Report - Music Genre Classification

Introduction:

Increasing music production and its shift to online platforms such as Spotify, YouTube Music, Apple Music, and many other streaming platforms has increased interest in music information retrieval as a topic in data science. The music genres are vague because of the evolution of music over time. However, customers of music streaming platforms prefer to listen to music categorised by genre, artist, or album. The successful music classification will improve customer experience and increase revenue for music streaming services.

As humans manually categorise and label the music genre, this may lead to different interpretations of the music genre from individual to individual. This study aims to investigate an extensive database of music with 10 music genres, namely, disco, metal, reggae, blues, rock, classical, jazz, hip-hop, country, and pop.

We are building a neural network that classifies music into its respective genre. This will automatically classify music based on different features and parameters instead of manually classifying it into its respective genres. We will compare the accuracies of this model and the preexisting testing dataset and draw the necessary conclusions. We will improve the model to identify and classify new music into its genre accurately.

Literature Review:

There have been various studies on music genre recognition, transfer learning, and audio-based classification. Jimenez and Jose (2018) explored how deep neural networks can recognise music genres, pointing out the benefits of using multiple frames and transfer learning. However, they also noted that dealing with large datasets can be time-consuming. Pan and Yang (2010) conducted a thorough review of transfer learning, highlighting its potential for transferring knowledge across different tasks. Despite its promise, they acknowledged the challenge of dealing with negative transfer. Van den Oord, Dieleman, and Schrauwen (2014) looked into using transfer learning for music classification, specifically when the source and target tasks are closely related. Tzanetakis and Cook (2002) focused on classifying musical genres from audio signals, considering features like rhythm and pitch. They also discussed the ongoing challenge of distinguishing between speech, sound, and music, especially in melody and singer voice extraction.

Dataset:

GTZAN is a dataset for the classification of audio signals by musical genres. The dataset consists of 1,000 audio tracks, each 30 seconds long. It contains 10 genres, each represented by 100 tracks. The tracks are all 22,050Hz Mono 16-bit audio files in AU format. The genres are blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock.

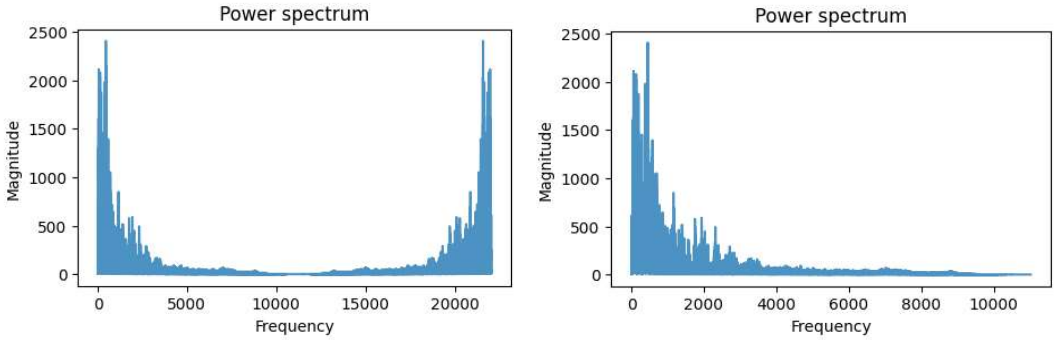
Music Features:

Mel-frequency cepstral coefficients (MFCCs) are a feature representation widely used in audio signal processing and speech recognition tasks. They are derived from the Mel-frequency cepstrum, which represents the short-term power spectrum of sound in a way that is perceptually relevant to humans.
The detailed description of

- **Perceptual Representation:** MFCCs are derived from the Mel-frequency cepstrum, which represents the short-term power spectrum of sound in a way that is perceptually relevant to humans. By focusing on the characteristics of human auditory perception, MFCCs provide a compact representation of audio signals.
- **Feature Extraction:** MFCCs capture essential aspects of an audio signal's spectral content, such as timbre and pitch, while discarding irrelevant information. This makes them effective features for tasks such as speech recognition, and music genre classification.
- **Robustness:** MFCCs are robust to variations in input signal characteristics, such as changes in amplitude and background noise. This robustness is crucial for real-world applications where audio signals may vary widely in their acoustic properties.
- **Widespread Usage:** Due to their effectiveness and efficiency, MFCCs are widely used in audio signal processing. They serve as a fundamental feature representation in numerous applications, ranging from speech recognition systems to music information retrieval algorithms.

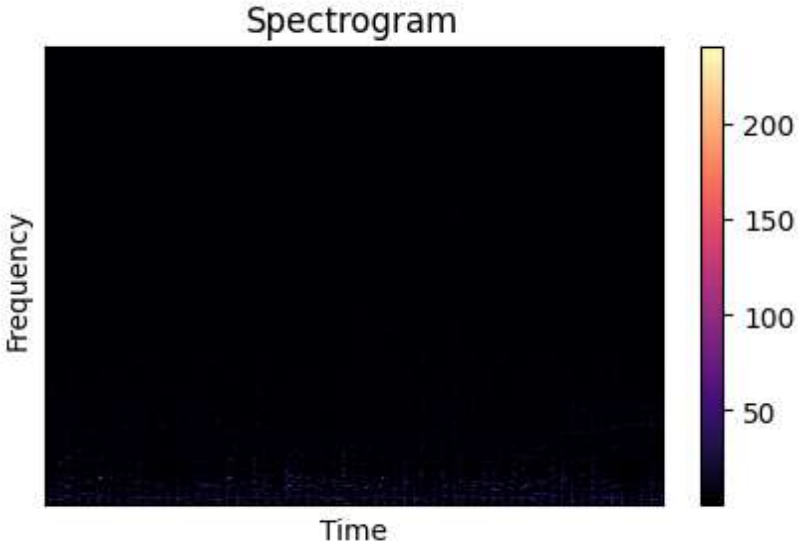
How are MFCCs extracted?

1. **Power Spectrum Calculation:**
    - The Fast Fourier Transform (FFT) is a mathematical algorithm that converts a signal from the time domain to the frequency domain. By applying FFT to the audio signal, we obtain the frequency components present in the signal and their corresponding magnitudes.
    - The power spectrum represents the signal's power distribution (or energy) across different frequencies. It provides insights into which frequencies contribute most to the overall signal. We consider only the left half of the spectrum for further analysis as it is symmetric around the median frequency.



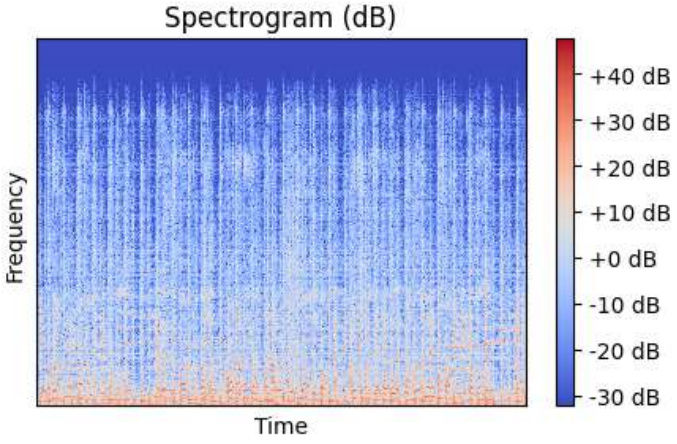
2. Short-Time Fourier Transform (STFT):

- The Short-Time Fourier Transform (STFT) is an extension of the Fourier Transform that allows us to analyse how the frequency content of a signal changes over time.
- STFT divides the audio signal into short overlapping segments, typically using a windowing function. It computes the Fourier transform for each segment, providing a time-frequency representation of the signal.
- By analysing the STFT, we can observe how the spectral characteristics of the signal evolve, which is valuable for tasks such as detecting transient events or analysing non-stationary signals.

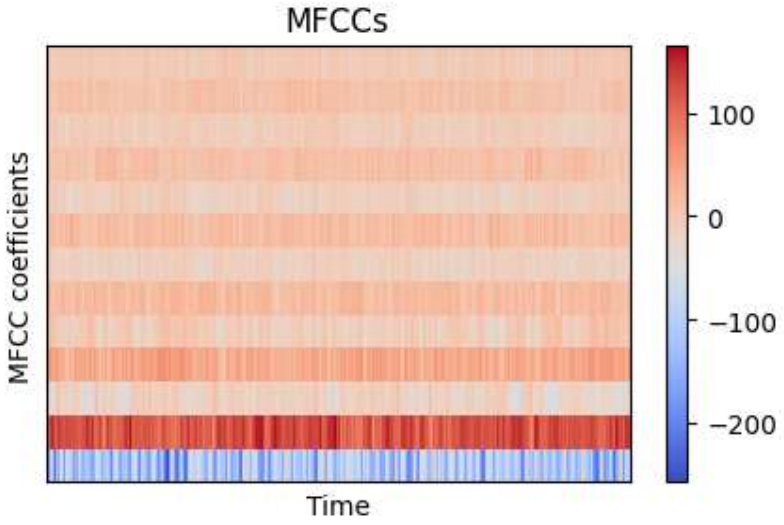


3. Logarithmic Scale for Spectrogram:

- After computing the Short-Time Fourier Transform (STFT) to obtain the spectrogram, we apply a logarithmic transformation to the magnitude values.
- The logarithmic scale compresses the magnitude values, making small changes more visible and improving the clarity of the spectrogram's representation.
- Additionally, the logarithmic scale allows us to express the magnitude values in decibels (dB), a standard unit for measuring the intensity or power of a signal.



4. **Mel-Frequency Cepstral Coefficients (MFCCs) Extraction:**
    - Mel-Frequency Cepstral Coefficients (MFCCs) are a set of features widely used in speech and audio processing tasks to capture the spectral characteristics of the signal.
    - MFCCs mimic the human auditory system's response to sound by applying a non-linear mel-frequency scale to the power spectrum.
    - The process of extracting MFCCs involves several steps, including applying a Mel filterbank to the power spectrum, taking the logarithm of the filterbank energies, performing Discrete Cosine Transform (DCT), and selecting the resulting coefficients as features.
    - MFCCs provide a compact representation of the spectral envelope of the audio signal, capturing essential features related to pitch, timbre, and phonetic content.



Neural Networks

1. **CNN**

    Convolutional Neural Networks (CNNs) have emerged as a powerful extension of traditional Artificial Neural Networks (ANNs), particularly well-suited for extracting features from grid-like matrix datasets.

    Convolutional neural networks (CNNs) have been actively used for various music classification tasks, such as music tagging, genre classification, and user-item latent feature prediction for recommendation. CNNs assume features in different levels of hierarchy and can be extracted by convolutional kernels. The hierarchical features are learned to achieve a given task during supervised training.

    This paper discusses the fundamental concepts of CNNs, their architecture, and their effectiveness in music classification tasks based on existing research findings.

    ARCHITECTURE

    Input Layer: Receives raw input data, such as images(MFCC in this case)
    Convolutional Layers:

- Extract features using learnable filters (kernels).
- Filters slide over input data, producing feature maps.
- Multiple layers capture hierarchical features.

Activation Function:

- Applies non-linearity to feature maps.
- Commonly uses ReLU to introduce non-linearity.

Pooling Layers:

- Downsample feature maps, reducing spatial dimensions.
- Retain essential information while reducing computational complexity.
- Max pooling and average pooling are common operations. (Max pooling used in this case)

Fully Connected Layers:

- Integrate extracted features for classification or regression.
- Each neuron is connected to all neurons in the previous layer.
- The final layer produces output predictions.

Output Layer:

- Produces predictions based on task requirements.
- Softmax activation for classification tasks.

Optional Layers:

- Flattening Layer: Converts multi-dimensional feature maps into one-dimensional vectors.
- Dropout Layer: Randomly sets a fraction of input units to zero during training to prevent overfitting.
- Batch Normalisation Layer: Normalizes activations, improving stability and speed of training.

Padding:

- Optionally preserves spatial dimensions of input data.
- 'Same' padding maintains input size; 'valid' does not.

2. RNN

Recurrent Neural Networks (RNNs) are a type of artificial neural network designed to process sequential data, making them particularly suitable for tasks where the input and output sequences can vary in length. Unlike feedforward neural networks, which process data in a strictly forward direction, RNNs incorporate recurrent connections to maintain a memory of previous inputs. At each time step, an RNN computes the current hidden state by combining the current input with the previous one. This recurrent structure enables RNNs to capture temporal dependencies within sequential data, making them effective in tasks that involve understanding context or patterns over time.

Recurrent Neural Networks (RNNs) play a vital role in music genre classification due to their ability to capture temporal dependencies within music sequences. When analysing music, the sequence of notes, chords, and rhythms over time holds crucial information about its genre. RNNs, with their recurrent connections, excel at understanding these sequential patterns, making them well-suited for music genre classification tasks.

3. CRNN

The Convolutional Recurrent Neural Network (CRNN) architecture is a hybrid model that combines convolutional neural networks (CNNs) with recurrent neural networks (RNNs). This combination allows CRNNs to process spatial and temporal features in sequential data such as audio, making them particularly well-suited for tasks like music genre classification.

ARCHITECTURE DESCRIPTION

Convolutional Layers:

- Responsible for capturing spatial patterns and representations from the input.
- Utilise learnable filters to generate feature maps.
- Apply activation functions to enhance discriminative power.

Pooling Layers:

- Downsample feature maps to reduce spatial dimensions.
- Extract salient features while controlling overfitting.
- Commonly employ max or average pooling operations.

Recurrent Layers:

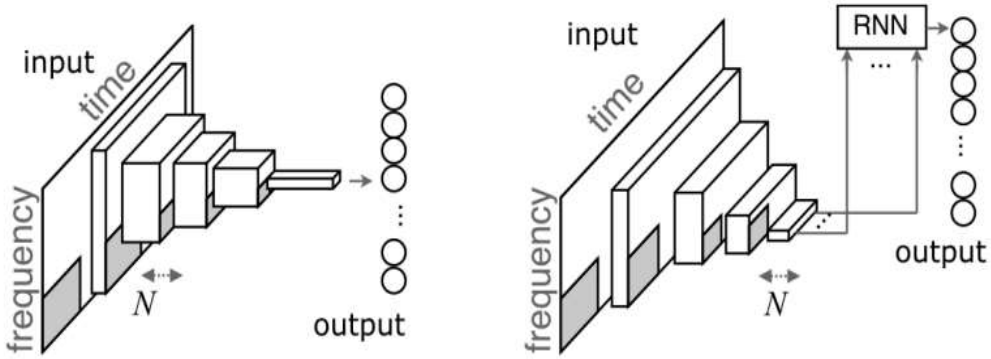
- Model temporal dependencies and sequential patterns.
- Maintain an evolving internal state over time.
- Bidirectional layers capture contextual information.

Fully Connected Layers:

- Integrate extracted features for prediction.
- Flatten output into one-dimensional vectors.
- Often employ softmax activation for multi-class tasks.

Training and Optimization:

- Trained via backpropagation and gradient descent.
- Utilise optimisation techniques such as SGD, Adam, or RMSprop.
- Learn discriminative features from input data for accurate predictions.



(a) CNN Architecture (b) CRNN Architecture

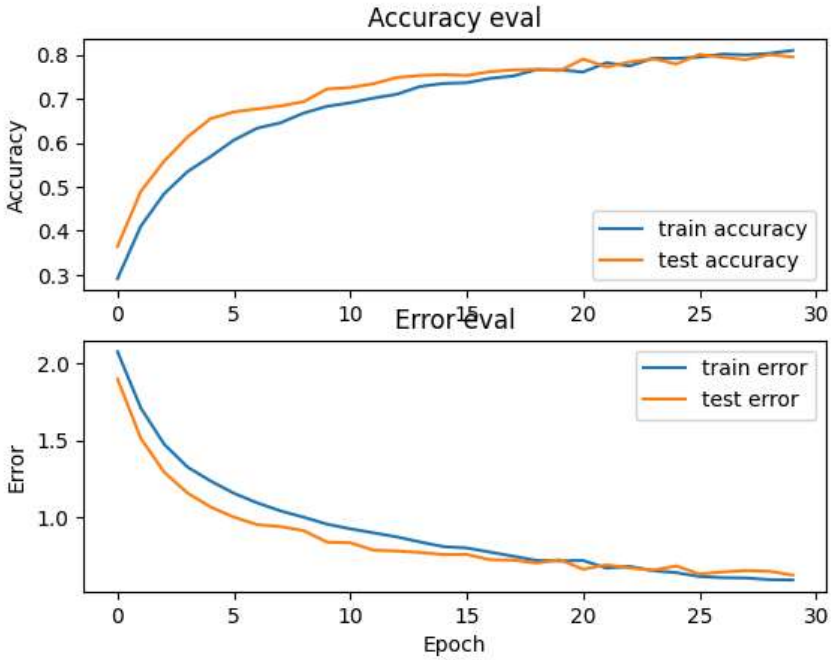
Model Network

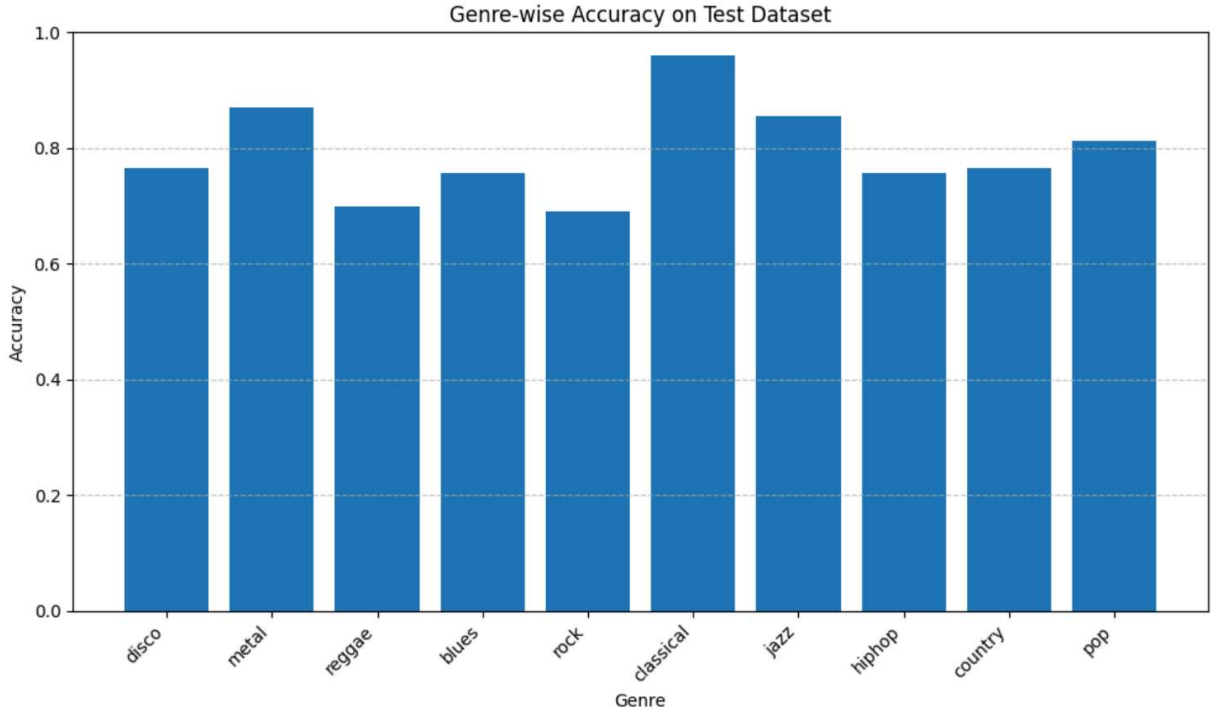
Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 128, 11, 32)	320
max_pooling2d_9 (MaxPooling2D)	(None, 64, 6, 32)	0
batch_normalization_9 (BatchNormalization)	(None, 64, 6, 32)	128
conv2d_10 (Conv2D)	(None, 62, 4, 32)	9,248
max_pooling2d_10 (MaxPooling2D)	(None, 31, 2, 32)	0
batch_normalization_10 (BatchNormalization)	(None, 31, 2, 32)	128
conv3d_3 (Conv2D)	(None, 30, 1, 32)	4,128
max_pooling2d_11 (MaxPooling2D)	(None, 15, 1, 32)	0
batch_normalization_11 (BatchNormalization)	(None, 15, 1, 32)	128
reshape (Reshape)	(None, 15, 32)	0
lstm (LSTM)	(None, 15, 64)	24,832
lstm_1 (LSTM)	(None, 64)	33,024
dense_6 (Dense)	(None, 64)	4,160
dropout_3 (Dropout)	(None, 64)	0
dense_7 (Dense)	(None, 10)	650

Performance Metrics

In our research on music genre classification, we investigated the performance of Convolutional Neural Networks (CNNs) and Convolutional Recurrent Neural Networks (CRNNs) as classification models. Our study utilised the GTZAN dataset, consisting of 1,000 songs categorised into 10 genres, each containing 100 songs. The experimental findings revealed significant differences in the performance of the two models. The CNN model achieved an accuracy of **68.34**% on the test dataset, showcasing its ability to extract spatial features from spectrogram representations of audio signals for genre classification.

However, the CRNN model surpassed the CNN model, achieving an accuracy of **79.23**% on the same test dataset. By integrating convolutional and recurrent layers, the CRNN model adeptly captured both spatial and temporal features of music signals, resulting in a notable improvement in classification accuracy compared to the CNN model. This enhancement underscores the importance of considering temporal dynamics alongside spatial features in music genre classification tasks. Overall, our results demonstrate the effectiveness of both CNNs and CRNNs in music genre classification, with CRNNs exhibiting superior performance by leveraging a richer representation of audio signals.





Conclusion

The application of CNN and CRNN in the case of music genre classification is explored. This method requires a large dataset that needs to be trained from scratch. In the case of having small data for transfer learning, the accuracy will be lower.

For each track of the GTZAN dataset, Mel Spectrum is obtained. This can be done by using the Python package library libROSA. A software system is implemented to classify the music according to its genre. The collection of 1000 songs of 10 different genres is experimented with, and 7 of these 10 meet with the highest accuracy rate.

In future work, some other techniques will be used to find subgenres in all these types of genres. By implementing this, the complete package of each piece of music has been found. Another extension in this work would be a methodology for analysing all input formats, such as WAV, MP3, AU, etc., which will be tested on a common platform.