within rock subgenres or overlapping musical elements.

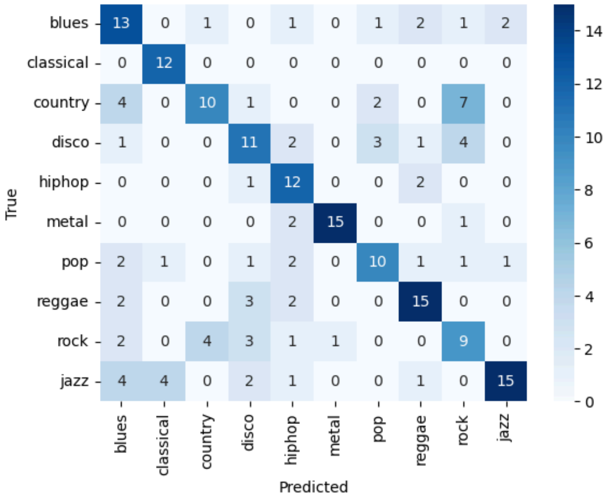


Figure 4: Confusion matrix for tuned model

In Figure 5, the graph on the left shows a large gap between training and validation accuracy after 20 epochs, suggesting that the model may have overfitted to the training data. This is also consistent with the graph on the right, where the training loss steadily decreases, indicating effective learning, while the validation loss plateaus and increases, suggesting overfitting.

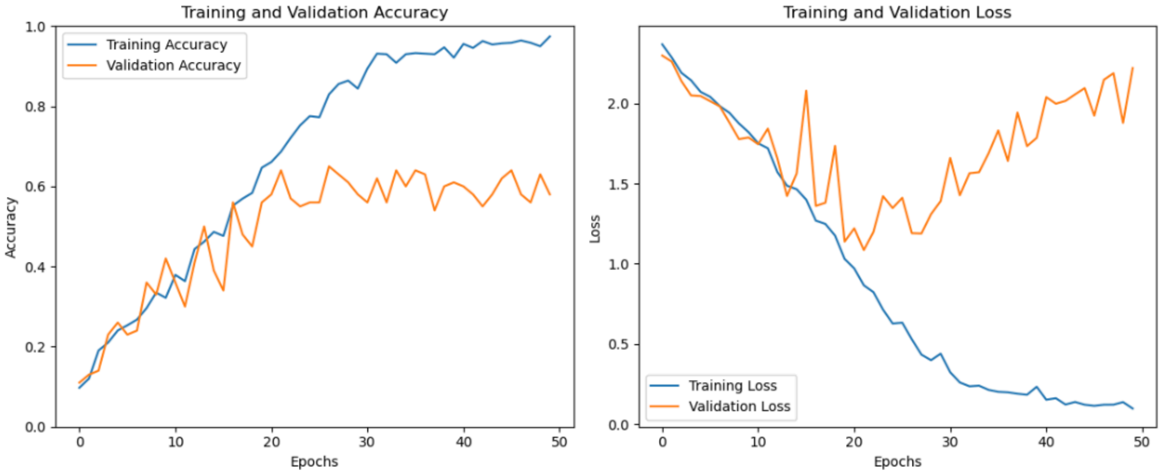


Figure 5: Graph showing training and validation accuracy as epochs increase (left) and a graph showing training and validation loss as epochs increase (right) for the hyperparameter-tuned model

The VGG-16 model achieved a validation accuracy of 62.00% and a test accuracy of 66.50%, performing the best on the test set out of all other architectures. In Figure 6, the graph on the left shows that the gap between training and validation accuracy remains relatively small throughout the training process, indicating better generalization compared to the tuned model. In the loss graph on the right, the training loss decreases consistently, but the validation loss flattens toward the end, indicating the model is still overfitting.

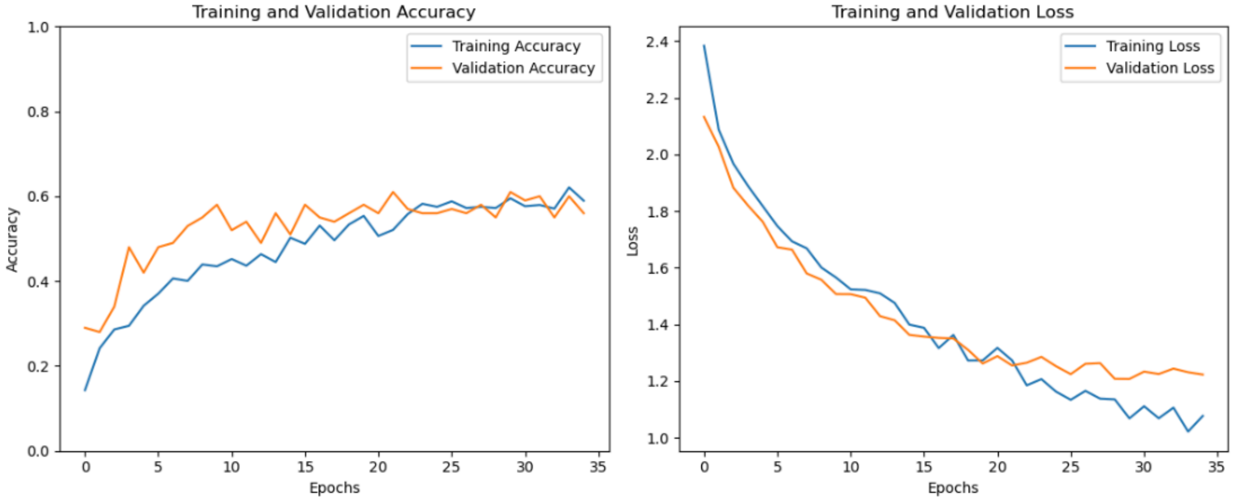


Figure 6: Graph showing training and validation accuracy as epochs increase (left) and a graph showing training and validation loss as epochs increase (right) for the VGG-16 model

The Multi-Genre Classification model, built on the VGG-16 architecture as it is the best model for single-genre classification, achieved a test accuracy of 40%. This significant drop in performance highlights the inherent challenges of multi-genre classification. The complexity of music genres, due to overlapping styles, tempos, and other musical characteristics, makes accurate genre classification more difficult. Additionally, the diversity within each genre further complicates the task, as songs that fall under the same genre can exhibit vastly different musical traits. As a result, the multi-genre classification model struggles to effectively capture all relevant features, leading to a lower overall accuracy.

Conclusion

This paper demonstrated how music genre classification can be performed using Convolutional Neural Networks (CNNs) with images of Mel spectrograms as an input. While the model was able to more accurately classify single genres, it faced challenges with multi-genre classification due to overlapping musical styles, tempos, and other genre-specific features. The results emphasize the potential of deep learning for genre classification but also highlight the complexities of accurately identifying multiple genres in a single track.

Future Steps and Limitations

In the future, one potential direction is to explore larger and more diverse datasets that reflect the complexities of modern music. While the GTZAN dataset serves as a useful benchmark, it contains only 100 tracks per genre, which limits the model's ability to generalize well to larger, more varied datasets. As streaming platforms and digital libraries continue to expand, datasets that better capture the wide range of musical styles, sub-genres, and cross-genre influences will be valuable.

Another potential project is to incorporate additional features about the songs to provide more context for the model, such as artist, album, or song lyrics. For instance, certain artists or albums might be strongly associated with specific genres or genre-blending, and this information could provide valuable context. Adding metadata could potentially lead to more accurate predictions, especially in cases where a song may span multiple genres or where the genre is ambiguous.

Despite these potential directions for future improvement, several limitations should be acknowledged. The GTZAN dataset is relatively small and lacks diversity in both sample size and content. With only 100 tracks with 10 genres, the dataset does not fully capture the variety and complexity of real-world music collections. Additionally, the use of Mel spectrograms may not capture all the dynamic elements of music. Music is inherently temporal and dynamic, and Mel spectrograms provide a time-frequency representation that may overlook important rhythmic, melodic, and harmonic shifts. For example, changes in tempo, key, or rhythm might be more significant for genre classification but are not always fully captured in the static representation of a spectrogram. Future work could explore alternative feature extraction techniques that focus on capturing the evolving nature of music or integrate dynamic features with Mel spectrograms to improve classification accuracy.