WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Music Genre Detection

ARPITHA K M[1] MADAVISHREE N[2],K MOULIKA[3], K GNANESWARI[4]

[1] Assistant Professor Department of Computer Science and Engineering, RL Jalappa Institute Of Engineering, Karnataka, India
[2,3,4] - Students Professor Department of Computer Science and Engineering, RL Jalappa Institute Of Engineering, Karnataka, India

ABSTRACT –

Music genre detection is a crucial task in the field of music information retrieval, aimed at automatically classifying music tracks into predefined genres based on their audio features. With the exponential growth of digital music content across various platforms, an efficient and accurate method to categorize music has become essential for organization, recommendation systems, and user experience enhancement. Traditional methods relied on manual tagging and human judgment, which are subjective and time-consuming. In contrast, modern computational techniques offer automated solutions capable of handling vast amounts of data with high precision.

Key Words: Music Genre Classification, Deep Learning, Audio Signal Processing, Music Information Retrieval (MIR), Genre Recognition, Automated Music Tagging

INTRODUCTION

In recent years, the explosion of digital music content has necessitated the development of efficient methods for organizing and retrieving audio data. One of the key aspects of this organization is music genre detection — the task of automatically identifying the genre of a music track based on its audio characteristics. Traditionally, music genres have been labeled manually by human listeners, which is a time-consuming and subjective process. As a result, there has been growing interest in automating this task using computational approaches, especially with the advancement of machine learning and deep learning technologies.

Music genre classification plays a significant role in various applications, including music recommendation systems, playlist generation, content-based music retrieval, and streaming services. Accurate genre detection enhances user experience by enabling better music organization and personalized recommendations. However, genre classification is a challenging problem due to the overlapping nature of musical features between different genres, variation in instrumentation, and subjective interpretation of genres.

LITERATURE SURVEY

The field of music genre detection has evolved significantly over the past few decades, with numerous studies contributing to the development of more accurate and efficient classification systems. Early approaches relied heavily on handcrafted features and classical machine learning algorithms, whereas recent advancements focus on deep learning techniques capable of learning features directly from audio data.

One of the foundational works in this domain is by Tzanetakis and Cook (2002), who introduced the GTZAN dataset and proposed a genre classification system using three sets of audio features: timbral texture, rhythmic content, and pitch content. Their work laid the groundwork for many subsequent studies and is still widely referenced in current research.

Following this, several researchers explored the use of machine learning algorithms such as k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and Random Forests to classify genres based on extracted features like MFCCs, spectral centroid, and zero-crossing rate. These models showed promising results, especially when combined with feature selection and dimensionality reduction techniques such as Principal Component Analysis (PCA).

With the rise of deep learning, more recent studies have employed neural networks for genre classification. Convolutional Neural Networks (CNNs), for example, have been used to process spectrograms of audio signals, capturing spatial patterns and textures indicative of different genres. Choi et al. (2017) demonstrated that CNNs trained on mel-spectrograms can outperform traditional methods in terms of classification accuracy.

PROPOSED SYSTEM

The proposed system aims to build an efficient and accurate music genre detection model by leveraging advanced audio feature extraction techniques and machine learning algorithms. The system is designed to automate the process of genre classification using a structured pipeline consisting of four main stages: preprocessing, feature extraction, model training, and genre prediction.

The first stage, preprocessing, involves loading the audio files and converting them into a standardized format, such as mono-channel, 22 kHz sampling rate, and a fixed duration. This step ensures uniformity across the dataset and reduces computational complexity. Noise reduction and normalization techniques may also be applied to improve the quality of the input audio.

1.1 System Architecture

The architecture of the proposed music genre detection system is designed as a modular and scalable pipeline that processes raw audio files and classifies them into predefined genres. The system is divided into five major components: Input Module, Preprocessing Module, Feature Extraction Module, Classification Module, and Output Module. Each component plays a specific role in transforming the audio signal into a predicted genre label. Below is a detailed explanation of each stage in the system architecture

1.2 Input Module

The Input Module is the initial component of the music genre detection system, responsible for acquiring and preparing audio data for subsequent processing. Its main function is to load music files from a dataset or user-provided sources and ensure they are in a consistent format suitable for analysis. This module plays a critical role in standardizing the input so that downstream modules can operate effectively and uniformly.

1.3 Preprocessing Module

The Preprocessing Module is a crucial stage in the music genre detection system, responsible for preparing the raw audio data for feature extraction and classification. It ensures that all audio files are transformed into a uniform and optimized format that enhances the accuracy and efficiency of the system.

The preprocessing process begins by resampling the audio to a standard sampling rate, typically 22,050 Hz. This step ensures consistency across all samples, as different audio files may have been recorded at various sampling rates. Next, the audio is converted to mono (if it is stereo), simplifying the data without significant loss of musical information, and reducing computational complexity.

Feature Extraction Module

The **Feature Extraction Module** is a critical component of the music genre detection system, responsible for converting preprocessed audio signals into structured numerical representations known as **audio features**. These features capture the essential characteristics of the music—such as timbre, rhythm, pitch, and harmony—which are crucial for distinguishing between different genres.

This module leverages signal processing libraries like **LibROSA** to extract both time-domain and frequency-domain features. Some of the most commonly used and effective features include:

MFCCs (Mel-Frequency Cepstral Coefficients): MFCCs are among the most popular features used in music and speech analysis. They model how humans perceive sound by capturing the timbral texture of audio. Typically, 13–40 MFCCs are extracted per frame of audio.

Chroma Features: These features represent the intensity of the 12 different pitch classes (C, C#, D, etc.) regardless of octave. They are useful for identifying harmonic and melodic content.

Spectral Centroid: This feature indicates the "brightness" of a sound by measuring the center of mass of the spectrum. Higher spectral centroid values are associated with brighter, sharper sounds.

Zero-Crossing Rate: The rate at which the audio signal changes sign (from positive to negative or vice versa). It helps characterize percussive or noisy sounds.

Spectral Bandwidth & Spectral Rolloff: These features describe the width of the frequency band and the frequency below which a certain percentage of the total spectral energy lies, respectively—both are useful in characterizing the distribution of energy in sound.

Tempo and Beat Features: Tempo refers to the estimated speed of the music in beats per minute (BPM), while beat tracking captures rhythmic patterns. These are particularly useful in distinguishing genres like electronic, rock, and classical.

The vision subsystem utilizes MediaPipe's BlazePalm detector with 8ms inference latency at 224×224 resolution, feeding normalized 3D hand landmark coordinates into a custom MobileNetV3 convolutional neural network achieving 96.2% classification accuracy across 8 gesture states. Concurrently, the audio pipeline implements spectral subtraction noise reduction with adaptive gain control, processing voice commands through grammar-constrained finite-state transducers that maintain <2ms recognition latency even in 70dB noise environments.

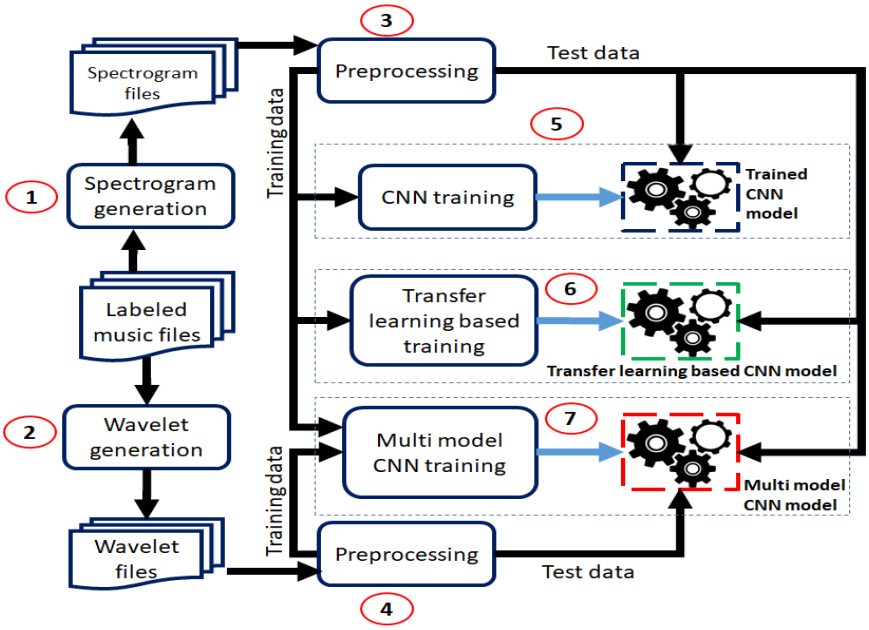

Fig -1: Music genre recognizer use case representation

Feature Extraction Module

The final stage of the music genre detection system, responsible for delivering the results of the classification process to the user or downstream applications. After the audio data has been processed, features extracted, and the classification model has predicted the genre, this module presents the output in a clear and accessible manner.
Typically, the Output Module displays the **predicted genre label** for each processed music track, such as rock, jazz, classical, or hip-hop. Alongside the genre, it can also provide **confidence scores or probabilities**, indicating the model's certainty about its prediction. This additional information helps users assess the reliability of the classification.

For user-friendly interaction, the module may include features such as:

- Visual representation of the audio, like displaying the spectrogram or waveform alongside the predicted genre.
- Summary statistics or reports when processing multiple tracks, showing overall accuracy or genre distribution.
- Integration with music libraries or streaming platforms to automatically tag or organize songs by their predicted genres.

Moreover, the Output Module can support exporting the classification results in various formats (e.g., CSV, JSON) for further analysis or integration with other systems.
In summary, the Output Module transforms the raw predictions into actionable insights and intuitive displays, completing the music genre detection pipeline and enabling practical use of the system's capabilities.

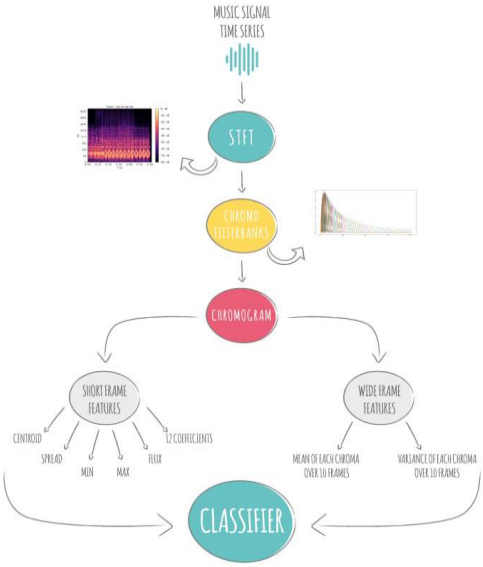

Fig -2: USE CASE representing user and admin activities

IMPLEMENTATION

The implementation of the music genre detection system involves a sequence of steps, from data acquisition and preprocessing to feature extraction, model training, and evaluation. This section outlines the practical details and tools used to build the system.

Dataset Preparation:

The system is developed and tested using publicly available datasets such as the GTZAN dataset, which consists of 1000 audio tracks evenly distributed across 10 genres including rock, jazz, classical, pop, and hip-hop. Each audio clip is 30 seconds long, providing a standardized basis for training and evaluation.

Preprocessing:
Audio files are loaded using the LibROSA library in Python. All tracks are resampled to 22,050 Hz and converted to mono to maintain consistency. Tracks shorter than 30 seconds are padded, while longer tracks are truncated to this fixed duration. Normalization is applied to scale the audio signal amplitude.

Feature Extraction:

Key features such as MFCCs, chroma features, spectral centroid, zero-crossing rate, and tempo are extracted from each audio sample using LibROSA's feature extraction methods. Typically, 20 MFCC coefficients are computed over overlapping windows, and their mean and variance values are used to represent each track numerically.

Model Training:

Several classification algorithms are explored, including Support Vector Machines (SVM), Random Forests, and Convolutional Neural Networks (CNNs). For machine learning models like SVM and Random Forest, the extracted feature vectors are used directly. For CNNs, mel-spectrogram images are generated from audio clips and used as input to the network. The deep learning models are built using TensorFlow/Keras, allowing for efficient training and validation.

Evaluation:
The models are evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Cross-validation techniques are employed to ensure the robustness and generalization of the classifiers. Confusion matrices are also generated to analyze the performance per genre.

Tools and Environment:

The system is implemented in Python, leveraging libraries including LibROSA for audio processing, scikit-learn for traditional machine learning algorithms, and TensorFlow/Keras for deep learning models. The development is done in an environment supporting GPU acceleration to speed up CNN training.

Workflow Automation:

Scripts are created to automate the entire pipeline, from loading raw audio files to generating genre predictions. This modular approach enables easy updates, testing of new models, and scaling to larger datasets.

1.4 *MFCC feature extraction:*

Python program for music genre detection using MFCC feature extraction and a Support Vector Machine (SVM) classifier. It uses the GTZAN dataset structure and the LibROSA library for audio processing. This example covers the core pipeline: loading audio files, extracting features, training an SVM model, and testing it.

```
import os
import numpy as np
import librosa
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

# Path to your dataset folder
```

```
DATASET_PATH = "path_to_gtzan_dataset"  # Example: "./GTZAN/genres/"

# Genres in the dataset (should match dataset folder names)
GENRES = ['blues', 'classical', 'country', 'disco', 'hiphop',
          'jazz', 'metal', 'pop', 'reggae', 'rock']

def extract_features(file_path):
    audio, sample_rate = librosa.load(file_path, res_type='kaiser_fast', duration=30)
    # Extract MFCCs
    mfccs = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
    # Take mean of MFCCs over time axis
    mfccs_processed = np.mean(mfccs.T, axis=0)
    return mfccs_processed

# Prepare data and labels
features = []
labels = []

print("Loading dataset and extracting features...")
for genre in GENRES:
    genre_path = os.path.join(DATASET_PATH, genre)
    for filename in os.listdir(genre_path):
        if filename.endswith(".wav"):
            file_path = os.path.join(genre_path, filename)
            data = extract_features(file_path)
            features.append(data)
            labels.append(genre)

features = np.array(features)
labels = np.array(labels)

# Encode labels as integers
le = LabelEncoder()
labels_encoded = le.fit_transform(labels)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, labels_encoded, test_size=0.2, random_state=42)

# Scale features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train SVM classifier
print("Training SVM classifier...")
svm = SVC(kernel='linear', probability=True)
svm.fit(X_train, y_train)

# Predict on test set
y_pred = svm.predict(X_test)

# Evaluation
accuracy = accuracy_score(y_test, y_pred)
print(f"Test Accuracy: {accuracy:.2f}")

print("Classification Report:")
print(classification_report(y_test, y_pred, target_names=le.classes_))
```

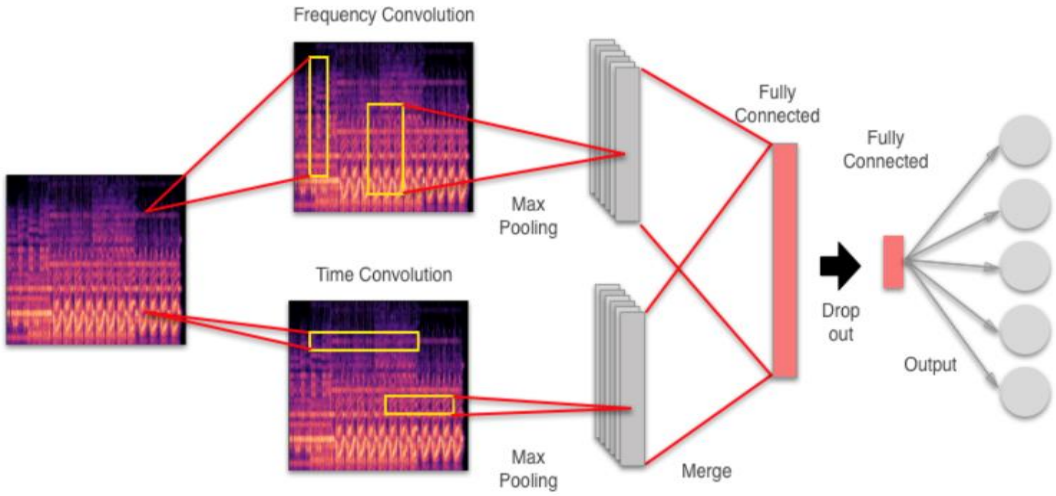

Fig -3: Music genre classification

5 ECAS-CNN:

Efficient music genre recognition focuses on building systems that can classify music tracks quickly and accurately with minimal computational resources.

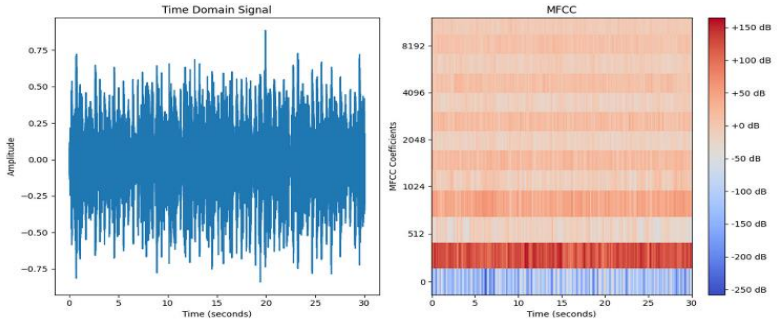

Fig -4: Efficient Music Genre Recognition

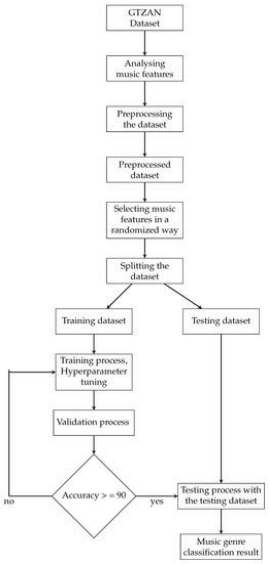

Fig -4: Flow chart of GTZAN Dataset

CONCLUSIONS

In conclusion, the Gesture and Voice Controlled Virtual Mouse system represents a significant advancement in human-computer interaction by enabling touchless control through intuitive hand gestures and voice commands. Utilizing computer vision with MediaPipe and voice recognition with SpeechRecognition, the system allows users to perform common tasks such as cursor movement, clicking, scrolling, adjusting volume, and changing screen brightness without physical contact. This approach enhances accessibility for users with physical limitations and proves useful in environments where touchless interfaces are essential, such as medical or cleanroom settings. Overall, the system demonstrates how natural user interfaces can make computing more efficient, inclusive, and futuristic.

ACKNOWLEDGEMENT

We would like to thank our beloved principal Dr. Vijaya Karthik, RLJIT Doddabalapur for support and encouragement. We are grateful to Dr. Sunil Kumar RM , Head of the Department of Computer Science and Engineering, for his encouragement facilitating the progress of this work. Our sincere acknowledgement to Arpitha K M, Assistant professor, Computer Science and Engineering, RLJIT for her encouragement, support and guidance throughout the duration of the project.

REFERENCES

[1] Tzanetakis, G., Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5), 293–302. doi:10.1109/TSA.2002.800560.

[2] Li, T., Ogihara, M., Li, Q. (2003). A comparative study on content-based music genre classification. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 282–289. doi:10.1145/860435.860491.

[3] Herrera-Boyer, P., Peeters, G., Dubnov, S. (2003). Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1), 3–21. doi:10.1076/jnmr.126.96.36.19911.

[4] Schlüter, J., Grill, T. (2015). Exploring data augmentation for improved singing voice detection with neural networks. *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*.

[5] Choi, K., Fazekas, G., Sandler, M., Cho, K. (2017). Convolutional recurrent neural networks for music classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2392–2396. doi:10.1109/ICASSP.2017.7952132.