Music Genre Classification

Leonor Gonçalves Gouveia (V12847) – Audio Pattern Recognition

***Abstract* — This study presents a comparative analysis of traditional Machine Learning and Deep Learning approaches for music genre classification on the GTZAN dataset. Two pipelines were constructed, consisting of a feature-based approach with a Random Forest and a Support Vector Machine (SVM), and an end-to-end approach using a Convolutional Neural Network (CNN) trained on Log-Mel spectrograms. The CNN achieved the highest test accuracy, establishing the best predictive benchmark. Permutation Importance demonstrated which features genre differentiation relies primarily on.**

I. Introduction

Music Genre Classification (MGC) aims to automatically label a song with its corresponding genre (e.g., Rock, Jazz, Classical). The identification of musical genres is fundamental in the digital era and has multiple applications.

The primary importance of MGC is in its role in digital music recommendation and categorization. Streaming platforms rely on genre tags to power collaborative filtering, personalize user playlists, and manage music libraries, thereby impacting user engagement and discoverability. Beyond commercial applications, MGC also contributes to music information retrieval research, musicology, and intellectual property management.

Early work in MGC relied on extracting acoustic features, such as Mel-Frequency Cepstral Coefficients (MFCCs) [1] and Perceptual Wavelet Packets (PWP), followed by traditional Machine Learning classifiers like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) [2]. More recently, the field has been innovated by the adoption of Deep Learning, where Convolutional Neural Networks (CNNs) are applied directly to two-dimensional representations of audio, such as Log-Mel spectrograms [3]. These approaches often achieve preeminent results by automatically learning genre-specific features, often exceeding traditional methods.

This project compares two distinct methodologies for music genre classification: a Traditional Machine Learning pipeline that relies on acoustic features, and a Deep Learning pipeline that employs a Convolutional Neural Network (CNN) to automatically learn features from Log-Mel Spectrograms, aiming for superior performance.

II. Methodology

A. Block Diagram of the Method

The diagram below illustrates the flow from raw data to the final classification and evaluation stages.

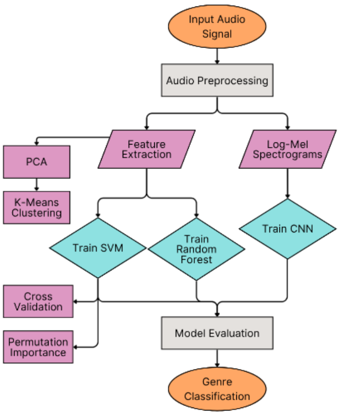

Fig. 1. Block Diagram of the Method

The project methodology was structured using a dual-pipeline approach, following the principles of the CRISP-DM framework [2] from data preparation through evaluation. The traditional Machine Learning pipeline first extracts a vector of features which are then standardized. This path includes exploratory analysis via PCA and K-Means clustering [2] and employs classifiers such as Random Forest and SVM, with the subsequent subjected to Permutation Importance [4] to quantify feature influence. In parallel, the Deep Learning pipeline focuses on end-to-end learning: raw audio is transformed into a Log-Mel Spectrogram, which feeds a 2D-CNN to automatically learn features. Both pipelines conclude by producing classification metrics, confusion matrices, and overall test accuracy for a balanced final assessment.

III. Data and Feature Engineering

A. Dataset: GTZAN

The project utilizes the GTZAN Genre Collection [5], a standard dataset for music genre classification.

TABLE I. GTZAN Dataset Description

Characteristic	**Value**
Total Samples	999 files
Classes (Genres)	10 (Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae, Rock)
Samples per Class	100 per genre (except Jazz: 99)
Audio Format	WAV files, mono, 22050 Hz sampling rate
Clip Duration	30 seconds

The data was split into an 80% training set and a 20% test set using stratified sampling to ensure the genre proportions were maintained in both subsets.

B. Feature Extraction

For the traditional pipeline, 62 features were extracted from each 30-second audio clip using the librosa [6] and pywt [7] libraries. The features were extracted frame-by-frame (with a frame size of N_{fft}=2048 and a hop length of 512 samples) and then summarized using the mean and standard deviation across all frames of the clip.

TABLE II. FEATURES EXTRACTED

Feature Domain	**Features Extracted**
Frequency-Domain	$MFCC_{1-13}$, Spectral Centroid, Chroma Vector
Entropy	Entropy of Energy, Spectral Entropy
Time-Domain	Zero-Crossing Rate (ZCR)
Wavelet-Domain	Perceptual Wavelet Packets (PWP)

C. Feature Normalization and Preprocessing

Prior to training the Machine Learning classifiers, the resulting 999×62 feature matrix was scaled using the StandardScaler. This technique transforms the data such that each feature has a mean of zero and a standard deviation of one, preventing features with large magnitudes from dominating the classification that rely on Euclidean distances.

D. Deep Learning Representation

For the Deep Learning pipeline, the audio signal was converted into a Log-Mel Spectrogram (128 Mel bands). This 128×1292 time-frequency representation serves as the 2D image input for the CNN. This bypasses feature engineering, letting the CNN automatically learn the most prominent features for genre separation.

IV. EXPLORATORY ANALYSIS AND FEATURE SPACE

A. Dimensionality Reduction and Visualization

To gain an initial understanding of the data's structure and separability, Principal Component Analysis (PCA) was applied to the feature space. The first two principal components (PC1 and PC2) captured only 45.00% of the total variance (PC1: 26.39%, PC2: 18.61%). This low explained variance indicates that the genre information is distributed across many features, and simple two-dimensional visualization is insufficient to reveal linear separation.

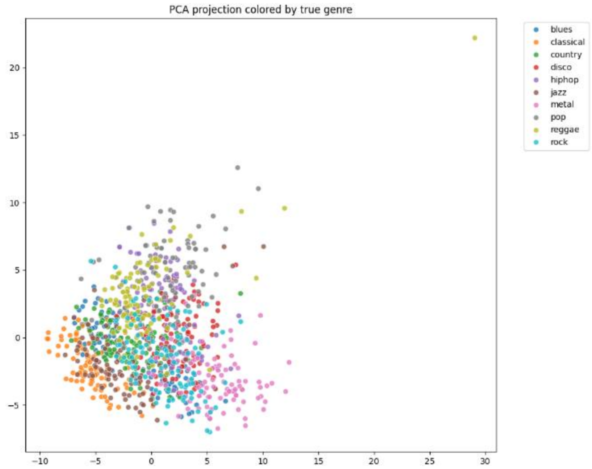

Fig. 2. PCA Projection Colored by True Genre

The PCA projection colored by true genre confirms this complexity. While some genres, notably Metal (pink) and Classical (orange), show slight concentrations on the periphery, most points belonging to genres like Pop, Rock, and Disco are highly overlapped.

B. Unsupervised Clustering (K-Means)

To formally test the compactness and separation of the feature space without relying on true labels, K-Means clustering was performed with k=10 (equal to the number of genres).

The Silhouette Score (0.1143) indicates poor cluster quality, as most data points are close to the boundaries between the groups, suggesting poor class separability. The Adjusted Rand Index (ARI) was low at 0.1928, and the Normalized Mutual Information (NMI) was 0.3354. These values collectively confirm that the unsupervised clustering failed to recover the true genre structure.

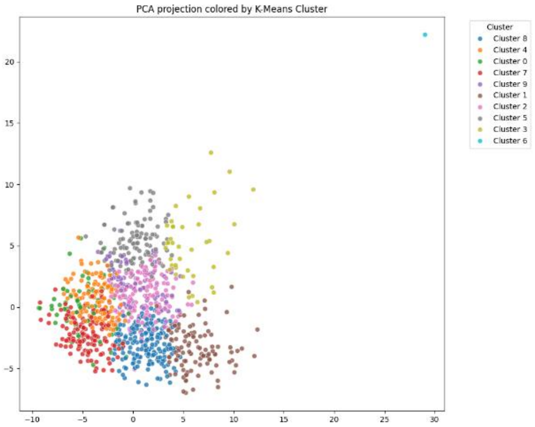

Fig. 3. PCA Projection Colored by K-Means Cluster

The PCA projection colored by K-Means cluster visually demonstrates the poor performance of K-Means. The resulting clusters are highly scattered containing mixtures of many different genres. This finding justifies the use of non-linear classification models like SVM and CNN in the subsequent steps.

V. EXPERIMENTAL SETUP AND RESULTS

This section details the parameters used for the classification experiments and presents the performance outcomes for the traditional machine learning and deep learning pipelines.

A. Traditional Machine Learning Results

Two classifiers were trained on the feature vectors:

1) Classifier Parameters

The Random Forest (RF) classifier utilized 100 trees ($n_{estimators}$=100) with the random state set to 42 for reproducibility. The Support Vector Machine (SVM) employed a Radial Basis Function (RBF) kernel, a regularization parameter (C) of 10, and an automatic kernel coefficient (gamma) scale. The probability parameter was set to True to enable the generation of probability estimates, which were necessary for calculating the One-vs-Rest ROC curve and AUC values.

2) *Performance Comparison*

The SVM achieved a test accuracy of 70.0%, representing a 7.0% gain over the RF model's 63.0%. This result confirms that the RBF kernel was effective at finding the non-linear decision boundaries necessary to separate the dense and overlapping feature clusters identified in the exploratory analysis (Section IV). The 5-Fold Cross-Validation result (73.7% ± 3.7%) provides a stable and representative measure of the SVM's generalized performance.

3) *Confusion Matrices*

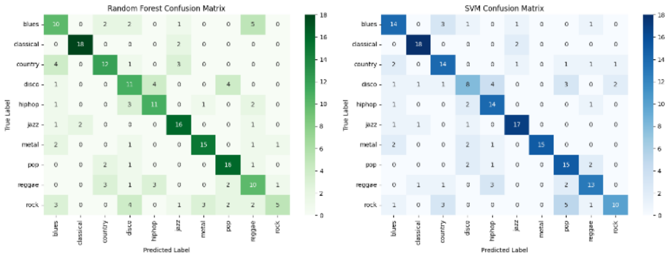

Fig. 4. Confusion Matrices Comparison (Random Forest vs. SVM)

The Confusion Matrix Comparison shows the superiority of the SVM over the RF. The SVM's overall stronger performance across most genres prompted its selection for the detailed evaluation and explainability analysis in the subsequent sections.

B. *Deep Learning (CNN) Results*

1) *Model Parameters*

The CNN architecture utilized three convolutional blocks (32, 64, 128 filters) with Batch Normalization and MaxPooling, followed by a Global Average Pooling layer and dense output layers. The model was trained for 50 epochs with an Adam optimizer (10^{-4} learning rate) and callbacks for early stopping (patience=5) and learning rate reduction.

2) *Performance and Training Dynamics*

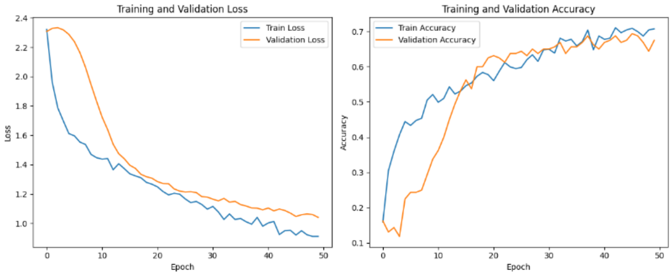

Fig. 5. Training and Validation Loss and Accuracy Plots

The CNN achieved a final Test Accuracy of 80%.

The Training and Validation History plot shows the model's learning evolution. Both training and validation curves show a smooth decrease in loss and an increase in accuracy up to the final epochs.

A clear separation is observed between the training and validation accuracy curves after approximately 20 epochs, with training accuracy pulling ahead. This indicates the CNN is beginning to overfit the training data, absorbing noise specific to the training set.

VI. Evaluation and Discussion

A. *Model Performance Analysis*

The overall accuracy improvement achieved by the SVM (70.0%) over the Random Forest (63.0%) warrants a detailed look at the performance across individual genres, as summarized by the Class-wise Accuracy and Confusion Matrix.

1) *Class-wise Accuracy and Strengths*

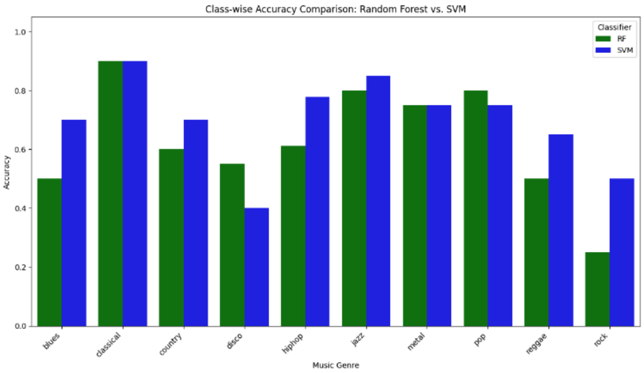

Fig. 6. Class-wise Accuracy Comparison: Random Forest vs. SVM

The Class-wise Accuracy plot highlights the SVM's superiority across most genres. Classical (90% recall) and Jazz (85% recall) were the easiest genres to classify correctly. The SVM Confusion Matrix reveals persistent confusion in the feature space. Disco, with a low recall of 40%, was frequently confused with other genres. Rock achieved 50% recall, indicating that the feature set struggles to distinguish its subtle timbral differences from other classes.

2) *Confusion and Weaknesses*

The Confusion Matrix (SVM) reveals persistent confusion in the feature space. With a low recall of 50%, Disco was frequently misclassified as Hiphop (20%), suggesting shared characteristics (e.g., strong basslines).

Rock achieved 60% recall and was often confused with Country (15%) and Blues (10%), three genres that often share similar timbre (electric and acoustic guitars) and tempo.

B. *One-vs-Rest ROC Curve*

The One-vs-Rest ROC Curve for SVM provides a measure of how well the SVM can discriminate each genre from all other genres combined (the "Rest"). The Area Under the Curve (AUC) values confirm the robust discrimination capability of the model.

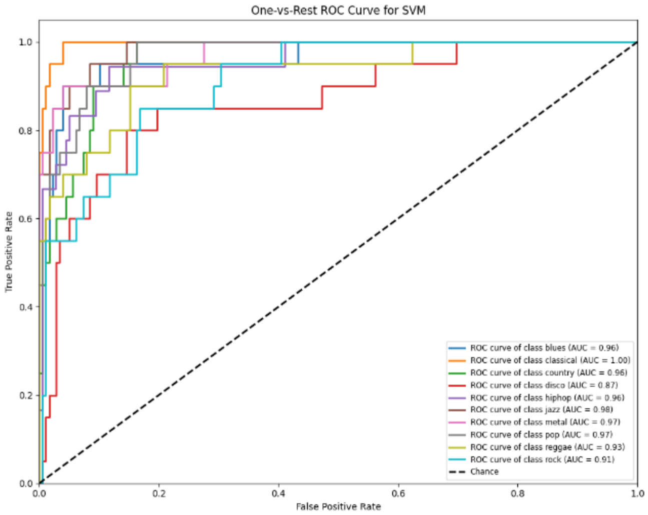

Fig. 7. One-vs-Rest ROC Curve for SVM

Classes like Classical (AUC=1.00) and Jazz (AUC=0.98) are highly separable, confirming the effectiveness of the frequency-domain features in isolating these genres. Classes like Disco (AUC=0.87) and Rock (AUC=0.91) have the lowest AUCs, confirming the difficulties seen in the confusion matrix.

C. *Feature Explainability (Traditional ML)*

The Top Feature Importances (Permutation Importance) plot quantifies the contribution of the features to the SVM's final accuracy.

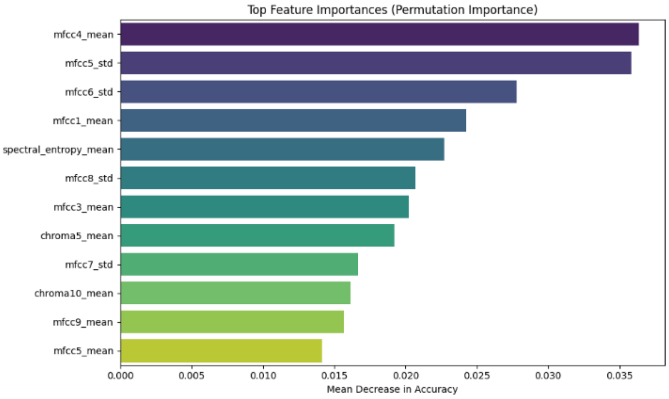

Fig. 8. Permutation Importance

The three most influential features are mfcc4_mean (ΔAcc: 0.0364) and mfcc5_std (ΔAcc: 0.359). These results indicate that the model relies mostly on the mid-order MFCCs to classify genres. This suggests that the timbre is the most prominent acoustic characteristic for genre differentiation.

Furthermore, spectral_entropy_mean (ΔAcc: 0.0227) is the fifth most important feature, confirming that the complexity and disorder of the power spectrum are also factors utilized by the classifier.

D. *Model Comparison: Performance vs. Explainability*

The SVM model achieved a test accuracy of 70.0%. The CNN model, however, demonstrated a significant predictive advantage, achieving 80.0% test accuracy, which is a 10.0% gain.

Given this significant difference, a real-world deployment would prioritize the CNN for accuracy, but the SVM remains essential. The SVM provides the crucial 70.0% performance with verifiable interpretability, allowing to trace why the prediction was made, which is important for applications requiring decision traceability.

VII. Conclusion

The comparative analysis established a significant 10.0% performance difference. The CNN (Deep Learning) model achieved the maximum performance with a test accuracy of 80.0%, achieving better performance through features learned from Log-Mel Spectrograms. In contrast, the feature-based SVM model achieved a test accuracy of 70.0%.

This performance gap confirms that the feature set cannot fully capture the complex variance that the CNN extracts from the 2D spectrogram input. Nevertheless, the SVM provides the advantage of interpretability: the Permutation Importance analysis confirmed that the model's decisions rely on the variability of the mid-order MFCC and the Spectral Entropy.

As the CNN was significantly under-trained at 20 epochs, future work should continue exploring methods like data augmentation to address the GTZAN dataset's limitation, to improve training stability and potentially raise the CNN's performance past 80.0%.

The most promising direction involves exploring hybrid architectures that combine the strengths of both pipelines. This could utilize the CNN's feature learning capabilities but force the final classification layers to incorporate the PWP and Spectral Entropy features directly.

References

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 4, pp. 357-366, August 1980, doi: 10.1109/TASSP.1980.1163420.

[2] Tan, P., Steinbach, M., Karpatne, K., & Kumar, V. (2018). *Introduction to Data Mining* (2nd ed.). https://www-users.cse.umn.edu/~kumar001/dmbook/index.php

[3] Pons, J., et al. (2017). End-to-end learning for music audio tagging at scale. *arXiv preprint arXiv:1711.02520.* https://arxiv.org/abs/1711.02520

[4] Alican Akman, Björn W. Schuller. Audio Explainable Artificial Intelligence: A Review. *Intell Comput.* 2024;3:0074. DOI:10.34133/icomputing.0074

[5] Olteanu, Andrada (2020). *GTZAN Dataset - Music Genre Classification*. Kaggle. Retrieved from: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

[6] Brian McFee & the librosa team. (2024). *librosa v0.10.1 documentation.* https://librosa.org/doc/latest/index.html

[7] PyWavelets Developers. (2024). *PyWavelets Documentation.* https://pywavelets.readthedocs.io/en/latest/