Music Genre Classification Project Using Machine Learning

Kavya Sree Soma
dept. of Computer Science
University of Massachusetts Lowell
Lowell, USA
KavyaSree_Soma@student.uml.edu

Vagdevi Nandimandalam
dept. of Computer Science
University of Massachusetts Lowell
Lowell, USA
vagdevi_nandimandalam@student.uml.edu

Arun Kumar Coimbatore Dada
dept. of Computer Science
University of Massachusetts Lowell
Lowell, USA
arunkumar_coimbatoredada@student.uml.edu

Udith Lakshmi Narayan
dept. of Computer Science
University of Massachusetts Lowell
Lowell, USA
fnu_udithlakshminarayan@student.uml.edu

Anurag kalapala
dept. of Computer Science
University of Massachusetts Lowell
Lowell, USA
anurag_kalapala@student.uml.edu

I. ABSTRACT

The goal of the music genre classification model is to group songs according to a variety of characteristics. The need for an accurate classification model is growing as people rely more and more on music and as technology and the internet become more accessible. This project effectively classifies and arranges audio music into its appropriate genres using machine learning techniques. The audio's spectrogram and acoustic characteristics are used in the classification. The objective is to develop a model with improved music genre classification accuracy. The music industry has changed significantly over the years, with diverse music styles catering to a growing customer base. Classifying music into genres is crucial to meet people's preferences, but manual ranking is time-consuming. In this research, various classification models were compared, and a new and improved CNN model was proposed and trained on the GTZAN dataset, outperforming previously suggested models. This study enhances the efficiency of music genre classification, particularly using audio files and spectrograms.

II. INTRODUCTION

For many generations, music has been a vital aspect of human existence, shaped by a variety of elements including history, culture, popular culture, and marketing. Music is categorized into genres to create order in this chaotic environment. Machine Learning techniques are used in our Music Genre Classification project to automate this process. In the real world, countless songs appeal to various demographics based on language, acoustic characteristics, age group, and community. We need a model that can automatically categorize songs into genres to handle this enormous dataset and do away with the need for manual classification. The software does the classification instead of the users. Frequency, decibel, and bandwidth are some of the parameters that define sound.

S. No	Class	Clips
1	Blues	100
2	Classical	100
3	Country	100
4	Disco	100
5	Hiphop	100
6	Jazz	100
7	Metal	100
8	Pop	100
9	Reggae	100
10	Rock	100
	Total	1000

Fig. 1. NUMBER OF AUDIO CLIPS IN EACH GENRE

These parameters are combined to form spectrograms, which are helpful tools for analyzing audio characteristics.

III. BACKGROUND

Music genre classification is a pivotal task in the field of audio signal processing and machine learning. As music continues to be a diverse and integral part of human culture, the ability to automatically classify music into distinct genres has numerous practical applications. From personalized music recommendations to content organization in digital libraries, accurate genre classification lays the foundation for enhanced user experiences.

The task of assigning genres to musical pieces is inherently complex due to the subjective nature of musical expression and the vast diversity within genres themselves. Manual annotation of music genres is time-consuming and subjective, making

automated classification systems desirable for scalability and objectivity.

The advent of machine learning techniques, particularly deep learning models like Convolutional Neural Networks (CNNs), has revolutionized the landscape of music genre classification. Unlike traditional methods that rely on handcrafted features, machine learning models can automatically learn hierarchical representations from raw audio signals, capturing intricate patterns and nuances inherent in different genres.

IV. DATASETS

Download GTZAN Dataset from: http://opihi.cs.uvic.ca/sound/genres.tar.gz Raw data is 1.2GB. Originally proposed by G. Tzanetakis in [1], GTZAN is a widely used dataset in music signal processing. It has one thousand 30-second tracks at a sampling frequency of 22050 Hz and 16 bits. There are 100 songs in each of the following genres: pop, rock, country, disco, hip-hop, jazz, metal, blues, pop, and reggae.

V. METHODOLOGY

We have a dataset called GTZAN that contains audio files in the .wav format. We utilize a Python package called Librosa to examine and extract features for machine learning from this data. Time domain and frequency domain features are both included in the extracted features.

Following extraction, we review each feature and choose the best ones. A CSV file containing these finished features is then saved. Our machine-learning classifiers will use this file as input to help them categorize various musical genres.

We perform some initial preprocessing using various techniques to clean and organize the data before putting it through the classifiers. Making sure the data is appropriate for our machine learning models requires taking this important step.

Lastly, we optimise the parameters of our model, paying particular attention to choosing the appropriate hyperparameters, in order to improve its performance. For accurate and efficient classification, determining the ideal values for these parameters is crucial because they have a substantial impact on the predictions our model makes.

K-NN, or K-Nearest Neighbour: A supervised machine learning algorithm is called K-NN. K-NN classifies a new case into the most similar category among the cases that are currently available based on how similar the new case is to the pre-existing case.

Convolutional neural networks (CNNs): CNN are used to analyse and comprehend data, such as patterns or images. To demonstrate how well the CNN could recognize and classify data, we built it with several layers that each learn a particular feature. We then trained the CNN on a dataset and evaluated its performance using accuracy and other metrics. Our findings opened the door for better techniques in related research by proving CNN's suitability for the given task.

Preparing the Dataset: Two prominent datasets for music genre classification are the FMA dataset, which includes information on the audio characteristics of 8,000 distinct songs from 8 different genres, and the GTZAN dataset, which includes 1000 audio files from 10 different genres.

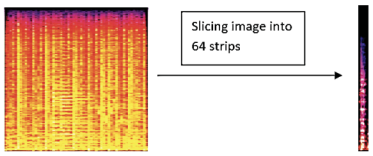

Fig. 2. Pre-processing of images, slicing image into 64 strips

Data Pre-Processing: There are only 100 audio clips in the dataset for each category, which might not be enough to produce reliable findings. To address this, one of two options is available: either use an expanded dataset that contains more audio clips or alter the current dataset to add more training and testing samples for improved accuracy.

Using the librosa library, we generated mel-spectrograms from audio files. By splitting each spectrogram into 64 strips, we were able to increase the number of samples we had to 64. As a result, 64,000 samples in all, each measuring 480 by 10, were produced for testing and training. 44,200 of these were used for testing, 7,000 for validation, and 7,000 for training purposes. For an example of the original image and the strips created during this process, refer to Fig. 1.

Librosa makes it super easy to create spectograms. A 3 line code can convert an audio file into a spectogram!

```
y, sr = librosa.load(filename)
spect = librosa.feature.melspectrogram(y=y,
        sr=sr,n_fft=2048, hop_length=512)
spect = librosa.power_to_db(spect,
        ref=np.max)
```

Feature extraction: Each audio can be expressed as an audio signal, which possesses various features. The pertinent audio features are extracted to solve the issue. There are two subcategories for these features.

A. *Time Domain Characteristic:*

- i. Zero Crossing Rate: Zero-crossing rate is the rate at which a signal shifts from positive to negative or the opposite.
- ii. Root Mean Square Energy: A signal's root mean square energy, or RMSE, indicates its loudness.

B. *Frequency Domain Features:*

- i. Mel-Frequency Capstral Coefficient: The Mel-Frequency Capstral Coefficient is a set of features (approximately 10–20) that characterize the audio signal's shape.
- ii. Chroma Features: A music signal is represented by projecting its whole spectrum onto 12 bits, which stand for the 12 different semitones of a musical octave.
- iii. Centroid Spectral: The "center of mass" of the signal is revealed by the weighted mean of the frequencies that make up the sound.

- iv. Spectral Roll-off: A defined percentage of the total spectral energy lies below a certain value frequency.

Convolution Neural Network: It is evident that features unique to each class exist even in an image as small as 480 x 10. These pictures serve as input for our CNN model.

CNN Model: Our deep neural network in fig 3, which consists of two sub-networks, receives the training images, or the sliced images of the spectrogram. To extract features from the images, the first neural network is a four-layer convolution neural network. The second sub-network receives these extracted features and uses them for classification. There are two fully connected layers in this fully connected network. Ultimately, the audio genre is predicted using a dense layer.

Layer (type)	Output Shape	Param #
batch_normalization_95 (BatchNormalization)	(None, 288, 432, 3)	12
conv2d_94 (Conv2D)	(None, 286, 430, 32)	896
max_pooling2d_4 (MaxPooling2D)	(None, 143, 215, 32)	0
conv2d_95 (Conv2D)	(None, 141, 213, 64)	18496
max_pooling2d_5 (MaxPooling2D)	(None, 70, 106, 64)	0
conv2d_96 (Conv2D)	(None, 68, 104, 128)	73856
max_pooling2d_6 (MaxPooling2D)	(None, 34, 52, 128)	0
conv2d_97 (Conv2D)	(None, 32, 50, 256)	295168
max_pooling2d_7 (MaxPooling2D)	(None, 16, 25, 256)	0
conv2d_98 (Conv2D)	(None, 14, 23, 512)	1180160
max_pooling2d_8 (MaxPooling2D)	(None, 7, 11, 512)	0
flatten_1 (Flatten)	(None, 39424)	0
dense_4 (Dense)	(None, 1024)	40371200
dropout_3 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 512)	524800
dropout_4 (Dropout)	(None, 512)	0
batch_normalization_96 (BatchNormalization)	(None, 512)	2048
dense_6 (Dense)	(None, 10)	5130

Total params: 42471766 (162.02 MB)
Trainable params: 42470736 (162.01 MB)
Non-trainable params: 1030 (4.02 KB)

Fig. 3. CNN model: Performed multiple convolution and passed it through the Dense layers by adding Drop out of 0.3.

Each layer of the CNN does the following operations:

- Convolution: The filters in this layer have a set of parameters that must be learned. Every filter has a smaller size than the input image, which is typically 3 by 3. This filter runs over the picture, encompassing every pixel value. The scaler product between the image and the filter is computed as the filter runs through the image.
- Max Pooling: The pooling layer's function is to conduct downsampling based on the input's dimension. decreasing the quantity of parameters associated with that activation. Average pooling and maximum pooling are two popular pooling techniques.
- Dropout: This is a method to keep our model from overfitting and boost its effectiveness. Every time we train our model, we randomly set some of the neurons' weights to zero. The final output is predicted by combining various neural combinations. There is a dropout rate of 0.3 or a neuron width

C.

Other Models: Convolutional Recurrent Neural Networks (CRNN) are a type of neural network architecture that combines the strengths of both CNNs and RNNs. Each component plays a specific role in processing different aspects of the input data, making CRNNs particularly effective for tasks that involve both spatial and temporal information.

- CNN (Convolutional Neural Network):
  Spatial Feature Extraction: CNNs are well-suited for extracting spatial features from input data. In the context of image data, like in computer vision tasks, CNNs can recognize patterns, edges, and textures. These spatial features capture the local information within the data.
- RNN (Recurrent Neural Network):
  Temporal Relationship Modeling: RNNs excel at capturing temporal dependencies and sequential patterns in data. They maintain an internal memory that allows them to consider the context of previous inputs when processing the current one. This makes RNNs effective for tasks where the order and temporal relationships among elements are crucial.
- Combining CNN and RNN in CRNN:
  Hybrid Architecture: In CRNN, the CNN and RNN components are integrated to create a hybrid architecture. The CNN component processes the input data, capturing spatial features, and produces a fixed-size representation. This representation is then fed into the RNN component, which utilizes its sequential processing capabilities to model temporal dependencies in the data.
  Example Application: In tasks such as image captioning or video analysis, CRNNs can be employed. The CNN extracts spatial features from the images or frames, and the RNN processes these features sequentially to generate a coherent output, taking into account both the content of the image and the temporal context.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

The accuracy of our model was evaluated by calculating the percentage of actual song genres that corresponded with the predicted genres. Considering the homogeneous distribution of genres in our dataset, confusion matrices provide a visual representation of the optimal model's performance. We used various methods to determine which hyperparameters were optimal for each algorithm. The classification outcomes for

each of the ten genres in our dataset are shown in the confusion matrix for each model.

"In the evaluation of our models, including Convolutional Neural Network (CNN), k-Nearest Neighbors (KNN), and Convolutional Recurrent Neural Network (CRNN), we have employed a comprehensive approach to assess their performance. Visual representations, such as accuracy and loss curves, as well as confusion matrices, have been meticulously analyzed to gain insights into each model's behavior.

CNN Model:

The accuracy and loss curves of the CNN model provide a dynamic view of its learning process, showcasing its ability to correctly classify instances and converge during training. The confusion matrix offers a detailed breakdown of the model's performance across different classes, guiding potential optimization strategies.

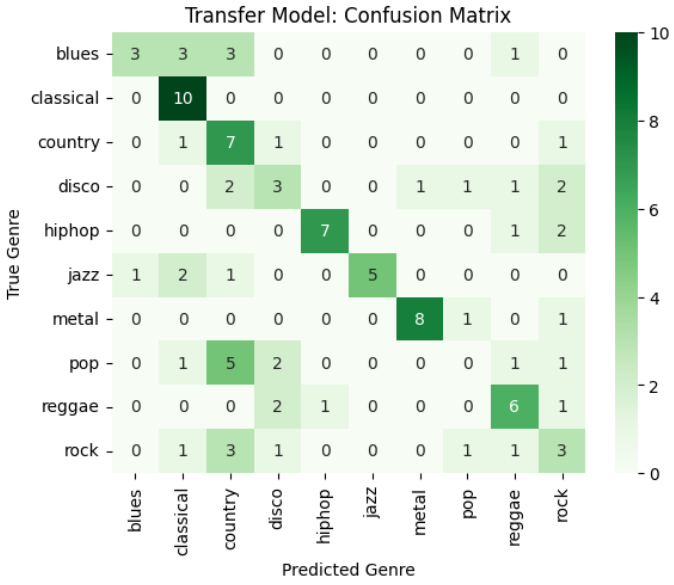

Fig. 4. CNN Confusion Matrix.

KNN Model:

For the KNN model, a thorough examination of accuracy and loss, though less applicable in the context of KNN, is complemented by the interpretation of the confusion matrix. The matrix allows us to understand how well the KNN model distinguishes between classes, providing valuable insights into its classification capabilities.

CRNN Model:

Similar to the CNN model, the CRNN model's accuracy and loss curves illustrate its learning dynamics, capturing both spatial and temporal features. The confusion matrix delves into the model's ability to navigate complex relationships between classes, enhancing our understanding of its overall performance.

```
Confusion Matrix (Mean Features):
 [[11  0  2  0  0  0  1  0  1  5]
 [ 0 12  0  0  0  1  0  0  0  0]
 [ 1  1 20  1  1  1  0  0  1  1]
 [ 1  0  3 13  1  0  1  1  0  1]
 [ 0  0  0  2 11  0  0  1  1  0]
 [ 0  3  2  1  0 15  0  0  0  1]
 [ 0  0  0  3  0  0 21  0  0  1]
 [ 1  0  0  1  0  1  0  9  0  1]
 [ 3  0  1  4  4  0  0  1 10  0]
 [ 2  1  3  2  0  0  0  2  1 10]]
```

Fig. 5. KNN Confusion Matrix.

```
Classification Report (Mean Features):
              precision    recall  f1-score   support

           0       0.58      0.55      0.56        20
           1       0.71      0.92      0.80        13
           2       0.65      0.74      0.69        27
           3       0.48      0.62      0.54        21
           4       0.65      0.73      0.69        15
           5       0.83      0.68      0.75        22
           6       0.91      0.84      0.87        25
           7       0.64      0.69      0.67        13
           8       0.71      0.43      0.54        23
           9       0.50      0.48      0.49        21

    accuracy                           0.66       200
   macro avg       0.67      0.67      0.66       200
weighted avg       0.67      0.66      0.66       200
```

Fig. 6. Calculated the precision, recall and F1 score for KNN model.

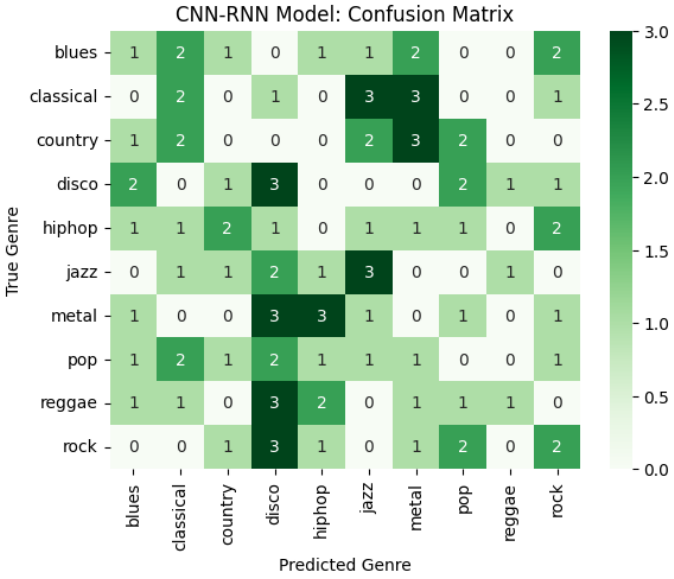

Fig. 7. CNN-RNN Confusion Matrix.

VII. CONCLUSION

In conclusion, the main goal of this research is to classify and recommend songs using acoustic characteristics that are acquired using convolutional neural networks and digital signal processing methods. The GTZAN dataset was used in the proposed research project to generate several models that

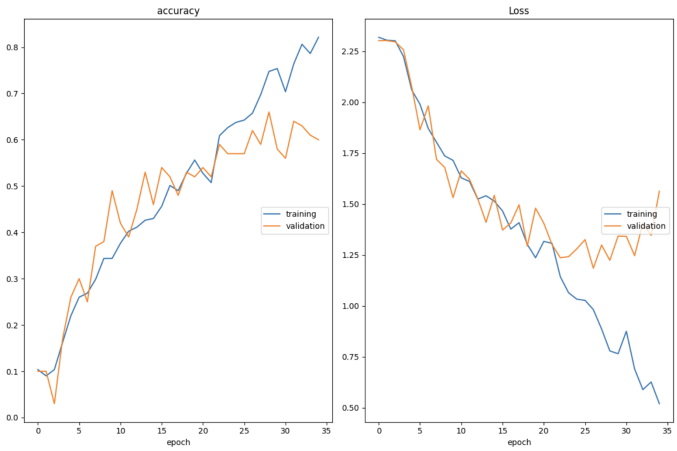

Fig. 8. The visual graph of accuracy and Loss for CNN-RNN model with the epoch of 60.

S.NO	Model	Training Accuracy	Testing Accuracy
1.	KNN	76%	72%
2.	CNN	84%	82%
3.	CNN-RNN	86%	54%

Fig. 9. The accuracies of testing and training data of all the models are tabulated

helped with the task of classifying this piece of music. The suggested model transferred the audio Mel spectrogram and multiple inputs for different models to our CNN and KNN. CNN classifier did the best with an 85 percent accuracy, showing it's great at understanding complex patterns. The KNN classifier did well too, hitting 82 percent. This tells us that deep learning, like CNNs, really helps boost how well our model works for this job.

VIII. FUTURE WORK

In future work, addressing the challenge of identifying genres for songs with multiple genre elements could involve extending the model to handle multi-label classification, allowing songs to be associated with multiple genres simultaneously. Techniques such as segmentation and fusion, sequential modeling using Recurrent Neural Networks (RNNs), attention mechanisms, and ensemble learning could be explored to capture temporal dependencies and genre transitions within songs. Additionally, the incorporation of large and diverse datasets, user feedback integration, and user-centric evaluation can contribute to the development of more robust and user-friendly models capable of accurately classifying the genres of entire songs with nuanced genre compositions.

IX. APPENDIX OF TASK DISTRIBUTION

Team Members	**Task Distribution**
Kavya Sree Soma, Vagdevi Nandhimandalam	Both collaborated on the CNN model.
Udith Lakshmi Narayan, Arun Kumar Coimbatore Dada	Took the lead in developing the CNN-RNN model.
Anurag Kalapala	Actively contributed to different components and stages of the KNN model.

TABLE I
TASK DISTRIBUTION AMONG TEAM MEMBERS

REFERENCES

[1] Jitesh Kumar Bhatia, Rishabh Dev Singh, Sanket Kumar, "Music Genre Classification", IEEE Oct 2021.
[2] S. Panwar, P. Rad, K.-K. R. Choo and M. Roopaei, "Are you emotional or depressed? Learning about your emotional state from your music using machine learning", 7e Journal of Supercomputing, vol. 75, no. 6, pp. 2986-3009, 2019.
[3] M. Serwach and B. Stasiak, "GA-based parameterization and feature selection for automatic music genre recognition", 17th International Conference Computational Problems of Electrical Engineering (CPEE), 2016.
[4] X. Liu, "An improved particle swarm optimization-powered adaptive classification and migration visualization for music style", Complexity, vol. 24, no. 7, pp. 0872, 2020.
[5] K. Balachandra, N. Kumari and T. Shukla, "Music Genre Classification for Indian Music Genres", International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 9, no. 4, pp. 1204-1212, Aug 2021.
[6] M. T. Quasim, E. H. Alkhammash and M. A. Khan, "Emotion-based music recommendation and classification using machine learning with IoT Framework", Soft Computing, vol. 25, no. 7, 2021.
[7] L. Van Dijk, "Finding musical genre similarity using machine learning", Bachelor thesis of Radboud Universiteit Nijmegen, pp. 1-25, 2014.
[8] D. Chaudhary, N. P. Singh and S. Singh, "Development of music emotion classification system using convolution neural network", International Journal of Speech Technology, vol. 24, no. 2, pp. 1-10, 2020.
[9] Y. Zheng, "The use of deep learning algorithm and digital media art in all-media intelligent electronic music system", PLoS One, vol. 15, 2020.
[10] Loris Nanni, Yandre MG Costa, Alessandra Lumini,Moo Young Kim, and Seung Ryul Baek. Combining visual and acoustic features for music genre classification. Expert Systems with Applications 45:108–117, 2016.
[11] Keunwoo Choi, George Fazekas, and Mark Sandler, "Explaining deep convolutional neural networks on music classification," arXiv preprint arXiv:1607.02444, 2016.
[12] Thomas Lidy and Alexander Schindler. Parallel convolutional neural networks for music genre and mood classification. MIREX2016, 2016.