Musical genre classifier – a comparison of different methods

Lucian-Daniel Jimon  
UTCN – TM 1st yearjimon.mi.lucian@student.utcluj.ro

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—component, formatting, style, styling, insert (key words)

# Introduction

Musical genre classification is a complex, but important problem in the domain of MIR (music information retrieval), being crucial for several other tasks like music recommendations, audio catalogs, playlist generation etc.

The current paper proposes to compare popular machine learning algorithms for solving the mentioned task, namely the SVM, Random Forrest and CNN algorithms, all which will be used on the GTZAN dataset, which is possibly the most used free online dataset for musical genre classification. This paper aims to provide a comprehensive comparison of SVM, RF, and CNN models for the task of musical genre classification using the well-known GTZAN dataset. We employ both traditional machine learning techniques and modern deep learning approaches to evaluate their performance in terms of classification accuracy, training time, and computational efficiency. The study involves several key steps, which are:

* Feature Extraction: A rich set of features is extracted from the audio files, like Mel-frequency cepstral coefficients (MFCCs), chroma features, spectral contrast etc. These features are then used to train SVM and RF classifiers
* Deep Learning Approach: For the CNN model, the spectrograms are used as input to capture the temporal and spectral characteristics of the audio signals
* Model Training and Evaluation: Each model is trained on the GTZAN dataset, and their performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. Additionally, an analysis of misclassifications is performed

By comparing these methodologies, the advantages and trade-offs associated with each model are aimed. This comparison could even provide valuable insight for future works.

# State-of-the-art

As being a crucial part of other more complex music related tasks like music generation, there are already well-based studies on genre classification, some of them are presented below.

## Music genre classification with ResNet

When talking about complex classification problems which require a robust CNN architecture, a starting point can be the use of pretrained ResNet models which can be deployed and fine-tuned for the current task []. In [] the authors managed to combine ResNet with BiGRU (Bi-gated recurrent unit), obtaining a final accuracy of 90%, which situates it as one of the best performances for genre classification.

## Comparison of music genre classifiers

Another paper treats the topic in a similar manner to the current one: comparing SVM, RF and CNN []. In this paper, a multi-model CNN is used based on spectrograms of the GTZAN dataset, obtaining a result for accuracy of 87%, 90% and 88% for CNN, SVM and RF.

## Comparison of music genre classifiers

The paper [] is treating the topic of music genre classification once again in an investigation manner, presenting a 3-layern CNN model, a SVM using an RBF (radial basis function) as kernel and the classic KNN (K-Nearest Neighbors) algorithms. The results show a significant difference in the preparation of the data, e.g., for the CNN the final accuracy was 82% if the data was preprocessed, and 53% otherwise.

Based on the mentioned research and many other papers, notebooks and presentations (mentioned in the Bibliography section of this paper), it can be perceived that the problem of genre classification is a rather complicated task which can underperform in many cases and it is extremely dependent on the features, dimension and quality of the training set and the overall musical characteristic of different songs, as some genres can overlap, and the limitations are plenty. Therefore, a comparison of methods can help the domain in advancing and enhancing the classification process.

# Theoretical foundations

## Introduction to Musical Genres

Musical genres play a crucial role in the classification and organization of music. Each genre has distinct musical characteristics, including specific rhythmic patterns, harmonic structures, melodicity, and instrumentation. These genres are used for various applications, including music recommendation systems, automated playlist generation, and enhancing user experience in digital music platforms (for example Spotify, YouTube etc.). Understanding and identifying musical genres create efficient navigation and enhance the satisfaction of the users, but could also help learn music for student, as they could compare genres which are shown to them based on reliable algorithms.

## Data Preprocessing and Feature Extraction

The preprocessing of audio data is essential to convert raw audio signals into a proper format for machine learning models. Raw audio is typically high-dimensional and contains noise, which can degrade the performance of classifiers. Preprocessing steps such as normalization, silence removal, and noise reduction improve the signal-to-noise ratio, making the data more tractable.

Feature extraction transforms raw audio data into a set of numerical features that capture the essential characteristics of the audio signal. Some of the key features include:

* Mel-frequency Cepstral Coefficients (MFCCs): MFCCs are extracted from the Fourier transform of the audio signal and represent the short-term power spectrum. They can be effective in representing the timbral texture of the audio.
* Chroma Features: Chroma features represent the 12 different pitch classes of the musical octave (per se, the present notes in a sequence, from C to B#), hence capturing harmonic and tonal content.
* Spectral Features: These include the spectral centroid (the center of mass of the spectrum), spectral bandwidth (the width of the spectrum), spectral roll-off (the frequency below which a certain percentage of the total spectral energy is), and spectral flatness (a measure of the noisy characteristic).
* Temporal Features: Temporal features such as the zero-crossing rate (the rate at which the signal changes sign) and root mean square (RMS) energy (a measure of the signal’s power) describe the signal's temporal dynamics.

## Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep learning models that have revolutionized the field of pattern recognition, particularly in the areas of image and audio processing. CNNs are designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolutional layers, pooling layers, and fully connected layers.

### Architecture of CNNs

CNNs are composed of several types of layers, each serving a specific purpose:

* Convolutional Layers: These layers apply a set of learnable filters (kernels) to the input data. Each filter slides (convolves) across the width and height of the input volume, computing the dot product between the filter entries and the input, and producing a two-dimensional activation map of that filter. The ability to use multiple filters allows CNNs to detect various features in the input, such as edges, textures, and patterns.
* Pooling Layers: Pooling, or subsampling, layers reduce the spatial dimensions (width and height) of the input volume for the next convolutional layer. This downsampling process helps in reducing the computational load, controlling overfitting, and making the model invariant to small translations of the input. Common pooling operations include max pooling and average pooling.
* Fully Connected Layers: After several convolutional and pooling layers, the high-level reasoning in the neural network is done via fully connected layers. These layers are similar to those in traditional neural networks and flatten the input to a vector of features, which are then fed into one or more fully connected layers to produce the final classification output.
* Activation Functions: Non-linear activation functions such as ReLU (Rectified Linear Unit) are applied after convolutional layers to introduce non-linearity into the model, enabling it to learn more complex functions.

### Advantages of CNNs for Musical Genre Classification

CNNs are particularly effective for musical genre classification for several reasons:

* Automatic Feature Learning: One of the key strengths of CNNs is their ability to automatically learn and extract hierarchical features directly from raw input data. For musical genre classification, spectrograms—visual representations of the spectrum of frequencies in a sound signal as they vary with time—are used as input to CNNs. Unlike traditional approaches that rely on handcrafted features, CNNs can learn both low-level features (e.g., edges, textures) and high-level features (e.g., musical patterns, rhythms) from the spectrograms.
* Spatial Hierarchy and Locality: Convolutional layers exploit the local spatial coherence in the data, which is crucial for understanding the local patterns in spectrograms. By using local receptive fields, CNNs can capture essential patterns in different regions of the spectrogram without the need for extensive preprocessing.
* Invariance to Translation: Pooling layers provide translation invariance, which means that small shifts in the input (e.g., changes in tempo or pitch) do not affect the output of the network significantly. This property is particularly useful for music data, where variations in the performance do not necessarily change the genre.
* Scalability and Performance: CNNs have been shown to scale effectively with large datasets and can leverage parallel processing capabilities of modern GPUs, making them suitable for handling extensive and complex datasets like the GTZAN dataset.

## Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are powerful supervised learning models used for classification and regression tasks. The fundamental idea behind SVMs is to find the optimal hyperplane that best separates the data points of different classes by maximizing the margin, which is the distance between the hyperplane and the nearest data points from each class. These nearest data points are called support vectors, and they are critical in defining the hyperplane.

### Kernel trick

One of the most significant strengths of SVMs is the kernel trick, which allows SVMs to operate efficiently in high-dimensional spaces. The kernel trick involves using a kernel function to implicitly map the input features into a higher-dimensional feature space without explicitly performing the transformation. This enables SVMs to find a linear separation in the higher-dimensional space even if the data is not linearly separable in the original space.

Common kernel functions include:

* Linear Kernel: Useful for linearly separable data.
* Polynomial Kernel: Adds polynomial features of the input data, allowing for more complex decision boundaries.
* Radial Basis Function (RBF) Kernel: Also known as the Gaussian kernel, it can handle non-linear relationships by creating decision boundaries that are complex shapes in the original feature space.
* Sigmoid Kernel: Similar to neural networks, it uses the sigmoid function to create a hyperplane.

### *SVM Parameters*

To effectively utilize SVMs for classification, several hyperparameters need to be tuned:

* Kernel: Specifies the type of kernel function to use (e.g., linear, polynomial, RBF).
* C (Regularization Parameter): Controls the trade-off between achieving a low training error and a low testing error. A small C value makes the decision surface smooth, while a large C value aims to classify all training examples correctly by allowing the model to overfit.
* Gamma (Kernel Coefficient): Defines how far the influence of a single training example reaches in the case of RBF, polynomial, and sigmoid kernels. A low gamma value means 'far' and a high gamma value means 'close'.
* Degree: Relevant only for the polynomial kernel, it specifies the degree of the polynomial function.

### *Advantages for Audio Classification*

SVMs are particularly suitable for audio classification tasks due to their robustness in high-dimensional feature spaces. Audio data, when transformed into features such as Mel-frequency Cepstral Coefficients (MFCCs), chroma features, and spectral features, often results in a large and complex feature space. SVMs can handle this complexity effectively.

* Effective in High-Dimensional Spaces: SVMs perform well in scenarios where the number of features is large compared to the number of samples, making them ideal for audio classification.
* Robust to Overfitting: Proper regularization ensures that SVMs are less likely to overfit, even in high-dimensional spaces.
* Flexibility with Kernels: The ability to choose different kernel functions allows SVMs to model various types of relationships in the data.

## Random Forests (RFs)

Random Forests (RFs) are an ensemble learning method that improves classification performance by combining the predictions of multiple decision trees. The main idea behind RFs is to create a 'forest' of decision trees (hence the name), each trained on a different subset of the data, and aggregate their predictions to make a final decision.

### Ensemble Learning

Ensemble learning involves combining the predictions of multiple models to produce a more accurate and robust prediction than any individual model could achieve. In the case of RFs, the ensemble consists of numerous decision trees. Each tree is trained on a bootstrap sample (a random subset with replacement) of the training data. The final prediction is made by aggregating the predictions of all individual trees, typically through majority voting for classification tasks.

### Advantages for Musical Genre Classification

RFs offer several advantages for musical genre classification:

* Feature Importance: RFs can emphasize the importance of each feature in making predictions. This helps in understanding which features contribute most to distinguishing between musical genres.
* Robustness to Noisy Data and Outliers: By averaging the predictions of multiple trees, RFs reduce the impact of noisy data and outliers, leading to more stable and reliable predictions.
* Handling Complex Interactions: RFs can model complex interactions between features, making them suitable for tasks where the relationships between features are not straightforward.
* Reduced Overfitting: The ensemble approach of combining multiple trees reduces the risk of overfitting, as individual trees are less likely to capture noise in the data.

Considering all the presented theoretical aspects, the implementation of the comparison-related experiments will rely on the characteristic features which will be extracted and used for the training, the optimization of the used models for SVM and CNN and the correct approach of testing/refining the models.

# Implementation

## Preprocessing and feature extraction

## Building the models

## Training and hypertuning

## Preliminary results