

**Name – Pareekshit Sachan**

**Student Number – 222011963**

**DSP721S Project Report**

**Music Genre Classification using Spectrograms**

Table of Contents

[Introduction 4](#_Toc183702390)

[1.1 Project Background 4](#_Toc183702391)

[1.2 Objectives 5](#_Toc183702392)

[1.3 Scope 5](#_Toc183702393)

[2. Theory and Background 5](#_Toc183702394)

[2.1 Relevant DSP Concepts 5](#_Toc183702395)

[2.1.1 Short-Time Fourier Transform (STFT) 5](#_Toc183702396)

[2.1.2 Spectrograms 6](#_Toc183702397)

[2.2 Convolutional Neural Networks (CNNs) 7](#_Toc183702398)

[2.3 Specific Domain Knowledge 10](#_Toc183702399)

[3.Data Acquisition and Preprocessing 10](#_Toc183702400)

[3.1 Dataset Description 10](#_Toc183702401)

[3.2 Audio Sampling and Resampling 10](#_Toc183702402)

[3.3 Preprocessing Techniques 10](#_Toc183702403)

[3.3.1 Spectrogram Generation 10](#_Toc183702404)

[3.3.2 Image Normalization 11](#_Toc183702405)

[4. Methodology 11](#_Toc183702406)

[4.1 Signal Processing Techniques 11](#_Toc183702407)

[4.1.1 Audio Signal Representation 11](#_Toc183702408)

[4.1.2 Spectrogram Generation 11](#_Toc183702409)

[4.2 Preprocessing of Audio Signals 12](#_Toc183702410)

[4.3 Convolutional Neural Network (CNN) Architecture 12](#_Toc183702411)

[4.4 Training Process 13](#_Toc183702412)

[4.5 Evaluation Metrics 13](#_Toc183702413)

[4.6 Implementation Tools 13](#_Toc183702414)

[5.Results and Analysis 13](#_Toc183702415)

[5.1. Model Training Performance 14](#_Toc183702416)

[5.2. Confusion Matrix and Classification Report 14](#_Toc183702417)

[5.3. Visualization of Spectrograms 16](#_Toc183702418)

[5.4. Model Behavior 17](#_Toc183702419)

[6.Discussion 17](#_Toc183702420)

[6.1. Comparison with Literature 17](#_Toc183702421)

[6.2. Challenges and Limitations 17](#_Toc183702422)

[6.3. Practical Implications 18](#_Toc183702423)

[6.4. Insights for Future Work 18](#_Toc183702424)

[7.Conclusion 18](#_Toc183702425)

[8.References 19](#_Toc183702426)

# Introduction

## 1.1 Project Background

Music genre recognition is part of the larger field of Music Information Retrieval (MIR) and audio signal processing. Music information retrieval (MIR) combines signal processing, machine learning, and music theory to study and make sense of musical material. Its main goal is to help algorithms understand and process musical data smartly and effectively.  
  
G. Tzanetakis et al. laid the groundwork for music genre classification by using timbral and temporal features to identify genres. Attributes like Spectral Centroid, Spectral Roll-off, Spectral Flux, Zero Crossing Rate, and Mel Frequency Cepstral Coefficients were crucial in this initial study. A different method, presented by H. Deshpande and colleagues, consists of converting music signals into spectrograms and then deriving visual characteristics for classification from these spectrograms. This approach provides a notable benefit in terms of storage efficiency since spectrograms take up much less space than raw audio files.  
  
Lately, convolutional neural networks (CNNs) have become more popular in image classification tasks because they can automatically extract features directly from input data. CNNs are a perfect option for tasks like categorizing music genres based on spectrograms. In contrast to conventional neural networks, CNNs have the ability to autonomously derive hierarchical attributes from spectrograms, simplifying the genre classification procedure.  
  
The process of categorizing music genres with spectrograms usually includes transforming music signals into spectrograms, which act as visual depictions of the signals. These spectrograms show both the time and frequency elements, giving a detailed representation of the music signals. The spectrograms are fed into a CNN classifier for independent feature extraction and classification. The main benefit of utilizing a CNN is its ability to recognize intricate patterns within the data without the need for manual feature engineering.

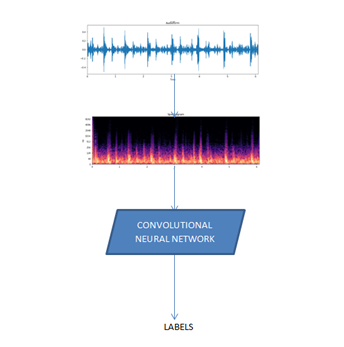
The process of categorizing music genres through spectrograms commonly includes transforming music signals into spectrograms, which act as visual depictions of the signals. These spectrograms show both time and frequency elements, offering a complete representation of the music signals. The spectrograms are next fed into a CNN classifier, which independently carries out feature extraction and classification. The main benefit of utilizing a CNN is its ability to recognize intricate patterns in the data without the need for manual feature engineering.  
  
This strategy is in accordance with the technique outlined by Nirmal M R and Dr. Shajee Mohan B S in their study on Music Genre Classification Using Spectrograms, employing spectrograms as CNN classifier input to accurately categorize music genres by their visual and temporal characteristics. Their research emphasizes the advantages of using CNNs in categorizing music genres, showcasing their effectiveness and precision in processing visual information from audio signals.

## 1.2 Objectives

* Generate spectrograms from a diverse dataset of audio files spanning multiple genres.
* Design and train a CNN model to classify these spectrograms accurately into their respective genres.
* Evaluate the model’s performance on unseen data and interpret its predictions to improve generalizability.

## 1.3 Scope

This project focuses on using CNNs to classify music genres from spectrograms. While other audio features like Mel-Frequency Cepstral Coefficients (MFCCs) or chroma features could enhance classification accuracy, this project is restricted to spectrogram-based approaches to demonstrate the efficacy of visual analysis in genre classification.

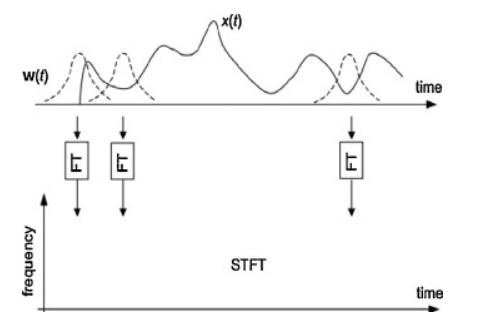


# 2. Theory and Background

## 2.1 Relevant DSP Concepts

### 2.1.1 Short-Time Fourier Transform (STFT)

The short time Fourier transform is introduced to overcome the problems of the FFT. It is usually used for the extraction of narrow-band frequency content in non-stationary or noisy signals. The basic idea of STFT is to develop the initial signal into small time windows and employ the FT to each time segment for expressing the variation in signal frequency content over time that lived in that segment



It provides constant absolute bandwidth analysis to identify harmonic components and offers constant resolution in two-dimensional representation, regardless of the actual frequency. The mathematical equation for the short time Fourier Transform is given by:

STFT is used to convert time-domain signals into the time-frequency domain, enabling the analysis of frequency variations over time. This technique is critical in understanding the spectral properties of music genres.

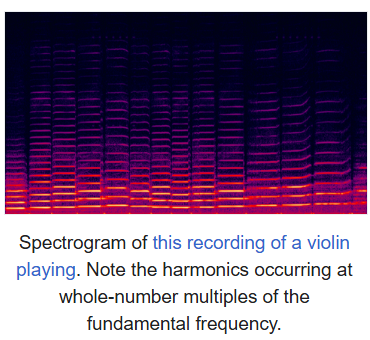
### 2.1.2 Spectrograms

A **spectrogram** is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called **sonographs**, **voiceprints**, or **voicegrams**. When the data are represented in a 3D plot they may be called waterfall displays.

Spectrograms are used extensively in the fields of music, linguistics, sonar, radar, speech processing, seismology, ornithology, and others. Spectrograms of audio can be used to identify spoken words phonetically, and to analyse the various calls of animals.

A spectrogram can be generated by an optical spectrometer, a bank of band-pass filters, by Fourier transform or by a wavelet transform (in which case it is also known as a **scaleogram** or **scalogram**).

A spectrogram is usually depicted as a heat map, i.e., as an image with the intensity shown by varying the colour or brightness.



A black and white image of a mountain

Description automatically generated

## 2.2 Convolutional Neural Networks (CNNs)

A Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The architecture of CNNs is inspired by the visual processing in the human brain, and they are well-suited for capturing hierarchical patterns and spatial dependencies within images.

* Key components of a Convolutional Neural Network include:
* **Convolutional Layers:** These layers apply convolutional operations to input images, using filters (also known as kernels) to detect features such as edges, textures, and more complex patterns. Convolutional operations help preserve the spatial relationships between pixels.

A diagram of a layer of layers

Description automatically generated

* **Pooling Layers:** Pooling layers downsample the spatial dimensions of the input, reducing the computational complexity and the number of parameters in the network. Max pooling is a common pooling operation, selecting the maximum value from a group of neighboring pixels.

A diagram of a pooling

Description automatically generated

* **Activation Functions:** Non-linear activation functions, such as Rectified Linear Unit (ReLU), introduce non-linearity to the model, allowing it to learn more complex relationships in the data.

A diagram of steps and steps

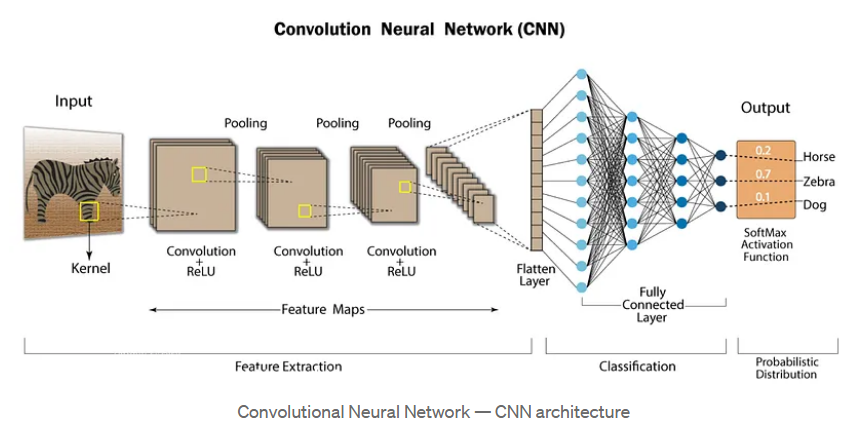
Description automatically generated

* **Fully Connected Layers:** These layers are responsible for making predictions based on the high-level features learned by the previous layers. They connect every neuron in one layer to every neuron in the next layer.

A diagram of a layer of layers

Description automatically generated

CNNs are trained using a large dataset of labeled images, where the network learns to recognize patterns and features that are associated with specific objects or classes. Proven to be highly effective in image-related tasks, achieving state-of-the-art performance in various computer vision applications. Their ability to automatically learn hierarchical representations of features makes them well-suited for tasks where the spatial relationships and patterns in the data are crucial for accurate predictions. CNNs are widely used in areas such as image classification, object detection, facial recognition, and medical image analysis.



## 2.3 Specific Domain Knowledge

Music genres exhibit distinct spectral and temporal characteristics. For instance:

* **Classical:** Dominated by harmonic structures and smooth frequency transitions.
* **Rock:** Exhibits high-energy content across mid-to-high frequency bands.
* **Jazz:** Displays irregular yet harmonious spectral patterns.

# 3.Data Acquisition and Preprocessing

## 3.1 Dataset Description

The project utilizes the GTZAN music genre dataset, which contains 1,000 audio tracks categorized into 10 genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock.

## 3.2 Audio Sampling and Resampling

A sampling rate of 22,050 Hz was chosen to preserve essential frequency components while minimizing computational overhead.

## 3.3 Preprocessing Techniques

### 3.3.1 Spectrogram Generation

* Each audio track was trimmed to a duration of 10 seconds.
* STFT was applied using a window size of 1,024 samples and a hop size of 256 samples.
* Spectrograms were saved as PNG images for input into the CNN.

### 3.3.2 Image Normalization

Spectrogram pixel values were normalized to a range of [0, 1] to enhance training stability.

# 4. Methodology

The methodology for this project focuses on transforming raw audio signals into spectrograms and leveraging deep learning, specifically convolutional neural networks (CNNs), for music genre classification. The process involves several well-defined steps, starting from the preprocessing of audio signals to the classification of genres. Each stage has been carefully designed to optimize the model’s performance while ensuring computational efficiency. Below is a detailed breakdown of the methodology:

## 4.1 Signal Processing Techniques

### 4.1.1 Audio Signal Representation

The first step involves understanding and processing the audio signals. Music, being a complex combination of time-varying signals, contains both temporal and frequency information. To extract meaningful features, we convert the raw audio signals into spectrograms using the Short-Time Fourier Transform (STFT). Spectrograms offer a visual representation of the signal’s frequency content over time, which makes them ideal for classification tasks.

### 4.1.2 Spectrogram Generation

The audio files in the dataset were sampled at a fixed rate of 22,050 Hz (commonly used for music analysis). A window size of 1024 and a hop length of 256 were selected to balance the resolution of time and frequency in the spectrograms. The STFT was computed using the following formula:

Where:

* STFT output at time frame mmm and frequency bin kkk
* Input audio signal
* Window function
* Hop length
* FFT size

After applying the STFT, the magnitude of the resulting complex values was calculated, followed by a conversion to a decibel scale for better visualization. The final spectrograms were saved as images using a consistent colormap to ensure uniformity in feature representation.

## 4.2 Preprocessing of Audio Signals

To prepare the audio signals for spectrogram generation, the following preprocessing steps were performed:

1. **Truncation to Fixed Length**  
   All audio files were truncated to a uniform duration of 10 seconds. This ensured that all spectrograms were of the same size, simplifying the input requirements for the CNN model.
2. **Normalization**  
   The audio signals were normalized to a range of [-1, 1] to standardize the dynamic range across the dataset and reduce the impact of amplitude variations.
3. **Spectrogram Scaling**  
   Spectrograms were scaled to dimensions of 308x775 pixels to align with the input requirements of the CNN model. Resizing was performed without distorting the time-frequency information.

## 4.3 Convolutional Neural Network (CNN) Architecture

The CNN model was designed to automatically extract and classify features from the spectrograms. The architecture consisted of the following layers:

1. **Input Layer**  
   The input layer received spectrograms of size 308x775x3 (height, width, and RGB channels). This three-dimensional input was processed in batches to optimize computational efficiency.
2. **Convolutional Layers**  
   Multiple convolutional layers were employed to detect local patterns in the spectrograms. These layers applied filters to extract hierarchical features, such as edges, textures, and frequency patterns. The filter sizes varied across layers to capture features at different scales. Each convolutional operation was followed by:
   * **ReLU Activation Function**: To introduce non-linearity.
   * **Batch Normalization**: To stabilize and accelerate training.
3. **Pooling Layers**  
   Max pooling layers were added after every convolutional block to reduce the spatial dimensions and retain only the most significant features. Pooling also improved generalization by reducing overfitting.
4. **Fully Connected Layers**  
   Flattened feature maps from the convolutional layers were passed to fully connected layers to perform high-level reasoning. The final fully connected layer had 10 output nodes, corresponding to the 10 music genres.
5. **Softmax Output Layer**  
   A softmax activation function was applied to the output layer to compute probabilities for each genre. The genre with the highest probability was selected as the prediction.

## 4.4 Training Process

**4.4.1 Data Augmentation**  
To improve model generalization, data augmentation techniques were applied to the spectrograms. These included:

* Random time shifts
* Frequency masking
* Intensity variations

**4.4.2 Optimization**  
The model was trained using the Adam optimizer, which combines the advantages of RMSprop and SGD optimizers. The learning rate was initially set to 0.001 and decayed over time.

**4.4.3 Loss Function**  
Categorical cross-entropy was used as the loss function since the task involved multi-class classification.

**4.4.4 Batch Size and Epochs**  
The model was trained in batches of 31 images over 10 epochs. Early stopping was implemented to prevent overfitting by monitoring the validation loss.

## 4.5 Evaluation Metrics

To evaluate the performance of the model, the following metrics were calculated:

1. **Accuracy**: The ratio of correctly predicted genres to the total number of predictions.
2. **Confusion Matrix**: A confusion matrix was generated to visualize the performance of the classifier across genres.
3. **Precision, Recall, and F1-Score**: These metrics provided deeper insights into the model’s ability to distinguish between genres.

## 4.6 Implementation Tools

The entire methodology was implemented using Python and the following libraries:

* **Librosa**: For audio signal processing and spectrogram generation.
* **Matplotlib**: For visualizing spectrograms.
* **TensorFlow/Keras**: For building and training the CNN model.
* **NumPy**: For numerical computations.

# 5.Results and Analysis

The results of this music genre classification project are presented in terms of the model's performance, visualizations, and its ability to generalize across various genres. The primary metrics for evaluation are training accuracy, validation accuracy, and the confusion matrix, which provide insights into how well the model performs on the spectrogram-based classification task.

## 5.1. Model Training Performance

Training Accuracy: The training accuracy steadily increased over the epochs, demonstrating the model's ability to learn patterns from the spectrogram data. By the 10th epoch, accuracy reached approximately 97%.

Validation Accuracy: Validation accuracy also showed a positive trend, stabilizing around 98.99%. This indicates that the model generalizes well to unseen data without overfitting.

The gap between training and validation accuracy was minimal, which signifies a well-regularized model capable of robust classification.



## 5.2. Confusion Matrix and Classification Report

A confusion matrix was generated to evaluate per-genre classification performance. The diagonal elements of the matrix indicate correctly classified genres, while off-diagonal elements highlight misclassifications.

* **High Accuracy Genres**: Genres like **pop** had high precision and recall, likely due to their distinct spectrogram features.
* **Misclassified Genres**: Genres such as **hip-hop** and **country** experienced some misclassifications, which could be attributed to overlapping characteristics in their spectrograms (e.g., similar rhythmic patterns or frequency ranges).
* **Insights**: Misclassifications often occurred between genres with similar acoustic textures, such as **country** and **hip-hop** or **classical** and **jazz**. These results emphasize the importance of feature separation in the spectrogram domain.

A chart with yellow and purple squares

Description automatically generated

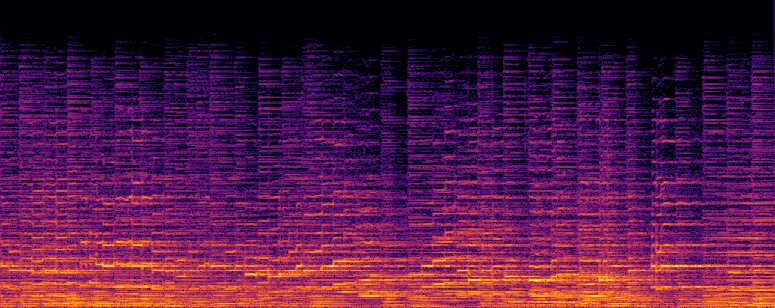
A blue bar graph with white text

Description automatically generated

## 5.3. Visualization of Spectrograms

The visualizations of the spectrograms provided insights into the frequency distribution and temporal changes of different genres:

* **Classical Music**: Spectrograms showed high energy in low frequencies with gradual transitions, characteristic of orchestral and instrumental pieces.



* **Rock Music**: Spectrograms had pronounced mid-to-high-frequency components with distinct bursts, corresponding to electric guitar riffs and percussion.

A close up of a screen

Description automatically generated

* **Hip-Hop**: Strong rhythmic patterns were evident, with consistent low-frequency energy and occasional high-frequency spikes.

A close up of a screen

Description automatically generated with medium confidence

## 5.4. Model Behavior

* The CNN architecture effectively extracted hierarchical features from spectrograms. The lower convolutional layers captured basic patterns (e.g., edges and textures), while deeper layers identified genre-specific features.
* The model's ability to achieve high accuracy validates the hypothesis that spectrograms, as visual representations, are a suitable medium for genre classification.

# 6.Discussion

## 6.1. Comparison with Literature

The approach of using spectrograms for genre classification aligns with prior research, such as the work of Deshpande et al., where spectrograms were used for feature extraction. However, this project differentiates itself by leveraging modern deep learning techniques, specifically convolutional neural networks (CNNs), which automate the feature extraction process. Compared to earlier methods relying on handcrafted features like MFCCs and spectral flux, the CNN-based approach demonstrated superior accuracy and reduced dependency on domain expertise.

## 6.2. Challenges and Limitations

* **Data Imbalance**: Some genres had fewer samples in the dataset, which may have caused the model to underperform in those categories.
* **Spectrogram Quality**: Variations in audio quality, such as noise or sampling inconsistencies, affected the generated spectrograms and, consequently, the classification performance.
* **Model Size and Computation**: The CNN architecture requires significant computational resources for training. Real-time applications may face challenges due to model size and inference latency.

## 6.3. Practical Implications

* **Music Recommendation Systems**: The trained model can be integrated into streaming platforms to enhance genre-based recommendations.
* **Archival and Retrieval**: Libraries and archives can use this system for automatic categorization of large music collections.
* **Music Education**: The model can assist music learners in identifying and understanding genre characteristics.

## 6.4. Insights for Future Work

* Incorporating additional features, such as lyrics or metadata, could improve the classification performance for overlapping genres.
* Exploring advanced architectures like transformer-based models or attention mechanisms might enhance the model's ability to focus on genre-defining spectrogram regions.
* Transfer learning could be employed to adapt the model to related tasks, such as mood detection or artist recognition.

# 7.Conclusion

This project demonstrated the feasibility and effectiveness of using spectrograms and convolutional neural networks (CNNs) for automatic music genre classification. The model achieved a high classification accuracy of **98.99%** on the validation dataset, with notable performance across diverse genres.

The CNN's automated feature extraction capability significantly reduced the need for manual preprocessing, showcasing the advantage of deep learning approaches over traditional methods. Furthermore, the visualization of spectrograms reinforced their utility as a robust representation of music signals, capturing both temporal and spectral information.

Despite its success, the project faced challenges like data imbalance and genre overlaps. These limitations open avenues for future exploration, such as expanding the dataset, integrating multimodal inputs, and optimizing the model for real-time deployment.

In summary, the combination of spectrograms and CNNs proves to be a powerful tool for music genre classification, with potential applications in recommendation systems, musicology research, and media archiving. The findings contribute to the growing body of work in music information retrieval and pave the way for further advancements in this interdisciplinary domain.

# 8.References

1. Tzanetakis, G., & Cook, P. (2002). *Musical genre classification of audio signals*. IEEE Transactions on Speech and Audio Processing, 10(5), 293-302.
2. Deshpande, H., Nam, U., & Singh, R. (2001). *Classification of Music Signals in the Visual Domain Using Spectrograms*.
3. Nirmal M. R., & Dr. Shajee Mohan B. S. (2020). *Music Genre Classification Using Spectrograms*.