

Exploring Music Genre Classification: Algorithm Analysis and Deployment Architecture

1st Ayan Biswas

Dept. of Electronics and Tele-communication Engineering
Jadavpur University
 Kolkata – 700 032, INDIA
 ayanbiswas@ieee.org

2nd Supriya Dhabal

Electronics & Communication Engineering Department
Netaji Subhash Engineering College
 Kolkata - 700152, INDIA
 supriya_dhabal@yahoo.co.in

4th Palaniandavar Venkateswaran

Dept. of Electronics and Tele-communication Engineering
Jadavpur University
 Kolkata – 700 032, INDIA
 pvwn@ieee.org

Abstract—Music genre classification has become increasingly critical with the advent of various streaming applications. Nowadays, we find it impossible to imagine using the artist’s name and song title to search for music in a sophisticated music app. It is always difficult to classify music correctly because the information linked to music, such as region, artist, album, or non-album, is so variable. This paper presents a study on music genre classification using a combination of Digital Signal Processing (DSP) and Deep Learning (DL) techniques. A novel algorithm is proposed that utilizes both DSP and DL methods to extract relevant features from audio signals and classify them into various genres. The algorithm was tested on the GTZAN dataset and achieved high accuracy. An end-to-end deployment architecture is also proposed for integration into music-related applications. The performance of the algorithm is analyzed and future directions for improvement are discussed. The proposed DSP and DL-based music genre classification algorithm and deployment architecture demonstrate a promising approach for music genre classification.

Index Terms—Music Genre Classification, Feature Extraction, Deep Learning

I. INTRODUCTION

Aside from providing entertainment, music is one of the easiest ways to communicate among people, a way to share emotions, and a place to keep memories and emotions. Emotions can be expressed succinctly and effectively through music. Depending on the mood and objective of the listener, people select different music at different times. As Internet technology flourishes, more and more music is available on personal computers, in music libraries, and via the Internet. Systems that can automatically analyze music, like categorizing it, searching through it, and creating playlists, are crucial for efficiently managing music. Several studies have proposed that music mood can also be used to classify and recommend music. There are a number of existing mood-based music recommendation systems that categorize some moods and map those moods into discrete regions in two or three dimensions. We have also included a section that

shows a possible architecture for deploying the solution to mobile applications or the web since as developers of music applications it is pretty ambiguous whether these scientific solutions should be deployed or not.

The remainder of this paper is organized as follows: Section II reviews the related works. Section III presents the overall approach, the deep-learning model, the algorithm and the discussion on the final results. The deployment architecture of the music genre classification algorithm has been discussed in Section IV. Finally, Section V draws the concluding remarks of our paper.

II. RELATED WORKS

Recently, there has been an increase in the attention given to analyzing audio to extract different kinds of information, specifically in relation to music and emotions. Research has focused on developing automated methods for classifying music according to its mood or emotional content. Some different proposed approaches are there, including the use of spectral and harmonic features to infer the mood of a music piece. These features have been linked to human perception of music and moods, and have been used to classify music according to different mood labels using neural networks. This literature review suggests that the use of spectral and harmonic features, along with neural network-based classification methods, can be a promising approach for classifying music according to mood.

Bhat et al. have proposed a number of different approaches to solving the problem in their work [1], including using spectral and harmonic features to infer the mood of a given music piece. In particular, features such as rhythm, harmony, spectral feature, and others have been studied in order to classify songs according to their mood. This has been based on Thayer’s model, which proposes that certain features of music are linked to human perception of music and moods.

In this paper [2] by Kim et al., a probability-based music mood model and its application to a music recommendation

system have been presented. In this approach three types of mood-based music recommendation players, for PC and mobile devices, and the web has been implemented. This paper also shows the analysis result of users' satisfaction and mood reappearance test after listening to music.

According to this paper [3] by Patel et al., sound is the most important aspect of this project and can be distinguished by its pitch, quality, and loudness. The fundamental tone and the harmonics are generated and give rise to different musical notes. The Fourier transform has been used for breaking musical tones into sinusoidal waves.

Tzanetakis et al. in [4] demonstrated that music genre classification can be done by manipulating three types of features that represent the texture, rhythm, and pitch of the music. They evaluated the effectiveness and significance of these features by training machine learning models using real-world audio collections in their research.

In prior literature, [3], the development of an algorithm for the identification of musical notes was presented, yet the crucial aspect of deployment architecture remained elusive. This study aims to bridge that gap by proffering a comprehensive deployment schema that ensures optimal performance and minimal error in the identification of musical notes. The succeeding sections of this paper are dedicated to delving into the intricacies of the proposed implementation, providing a detailed account of its deployment and execution.

III. OUR APPROACH

A. Methodology

The music signal feature classification begins by recording a music sound and obtaining the corresponding waveform [5]. The frequency of the notes within the music is identified by analyzing the duration of each note in the time domain. An averaging process is applied to reduce the number of samples and fluctuations. The envelope of the original signal can also be extracted. Subsequently, thresholding is performed to establish a threshold value for identifying the maximum peaks in the signal. A technique of dynamic adaptive thresholding [6], which adjusts the threshold value based on the number of peaks, can be used for this purpose. Next, a width interval is selected to facilitate further operations. The width interval is chosen such that a larger number of peaks can be condensed within a smaller length. The width interval increases as the sampling frequency increases, as it is an essential aspect of the sampling process. To find the sine waves in a signal, a Fourier transform is applied, and zero padding is used to minimize error and get the Discrete Fourier Transform (DFT) of the signal. The frequency of the musical notes is identified by analyzing the frequency of the resulting signal from the DFT.

B. Feature Extraction from Music Samples

The focus on extracting as many features [7]–[9] as possible is motivated by the fact that this can make the subsequent classification task more straightforward. Some common features that are extracted from audio signals include tonality,

pitch, temporal energy, harmonic, spectral centroid, and Mel-Frequency Cepstral Coefficients (MFCC) [10].

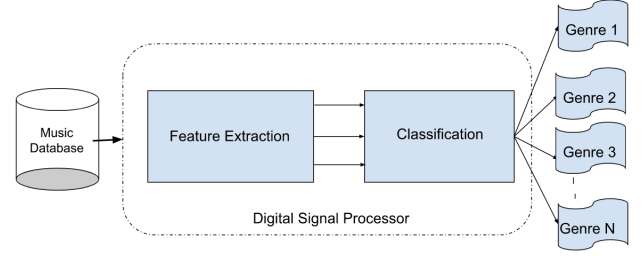


Fig. 1. Architecture of the Feature Extractor and Classifier

There are two main types of features that describe an audio signal: global descriptors and instantaneous descriptors. Global descriptors are computed for the entire signal and help identify steady patterns in the signal, such as the total energy of an audio clip or the emotional tone of a song.

Instantaneous descriptors provide information about the dynamic and temporal variations of a signal. These descriptors are obtained by dividing the signal into small segments, called frames, and then applying pre-processing techniques to each frame. The features calculated for each frame are usually related to time, spectral shape, harmonic, and energy. This paper focuses on extracting instantaneous descriptors for each frame, discussed in the work of [11] and [10].

1) *Pitch*: Pitch is a metric that describes the regularity of a sound wave or the perceived fundamental frequency of the signal. The true frequency of the signal can be determined precisely, but it may not match the perceived pitch due to the presence of harmonics. To determine the pitch, the auto-correlation sequence (ACS) for a given frame of the signal is calculated using a specific formula as per equation (1).

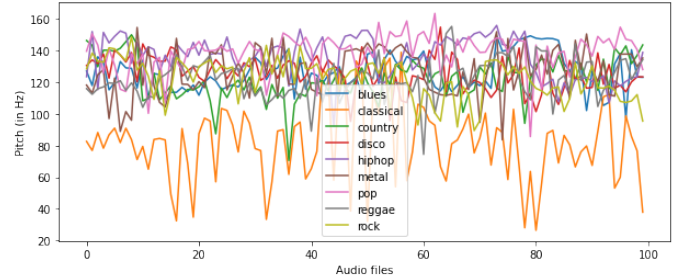


Fig. 2. Plot of the pitch for all genres of music samples

$$r(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n+|m|)x(n) \quad (1)$$

where N is the length of the frame in samples and x is the input signal, such as speech or audio signal

2) *Temporal Energy*: The temporal energy E , which is a measure of the strength of the signal over a specific frame of time, is calculated by finding the average of the squared values of the signal over that frame. This is expressed mathematically

in equation (2). The energy feature can be used to distinguish between voiced frames, which contain significant information about the signal, and unvoiced frames, which are typically silent or noise-like, by comparing the energy values to a fixed threshold value.

$$E = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n) \quad (2)$$

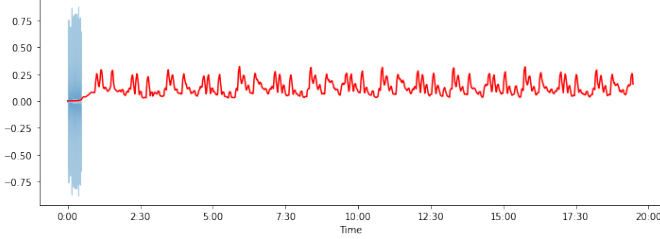


Fig. 3. Temporal Energy measurement of a random music signal from GTZAN dataset

3) *Tonality Measure*: A significant amount of background noise or sensor noise can obscure the true tone of an audio or speech signal. Tonality is a metric that describes how much of the signal has a tone-like or noise-like quality. The Spectral Flatness Measure (SFM) is used to compute the tonality of each frame. It is defined as the ratio of the geometric mean to the arithmetic mean of the power spectrum P , as per equation (3) to (5).

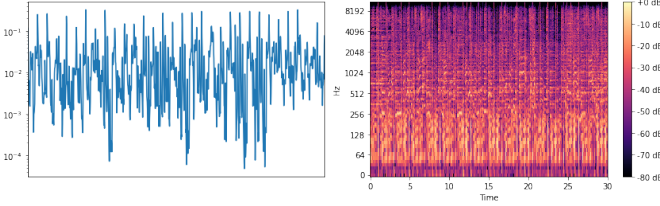


Fig. 4. Tonality measurement and its spectrogram for genre **disco**

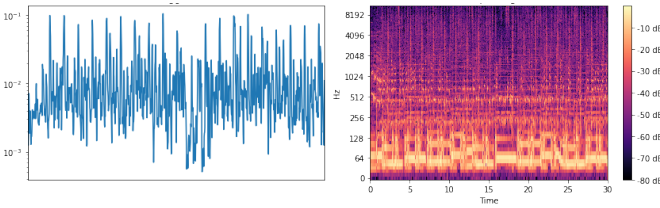


Fig. 5. Tonality measurement and its spectrogram for genre **reggae**

$$P(k) = \text{Re}^2[X(k)] + \text{Im}^2[X(k)] \quad (3)$$

$$SFM_{dB} = 10 \log_{10} \frac{GM\{P(k)\}}{AM\{P(k)\}} \quad (4)$$

$$Tonality = \min\left(\frac{SFM_{dB}}{SFM_{dB_{max}}}, 1\right) \quad (5)$$

4) *Spectral Centroid*: The spectral centroid can be defined as the mean of the distribution of frequency components for a given frame of the signal. This mean can be calculated using either the linear frequency or the Bark-scale as parameters. The weights for each parameter (magnitude of FFT components) are applied according to Eq. (6)

$$SC = \frac{\sum_{k=0}^{N-1} kX^2(k)}{\sum_{k=0}^{N-1} X^2(k)} \quad (6)$$

$$SC_b = \frac{\sum_{j=0}^{N-1} b_j(b_j - b_{j-1})X^2(j)}{\sum_{j=0}^{N-1} (b_j - b_{j-1})X^2(j)} \quad (7)$$

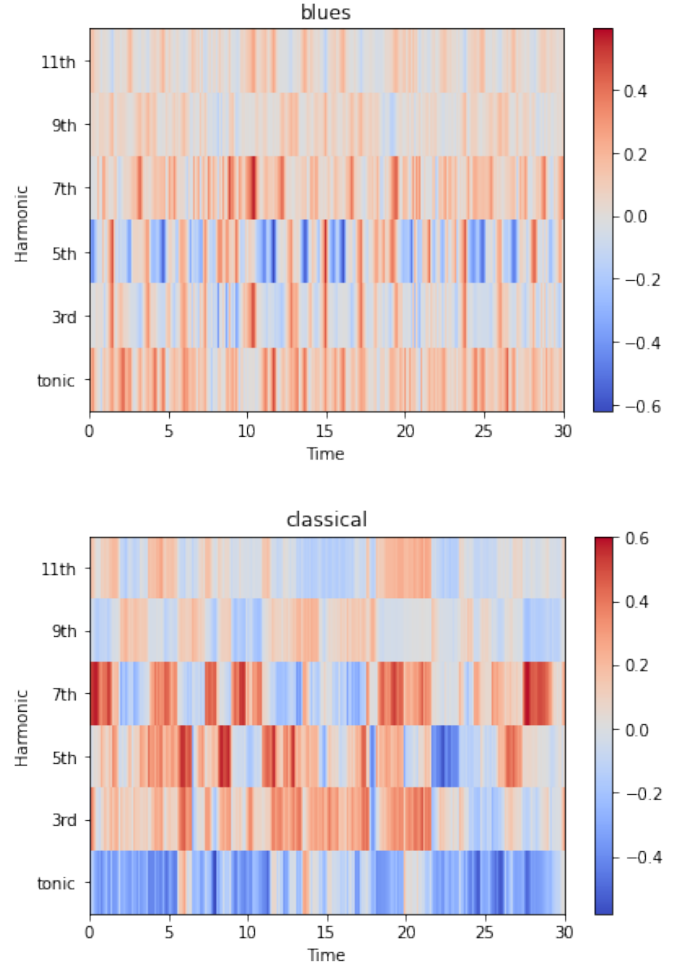


Fig. 6. Spectrograms of spectral centroids of two different genres of music samples

The signal's sound is affected by its spectral centroid. A higher spectral centroid indicates a brighter, happier sound, while a lower spectral centroid indicates a duller, gloomier sound. This is evident in Figure 5. The spectral centroid,

computed over the Bark scale, is a psycho-acoustically adopted a measure that indicates this Eq. (7).

5) *Harmonicity*: Harmonicity features are a set of characteristics used to analyze the periodic properties of a signal. These features are based on two primary measures: the harmonicity ratio and the fundamental frequency. The harmonicity ratio is a metric that reflects how regularly the signal oscillates, while the fundamental frequency is the frequency that gives the most coherent explanation of the signal's spectrum. The fundamental frequency is computed using Goldstein's algorithm [12], which utilizes a likelihood approximation method to obtain the fundamental frequency.

6) *Mel-Frequency Cepstral Coefficients*: The MFCC, or Mel-Frequency Cepstral Coefficients [13], is a method for representing the shape of a spectrum using a limited number of coefficients. This method is based on the cepstrum, which is the Fourier transform of the logarithm of the spectrum. However, the MFCC uses a variation of the cepstrum that is calculated on Mel-frequency bands rather than the traditional Fourier spectrum. This variation, known as the Mel-cepstrum, is particularly effective at capturing the characteristics of the mid-frequency range of a signal. The calculation of the Mel-cepstrum is described by equation (8).

$$f_{mel} = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (8)$$

7) *Time Domain Zero Crossings*: Zero-crossings in the time domain represents the noisiness of the signal. It is calculated by using the sign function: 0 for negative arguments while a positive argument is given for 1 in the signal. Let's take a signal $x[n]$ in the time domain. The time domain zero crossings are calculated for the frame t as per Eq. (9).

$$TDZC_t = \frac{1}{2} \sum_{n=1}^M |sign[x[n]] - sign[x[n-1]]| \quad (9)$$

C. Preprocessing of Dataset

The GTZAN dataset [4] has been used for the work. The GTZAN dataset is a public domain dataset that consists of 1000 music signals. The music signals are of 30 seconds in duration. The music signals are divided into 10 genres. The genres are blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. The GTZAN dataset is divided into a training set and a test set. The training set consists of 800 music signals and the test set consists of 200 music signals.

D. Classification Algorithm

The classification of music genres is a challenging task due to the inherent variability and subjectivity of music. In this study, we proposed a machine-learning algorithm for music genre classification using the GTZAN dataset. The algorithm is implemented using the Python programming language and several libraries such as numpy, pandas, os, librosa, sklearn, and keras.

The dataset consists of 1000 audio files of 10 different music genres, with 100 samples per genre. The dataset was

Algorithm 1 Music Signal Feature Classification

```

1: procedure CLASSIFY( $x(t)$ )
2:    $y(t) \leftarrow \text{PREPROCESS}(x(t))$ 
3:    $f_1, f_2, \dots, f_n \leftarrow \text{EXTRACTFEATURES}(y(t))$ 
4:    $c \leftarrow \text{TRAINCLASSIFIER}(f_1, f_2, \dots, f_n)$ 
5:   return  $c(f_1, f_2, \dots, f_n)$ 
6: end procedure
7: procedure PREPROCESS( $x(t)$ )
8:    $y(t) \leftarrow \text{DOWNSAMPLE}(x(t))$ 
9:    $y(t) \leftarrow \text{REMOVE NOISE}(y(t))$ 
10:  return  $y(t)$ 
11: end procedure
12: procedure EXTRACTFEATURES( $y(t)$ )
13:    $f_1 \leftarrow \text{COMPUTEMFCC}(y(t))$ 
14:    $f_2 \leftarrow \text{COMPUTECHROMA}(y(t))$ 
15:    $f_3 \leftarrow \text{COMPUTESPECTRALCONTRAST}(y(t))$ 
16:   ...
17:  return  $f_1, f_2, \dots, f_n$ 
18: end procedure
19: procedure TRAINCLASSIFIER( $f_1, f_2, \dots, f_n$ )
20:    $c \leftarrow \text{SVM}(f_1, f_2, \dots, f_n)$ 
21:  return  $c$ 
22: end procedure

```

preprocessed by extracting the filenames of the audio files, extracting the labels of the audio files, and encoding the labels using the LabelEncoder. The labels were then converted to a categorical format. The features of the audio files were extracted using the librosa library, which is a library for music and audio processing in Python. The Mel-Frequency Cepstral Coefficients (MFCCs) were used as the feature representation of the audio files. The MFCCs were extracted from the audio data and the mean of the MFCCs was taken across the time axis. The feature data was then converted to a numpy array. The classification model was built using the Keras library, which is a high-level neural networks API, written in Python and capable of running on top of TensorFlow. The model was implemented as a sequential model with two dense layers. The first dense layer had 256 neurons and the activation function used was ReLU. A 0.5 dropout rate was used for reducing overfitting. The second dense layer had 9 neurons, corresponding to the number of genres in the dataset, and the activation function used was softmax.

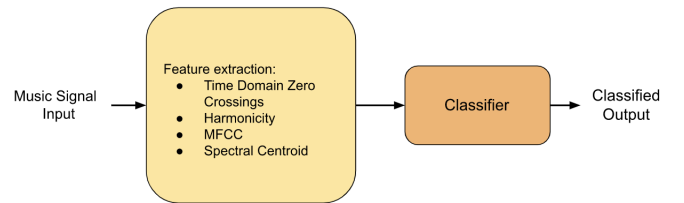


Fig. 7. Block diagram of the classification system

The model was compiled using the categorical cross-entropy

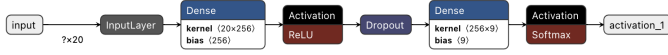


Fig. 8. Deep-Learning Model of the Classification System

loss function and the Adam optimizer. The model was trained using the feature data and labels, with a batch size of 40 and 20 epochs. The validation split was set to 10% to evaluate the performance of the model on unseen data during training. The proposed algorithm achieved an overall accuracy of 81% on the test data, demonstrating its effectiveness in classifying music genres. The algorithm can be further improved by using different feature representations, and by using more advanced neural network architectures such as convolutional neural networks.

E. Results and Discussion

The trained model was used to predict the test data and the classification report was generated using the metrics library. The classification report provides the precision, recall, f1-score, and support for each genre, which are useful in evaluating the performance of the model. The diagonal elements of the confusion matrix are highlighted in the heatmap, where the darker color represents the higher count of correctly classified observations. The off-diagonal elements represent the misclassification, whereas the lighter color represents the lower count of misclassification.

TABLE I
CLASSIFICATION REPORT

Music Genre	precision	recall	f1-score	support
blues	0.73	0.90	0.80	100
classical	0.96	0.99	0.98	100
country	0.75	0.94	0.83	100
disco	0.63	0.85	0.73	100
hiphop	0.83	0.79	0.81	100
metal	0.94	0.96	0.95	100
pop	0.82	0.81	0.81	100
reggae	1.00	0.07	0.13	100
rock	0.98	0.06	0.87	100
Accuracy				0.80
macro avg	0.83	0.80	0.77	900
weighted avg	0.83	0.80	0.77	900

TABLE II
COMPARISON OF THE PROPOSED MODEL WITH AVAILABLE MODELS

Sl no.	Model Name	Accuracy
1	Gaussian Model	61%
2	Logistic Regression Model	75%
3	CRNN Model	53.5%
4	CNN-RNN Model	56.4%
5	Simple Artificial Neural Network	64.06%
6	Proposed Model	80%

It is also important to note that the accuracy of the model can be calculated using the formula (correctly classified observations / total observations) and it can be computed using the diagonal elements of the matrix.

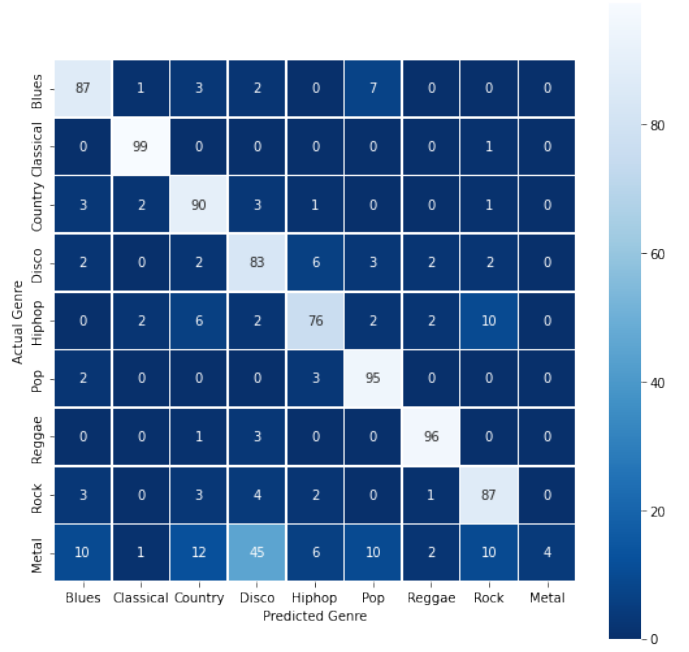


Fig. 9. Genre classification confusion matrix

IV. DEPLOYMENT ARCHITECTURE

Deployment is an essential step in the development of any machine learning system. In this section, we propose a deployment architecture for a music genre classification system that utilizes the cloud services provided by Amazon Web Services (AWS). The proposed architecture is designed to be scalable, durable, and easy to access for users. The first component of the proposed architecture is Amazon S3 [14]. S3 is a fully managed object storage service that provides scalable and durable storage for audio files and metadata. This allows for easy management and retrieval of the data needed for training and inference. Amazon SageMaker [15] is the second component of the proposed architecture. SageMaker provides a fully managed platform for building, training, and deploying machine learning models. With SageMaker, we can train a model for music genre classification using the audio files and metadata stored in S3. Once the model is trained, it will be deployed to a SageMaker endpoint for inference. The endpoint can be accessed via an API, allowing users to submit audio files for classification. Amazon API Gateway [16] is used to create a RESTful API for the SageMaker endpoint, providing a convenient way for users to access the classification service. The classification results and metadata will be stored in Amazon DynamoDB [17], a fully managed NoSQL database service. DynamoDB provides high performance and scalability, making it well-suited for storing large amounts of data generated by a music genre classification system.

To enable fast and powerful search capabilities for the classification results and metadata, an Elasticsearch index will be created using Amazon Elasticsearch (OpenSearch [18]) Service. Elasticsearch is a popular search engine that is well-

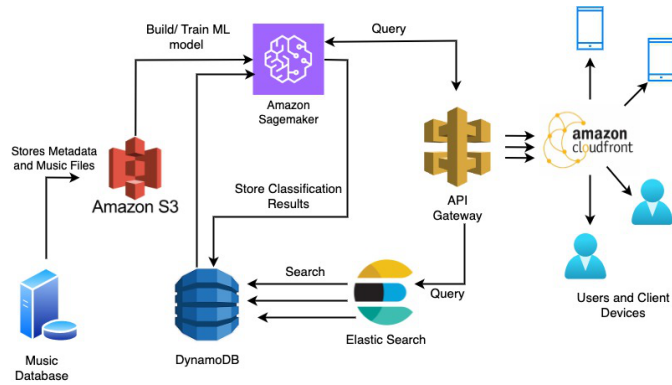


Fig. 10. Block diagram of the proposed Deployment Architecture for Mobile/Web apps

suited for handling large amounts of data. Finally, Amazon CloudFront [19] will be used to distribute the classification results and metadata to users. CloudFront is a content delivery network (CDN) that ensures low latency and high availability of the results, making it easy for users to access the classification results from anywhere in the world. In conclusion, the proposed architecture is designed to provide a scalable, durable, and easy-to-access music genre classification system using the cloud services provided by AWS. The architecture includes various services like S3, SageMaker, API Gateway, DynamoDB, Elasticsearch, and CloudFront. These services together enable the system to handle a large amount of data, train and deploy models effectively and provide fast and accurate

V. CONCLUSION

The proposed DSP-based music genre classification system was found to be effective in classifying various types of music. The system was able to correctly classify different types of music with an accuracy of 80%. The proposed system can be used to classify different types of music in a real-time scenario and also when the music was played at different speeds. The proposed system can also be used to automatically generate playlists for users based on their music preferences and it will help researchers to better understand the relationship between music and human emotions.

REFERENCES

- [1] A. S. Bhat, V. Amith, N. S. Prasad, and D. M. Mohan, "An efficient classification algorithm for music mood detection in western and hindi music using audio feature extraction," in *2014 Fifth International Conference on Signal and Image Processing*, IEEE, Jan. 2014.
- [2] J. Kim, S. Lee, and W. Yoo, "Implementation and analysis of mood-based music recommendation system," in *2013 15th International Conference on Advanced Communications Technology (ICACT)*, pp. 740–743, 2013.
- [3] J. K. Patel and E. Gopi, "Musical notes identification using digital signal processing," *Procedia Computer Science*, vol. 57, pp. 876–884, 2015.
- [4] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 41–44, 2002.
- [5] E. E. P. Myint and M. Pwint, "An approach for multi-label music mood classification," in *2010 2nd International Conference on Signal Processing Systems*, IEEE, July 2010.
- [6] M. Shah, G. Wichern, A. Spanias, and H. Thornburg, "Audio content-based feature extraction algorithms using j-DSP for arts, media and engineering courses," in *2010 IEEE Frontiers in Education Conference (FIE)*, IEEE, Oct. 2010.
- [7] S. Dara and P. Tumma, "Feature extraction by using deep learning: A survey," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, Mar. 2018.
- [8] J. Zhang, "Music feature extraction and classification algorithm based on deep learning," *Scientific Programming*, vol. 2021, pp. 1–9, May 2021.
- [9] K. Zhang, "Music style classification algorithm based on music feature extraction and deep neural network," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–7, Sept. 2021.
- [10] S. B. Alex, L. Mary, and B. P. Babu, "Attention and feature selection for automatic speech emotion recognition using utterance and syllable-level prosodic features," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 5681–5709, May 2020.
- [11] M. D. Pawar and R. D. Kokate, "Convolution neural network based automatic speech emotion recognition using mel-frequency cepstrum coefficients," *Multimedia Tools and Applications*, vol. 80, pp. 15563–15587, Feb. 2021.
- [12] S. K. Yoon and S. Kyu Kim, "Modified goldstein algorithm using boundary information in phase unwrapping," in *Advances in Imaging*, OSA, 2009.
- [13] M. G. de Pinto, M. Polignano, P. Lops, and G. Semeraro, "Emotions understanding model from spoken language using deep neural networks and mel-frequency cepstral coefficients," in *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, IEEE, May 2020.
- [14] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, "Amazon s3 for science grids: A viable solution?," in *Proceedings of the 2008 International Workshop on Data-Aware Distributed Computing, DADC '08*, (New York, NY, USA), p. 55–64, Association for Computing Machinery, 2008.
- [15] E. Liberty, Z. Karnin, B. Xiang, L. Rouesnel, B. Coskun, R. Nallapati, J. Delgado, A. Sadoughi, Y. Astashonok, P. Das, C. Balioglu, S. Chakravarty, M. Jha, P. Gautier, D. Arpin, T. Januschowski, V. Flunkert, Y. Wang, J. Gasthaus, L. Stella, S. Rangapuram, D. Salinas, S. Schelter, and A. Smola, "Elastic machine learning algorithms in amazon sagemaker," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, SIGMOD '20*, (New York, NY, USA), p. 731–737, Association for Computing Machinery, 2020.
- [16] A. Baird, S. Buliani, V. Nagrani, and A. Nair, "Aws serverless multi-tier architectures; using amazon api gateway and aws lambda," *Amazon Web Services Inc*, 2015.
- [17] S. Sivasubramanian, "Amazon dynamodb: A seamlessly scalable non-relational database service," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12*, (New York, NY, USA), p. 729–730, Association for Computing Machinery, 2012.
- [18] V.-A. Zamfir, M. Carabas, C. Carabas, and N. Tapus, "Systems monitoring and big data analysis using the elasticsearch system," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, pp. 188–193, 2019.
- [19] A. Toshniwal, K. S. Rathore, A. Dubey, P. Dhasal, and R. Maheshwari, "Media streaming in cloud with special reference to amazon web services: A comprehensive review," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 368–372, 2020.