Grenze International Journal of Engineering and Technology, June Issue

Waves to Genres: An Ensemble Machine Learning Approach to Music Classification

V. Sagar Reddy[1], G. Venkata Viswas Reddy[2], K. Pradhyuth Mohan[3], T. Lohita[4] and Ranjan K. Senapati[5]
[1-5]VNR, VJIET/ECE, Hyderabad, India
Email: vsagarreddy1990@gmail.com viswasreddy1804@gmail.com, pradhyuth18@gmail.com, tlohita19@gmail.com, ranjan_ks@vnrvjiet.in

***Abstract*— The Classification of Music genres is an absolutely crucial task in music analysis, with applications ranging from user-tailored music recommendation systems to automated music cata- loguing. This paper presents a novel approach to music genre classification using an ensemble ma- chine learning model which combines the strengths of both Artificial Neural Networks and Convolu- tional Neural Networks. Our objective is to efficiently and accurately classify music samples into their respective genres, while maintaining a focus on enhancing the classification accuracy. The ap- proach involves preprocessing of audio samples to extract their Mel-Frequency Cepstral Coefficients (MFCCs), which are robust feature representations for audio-based data. We compile a comprehen- sive JSON dataset containing MFCC features for each music sample. The ANN is employed for initial data processing, efficiently handling a large dataset, while the CNN excels in capturing spatial pat- terns in audio data, enhancing genre classification accuracy. In this paper, we provide a detailed overview of our methodology, including data preprocessing, model architecture, and training proce- dures. We also conduct experiments using the GTZAN dataset, consisting of 1,000 music samples, to assess the performance of our ensemble model approach. The results produced demonstrate the ef- fectiveness of our strategy in achieving the goal accurate genre classification. This research contrib- utes to the improvement of music genre classification techniques, showcasing the potential of com- bining machine learning models to handle the complexity of audio data. The findings have practical implications for music streaming platforms, content recommendation systems, and music catalogu- ing, thus offering new avenues for enhancing user experiences and automating music analysis.**

***Index Terms*— Spectrogram, Convolutional Neural Network, Ensemble Machine Learning Model, Flattened layer, Mel-frequency cepstral coefficients.**

I. INTRODUCTION

Music, a universal language that transcends cultural boundaries and resonates with the human soul, has experienced remarkable evolution in the modern era. From the enduring harmonies of classical compositions to the pulsating rhythms of contemporary electronic dance music, the world of music encompasses a diverse and ever-expanding array of genres. Amidst this rich musical tapestry, accurately classifying songs into their respective genres presents an intriguing challenge at the intersection of music and machine learning. Music genre classification serves as the foundation for various applications, including personalized music recommendations, content organization for music streaming platforms, and insights for musicologists studying evolving musical

Grenze ID: 01.GIJET.10.2.94

II. Literature Review

A music genre recommendation system based on signal processing and a CNN model named as MusicRecNet is used in which they use features extracted by TF analysis such as MFCC, Spectral Centroid, Chroma short-time Fourier transform(STFT) and wavelet analysis. [1], [2]
The use of GTZAN dataset for the classification of music genre and the use of Librosa library in python to extract the features of the dataset to obtain the log energy spectrum on the Mel-frequency scale, MFCCs incorporate timbral features of the music signal. [9]
The application of CNN in activation function ReLU. The network is made up of dense layer of 512 nodes and an outgoing layer with nodes equal to the number of classes to be separated. They concluded that music information retrieval is to classify musical elements such as genre, mood, rhythm and instrumentation. [6] Using single CNN gives accuracy ranges of 50% to 90% with average accuracy of prediction = 83.30%. [3]
The authors compared deep learning technique with traditional machine learning approaches. Their hypothesis is that accuracy of a deep-learning convolutional neural network, using spectrogram images as input, will be greater than the classification precision of traditional machine learning approaches, using content-based features for input, for the same given audio dataset. They compared between deep learning approach and machine learning techniques like Logistic Regression, K-Nearer Neighbors, Multilayer Per- ceptron, etc. At last, they find that accuracy for both the approaches give the similar result. [4], [8]

III. Method Of Classification

The dataset GTZAN was utilized in this experiment to get audio recordings for pre-processing. The GTZAN dataset is made up of ten genres, each with 100 audio recordings lasting 30 seconds. Hip-hop, disco, jazz, blues, rock, classical, reggae, country, metal, and pop are among the genres. The data set contains audio files with unique markers each to themself and no common means to distinguish them. The loading of audio files is possible due to the use of the python library “Librosa”. To perform a distinction, we must be able to quantify these unique markers that belong to each audio sample which can be used indefinitely with no underlying or specific discrimination to one particular audio mode. To perform this distinction the use of “ Mel Frequency Cepstrum” or MFC’s for short is optimal and robust. In sound processing, the Mel-frequency cepstrum is a representation of a sound's short-term power spectrum based on a linear cosine transform of a log power spectrum on a nonlinear Mel frequency scale. Firstly, since all of our music samples are continuous, we must convert them to their respective frequency domain functions using a Fast Fourier Trans- form, whose application on a continuous sound wave is shown in the equation below.

$$Xn = \sum N-1\ xke-2\pi jkn/N \qquad (1)$$

After that we have to find the amplitude spectrum of the samples and generate a frequency variable that is independent to each sample using the Mel-frequency equation specified below.

$$mel(f) = 2592 * log10$$

$$(1 + f\ ) \qquad (2)$$

$$700$$

Then, using half of the spectrum and frequency respectively we generate a spectrogram by performing a short time Fourier trans- form and taking its absolute value over the half-spectrum-frequency range, the generation of said Mel Spectrum is specified in the equation below.

$$Xi = log10(\sum N-1|X(k)| * Hik) \qquad (3)$$

All that is left is to generate the Cepstrum co-efficients from the Mel Spectrum by performing a Discrete Cosine Transform as shown in the equation below.

$$C\_j= [\![\sum]\!]\_(i=1)\^M [\![X]\!]\_i \cos(j((i-1)/2)*\pi/M\ ) \qquad (4)$$

After obtaining the Cepstrum Co-efficients of a sample we must label the sample using it, to do this we first choose a specific genre and then save its genre label as a mapping variable and then process all the files under the same genre label in order of genre tags. A pictorial representation of the process is defined below in flow-chart format with the corresponding numbered equations in Fig. 1, represented below.

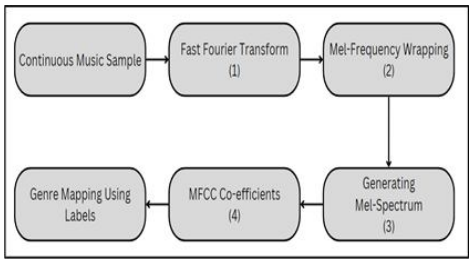

Figure 1. Flow Chart of Co-efficient generation process from audio samples.

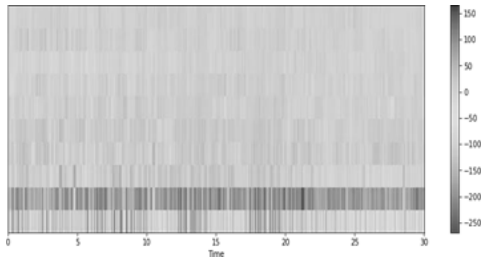

Figure 2. MFCC data of a randomly chosen audio sample.

The MFCC data of a random audio sample that we chose for representation purposes is displayed below in Figure 2.

The training parameters of the entire model is specified in the table below including the learning rate of samples which affects the efficiency of the model significantly and the most optimal value we recorded was a learning rate of 0.0001 which provided a quick run-time and also minimal losses.

TABLE I. TABLE OF TRAINING PARAMETERS

Training Parameters	Value
Learning Rate	0.0001
Total Trainable Parameters	1014218
Batch Size	32
Number of Genre Labels	10
Number of Audio Samples Per Label	100
Train/Test Size	700/300

IV. PROPOSED MODEL

We have trained and tested our model using software python language. As python is a concise and easily readable language, it allows us to develop readable deep learning models easily. We opted to use the .ipynb format for our code files since it enables the execution of separate code-blocks independently and helped us very much in the testing and optimization of our software program. The flow chart in Fig. 3, represents the execution flow of our model.

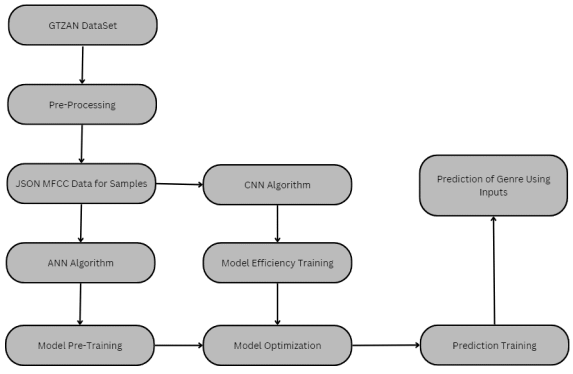

Figure 3. Flow chart of proposed model

The details of the proposed model include the steps below:

1. Loading Samples
2. Pre-processing Data Samples
3. Creating an Artificial Neural Network
4. Managing Overfitting of Neural Network
5. Creating a Convolutional Neural network
6. Making Predictions on Test Data
7. Generating Predictions using Model and an External Input

8. Saving and Executing the Model

A. Loading Samples

Using 30% of the dataset for testing and 70% for training, the downloaded dataset is split into two sets. This procedure helps to prepare the data for training and evaluation.

B. Pre-processing Data Samples

Preprocessing aims to improve the data's suitability for the analysis or machine learning task that it will be used for. Preprocessing in the context of audio data, such as music, might entail a number of procedures, including Data Cleaning, Normalization, Feature Extraction, etc. In the situation at hand, we compute the Mel-Frequency Cepstral Coefficients (MFCCs, for short) for each music sample and turn them into a JSON data sheet.

C. Creating an Artificial Neural Network

We design a simple Artificial Neural Network, or ANN, to handle the bulk of the data because the quantity of the training data (700) is still far too huge for the training to be completed quickly and accurately. Instead of sacrificing efficiency for accuracy, we chose to utilize an ANN because all of our data is in Structured format and the ANN can scan through this volume of data in a couple of seconds.

D. Managing Overfitting of Neural Network

Artificial Neural Networks (ANNs) and other machine learning models frequently experience overfitting. When a model excels at performing extraordinarily well on training data but struggles to generalize its predictions to brand-new, untried data, it experiences this problem. In essence, the model has learned the training data too well, collecting noise and peculiarities rather than the under- lying patterns in the data.

E. Creating a Convolutional Neural Network

CNNs are quite good at jobs like genre categorization since they are built to capture local patterns and features in images. They automatically learn and extract these features using convolutional layers. For the most effective prediction training in our scenario, we utilized a 5-layer CNN to maintain the model time-efficient. Although CNNs are excellent at these kinds of jobs, some structured data tasks may find them to be too difficult. Because structured data lacks inherent spatial correlations, we employed an ANN, which is simpler and more effective.

F. Making Predictions on Test Data

We first run each test sample through the trained model and compare the predictions it makes to the expected results as shown, using the remaining 30% of the dataset set aside for evaluating the model's training. We then create metrics like accuracy and error rates to confirm the integrity of the trained model. The evaluation results on the testing set provide a clear and unbiased measure of the model's performance.

G. Generating Predictions using Model and an External Input

Once we have a model with adequate accuracy and a minimal error rate, we may test it against any input we like. To ensure that the output result of the model was as clear and obvious as possible with few to no exceptions, we in this instance chose music clips that were 30 seconds long and clearly identified as belonging to a specific genre of music.

H. Saving and Executing the Model

We decided to save the model as a two-part file, with the first half continuing until the data is preprocessed and our JSON data file holding the MFCCs of data samples is created. The development and improvement of our neural networks, as well as the code for evaluating the model and making predictions, would be covered in the second half. By dividing the model into two halves, we can ensure that preprocessing has no impact on the way the model is constructed and that the only file on which the neural networks depend is the JSON data file received from the first half. We only need to restart the first half of the model if there is any discrepancy in the preparation of the data; we do not need to wait until the complete data model has been executed.

V. Experimental Results

This section holds the evaluation of our Machine learning Model along with their accuracies when trained using our GTZAN dataset. It also has the prediction results for when the model is tested with a sample of data taken from the internet.

When the model is tested against music of known genre labels as shown below in Fig. 4, it returns the output fairly accurately with around 92 samples being returned with the correct genre mapping out of 100 samples, which is because of the variable nature of music and its tendency to change with respect to the artist or genre.

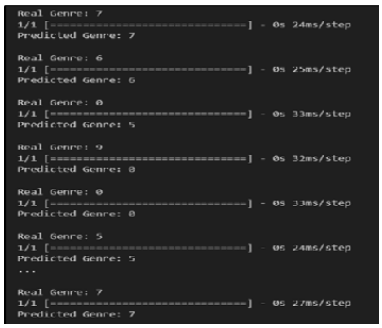

Figure 4. The above figure shows the results after we trained a model to know how well our model performs for specific genre and how many genres it can predict.

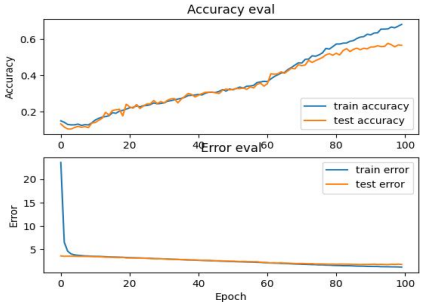

Figure 5. Model accuracy and error of ANN

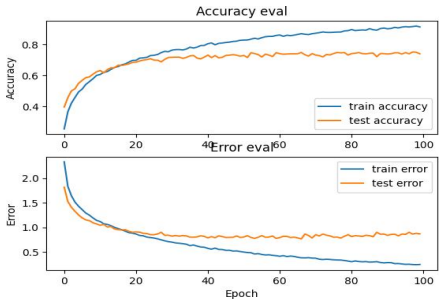

Figure 6. Model accuracy and error of CNN

Fig. 5 conveys that accuracy level of training data and testing data is very close to each other so that the output can be predicted more accurately using an ANN model and error rate is very low whereas in Fig. 6 there is gap between accuracy level of training and testing data and error rate is higher in testing.

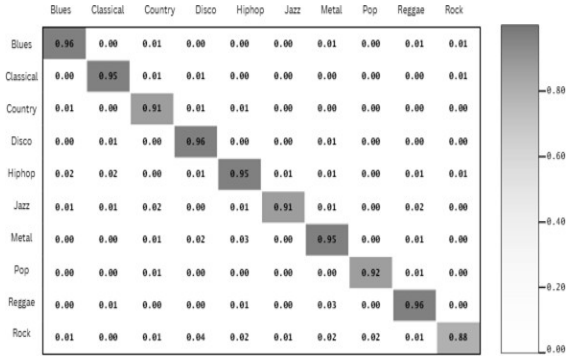

Figure 7. Confusion Matrix

The Confusion Matrix in Fig. 7 represents the Mapping Accuracy of our Ensemble model. In our classification of music genre each row represents the instances of a predicted genre, and each column represents the instances of an actual genre. The diagonal elements represent the number of correctly classified instances for each genre, and off-diagonal elements represent misclassifications.

VI. Discussion

The process of music genre classification using spectrograms typically involves these three key stages, Data Preprocessing: the initial step involves preprocessing the audio data to eliminate noise and other artifacts that might hinder the classification process. Common preprocessing techniques include filtering, normalization, and silence removal.

Secondly, Feature Extraction: Spectrograms are generated from the pre-processed audio data. These spectrograms serve as the basis for extracting relevant features for music genre classification. Mel-frequency cepstral coefficients and spectral centroid are among the commonly extracted features. Lastly, Classification: The features are then employed to train a classification model. This trained model is subsequently utilized to classify new audio recordings into their respective music genres. Convolutional neural networks have emerged as a powerful deep learning architecture, particularly adept at image classification tasks. Their effectiveness extends to music genre classification using spectrograms as well.

Some other limiting factors that we realized were some faults in the dataset we chose such as repetitions and mislabeling which we were able to recognize. CNNs excel in learning complex patterns within spectrograms that are indicative of different music genres. They possess the capability of capturing both local and global information in spectrograms, a crucial aspect for accurate music genre classification. The used convolutional neural network that includes Squeeze & Excitation Block (SE-Block) and obtained 92.7% accuracy on classification by using GTZAN dataset.

The Spectrograms effectively capture both temporal and spectral information from audio signals, providing a rich representation for music genre classification. Their ability to represent both local and global patterns makes them particularly well-suited for this task. Data augmentation techniques, such as pitch shifting and time stretching, can significantly increase the training dataset, lead- ing to improved classification accuracy. By artificially increasing the amount of training data, the model is exposed to a wider range of musical patterns, improving its ability to generalize to unseen data.

Other models which use just a single neural network have shown relatively lower accuracy rates under a comparative analysis of their results and it is very clear that an ensemble model proves to provide enhanced results[7]. Another combination of neural networks we considered was the ensemble of a KNN and PCA [5], but the result of it averaged from between 75-80% which is much lower than what we obtained from our proposed model. Another keen observation and determining factor were the choice of spectrogram parameters, such as window size and hop length, as these can significantly impact classification performance. Window size determines the time resolution of the spectrogram, while hop length determines the frequency resolution. Selecting appropriate values for these parameters is crucial for extracting optimal features for classification.

Table II below provides a comparison between the accuracy findings from our ensemble machine learning model and those from other pertinent studies on the categorization of musical genres. The accuracy metrics from several models, including our own, are compared to those reported in previous studies. Every entry in the table shows how well a model performed in categorizing different musical genres using the relevant datasets and techniques.

TABLE II. COMPARATIVE ANALYSIS OF MUSIC GENRE CLASSIFICATION ACCURACY

Named Model	Accuracy Value (%)	Dataset Used	Key Method
Our model	92.7	GTZAN	Ensemble Modelling
MusicRecNet [1]	81.8	GTZAN	Deep Learning
Shin et al. [11]	84.5	ISMIR2004	Auditory Spike Coding
NMF [12]	63.5	IMSIR2004	Expectation Maximization
Dhevan S. Lau & Ritesh Ajoodha [8]	81%	GTZAN	Logistic Regression

The reason behind choosing the models above to perform a comparative analysis is because they show similar inspirations to the design of our model and they approach processing of the audio samples using methods similar to MFCC conversion which convert the sample into a spectrum rather than use any preset feature values that are included in the dataset itself.

VII. CONCLUSION

Classification using spectrograms and convolutional neural networks (CNNs) has emerged as a powerful and effective approach in the field of music information retrieval. There is an imperative need to enhance the training data to the extent where we can use samples that are as accurate as possible and have minimal repetition and mislabeling but at the same time can have a scale large enough to accommodate the training requirements of heavy-weight learning algorithms. The approach of systems such as the one in this paper have found widespread applications in music recommendation systems, music search engines, and automatic playlist generation,

enhancing the user experience and personalization of music consumption and hopefully can accommodate the ever- changing creativity of artists that create music that transcends the barriers of genres.
As represented in Table II, our model has a reliably large accuracy despite not having any other classifier or classifying technique to supplement the final accuracy values that are obtained from base values. We could also use a classifier to supplement our results such as a Support Vector Machine (SVM) algorithm but we simply decided that the method of ensembling two machine learning algorithms was enough to produce results that could be considered above the range of current models while maintaining its efficiency.

REFERENCES

[1] Elbir and N. Aydin, "Music genre classification and music recommendation by using deep learning", 2020.
[2] Lakshman Kumar Puppala, Siva Sankar Reddy Muvva, Sudarshan Reddy Chinige, P.Selvi Rajendran, "A Novel Music Genre Classification Using Convolutional Neural Network", IEEE, 2021.
[3] Yu-Huei Cheng, Pang-Ching Chang, Che-Nan Kuo, "Convolutional Neural Networks Approach for Music Genre Classifiaction",2020.
[4] Sheetal Janthakal, "Automatic Classification of Music Genre Using the Deep Learning Approach, 2023.
[5] Suriya Prakash J, Kiran S, "Obtain Better Accuracy Using Music Genre Classification System on GTZAN Dataset", IEEE, 2022.
[6] Nikki Pelchat, Craig M. Gelowitz, "Neural Network Music Genre Classification",2020.
[7] Anirudh Ghildiyal, Komal Singh, Sachin Sharma, "Music Genre Classification using Machine Learning", 2020.
[8] Dhevan S. Lau & Ritesh Ajoodha, "Music Genre Classification: A Comparative Study Between Deep Learning and Traditional Machine Learning Approaches", 2021.
[9] Naman Kothari, Pawan Kumar, "Literature Survey for Music Genre Classification using Neural Network", 2022.
[10] Yije Xu, Wuneng Zhou, "A deep music genres classification model based on CNN with Squeeze & Excitation Block", 2022.
[11] Seong-Hyeon Shin, Ho-Won Yun, Woo-Jin Jang, Hochong Park "Extraction of acoustic features based on auditory spike code and its application to music genre classification", 2019
[12] AndrÉ Holzapfel, Yannis Stylianou "Musical Genre Classification Using Nonnegative Matrix Factorization-Based Features", 2008