IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Music Genre Classification

[1]K.T. Krishna Kumar, [2]Bharathi Erothi,
[1]Associate Professor and Placement Officer, [2]MCA Final Semester,
[1]Master of Computer Applications
[1]Sanketika Vidya Parishad Engineering College, Visakhapatnam, Andhra Pradesh, India.

Abstract: Music genre is a classification system that classifies music into different types. The classification of music genre is very important to make a selection of songs from a large collection of data. Different features have been extracted as it is essential for the genre classification. We use convolution neural networks to classify music genres. The project aims to create an automated system for classification models of music genres. We use the most publicly used data set which is GTZAN for evaluation of music genres. It contains 10 genres (blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, rock). The audio file is sent as input and the classification is done. CNN is used to classify the data furthermore.

***Index Terms* -** Music Genre Classification, Machine Learning, Audio Processing, Feature Extraction, Convolutional Neural Networks (CNN), Genre Detection, Audio Signal, Music Dataset (e.g., GTZAN, FMA), Audio File Formats (e.g., MP3, WAV), Signal Processing.

I. INTRODUCTION

Music is so important to everyone's life; it brings out so many emotions in us like excitement. Music can change someone's mood, get them productive, the possibilities are endless. Music Genre Recognition is an important field of research in Music Information Retrieval. A music genre is a conventional category that identifies some pieces of music as belonging to a shared tradition or set of conventions, i.e. it depicts the style of music. Music Genre [1] Recognition involves making use of features such as spectrograms, MFCC's for predicting the genre of music. Here we are going to make use of GTZAN [2] Dataset which is really famous in Music Information Retrieval (MIR). The Dataset comprises 10 genres namely Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae, Rock. Each genre comprises 100 audio files (.wav) of 30 seconds each. The model can learn the genre of the music by listening to the song 4-5 seconds without listening to the complete 30 sec song which takes more time to classify entire songs. We are going to use a Convolutional Neural Network [3], we need an image as an input, for this we will use the Mel spectrograms of audio files and save the spectrograms as an image file.

1.1 Existing System

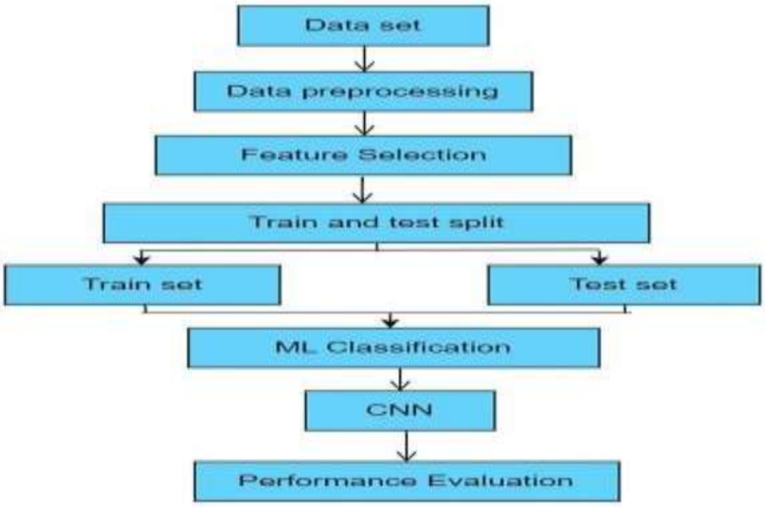
Most of the existing music genre recognition algorithms are based on manual feature extraction techniques. These extracted features are used to develop a classifier model to identify the genre. However, in many cases, it has been observed that a set of features giving excellent accuracy fails to explain the underlying typical characteristics of music genres. It has also been observed that some of the features provide a satisfactory level of performance on a particular dataset but fail to provide similar performance on other datasets. Hence, each dataset mostly requires manual selection of appropriate acoustic features to achieve an adequate level of performance on it.

1.1.1 Challenges

- Ambiguity in Genre Boundaries
- Complexity of Audio Signals
- Background Noise and Recording Quality
- Limited Dataset Size
- Real-Time Processing

1.2 Proposed System

A proposed system for music genre classification leverages machine learning techniques to automatically categorize music tracks into their respective genres. The system begins with data collection, gathering a diverse dataset of music tracks labeled with their genres from various sources such as online music databases and streaming services. Preprocessing the audio data involves converting tracks into a consistent format and extracting features using techniques like Mel-frequency cepstral coefficients (MFCCs), chroma features, and spectral contrast, which capture essential characteristics of the music.



Feature extraction is followed by the use of dimensionality reduction techniques like Principal Component Analysis (PCA) to minimize the feature set while retaining the most informative aspects. The reduced feature set is then fed into machine learning models such as k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and deep learning architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) designed for sequential data.

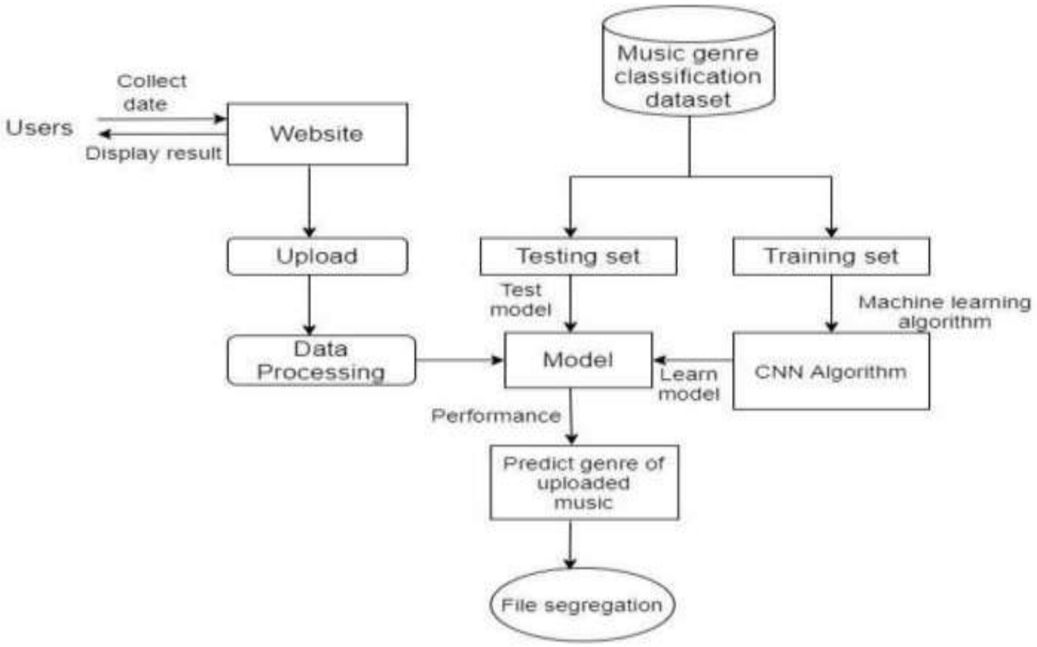
The system's performance is evaluated using metrics like accuracy, precision, recall, and F1 score, ensuring robust and reliable classification. Cross-validation techniques are employed to validate the models and prevent overfitting, ensuring that the models generalize well to unseen data. Additionally, ensemble methods can be explored to combine multiple models for improved classification performance.

Challenges include handling the high variability within genres, the overlap between genres, and the need for large, labeled datasets. To address these, data augmentation techniques can be employed to artificially increase the size of the dataset and improve model robustness. Furthermore, the system can be continuously updated with new data to adapt to emerging genres and changing musical trends.

Figure 1: Proposed System.

1.2.1 Advantages

- Enhanced User Experience
- Efficient Music Organization
- Improved Music Discovery
- Enhanced Music Analytics
- Personalized Recommendations
- Music Analysis and Research



II. LITERARTURE REVIEW

The architecture for music genre classification typically involves several key components. It starts with data collection, where a diverse and well-labelled dataset of audio tracks is gathered. Preprocessing follows, converting audio files into a suitable format, often using techniques like resampling and normalization.

Feature include Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, and spectrograms, which capture essential aspects of the audio signal. The modelling phase involves selecting an appropriate machine learning or deep learning model. Convolutional Neural Networks (CNNs) are popular due to their effectiveness in handling spectrograms, while Recurrent Neural Networks (RNNs)[[5] or Long Short-Term Memory (LSTM) networks can capture temporal dependencies. The network architecture often includes several convolutional layers for feature learning, followed by pooling layers to reduce dimensionality, and dense layers for classification. Training involves feeding the extracted features into the model, optimizing it using a loss function (e.g., cross-entropy) and backpropagation. Validation and testing are crucial to evaluate model performance and avoid overfitting, typically using techniques like k-fold cross-validation [6]. Finally, the model is deployed, where it can classify new audio tracks into genres in real-time. Throughout, tools like TensorFlow [7], Keras [8], and PyTorch [9] are commonly used for implementation.

Figure 2: Architecture Diagram

2.1 Algorithm

A convolutional neural network, or CNN, might be a great understanding neural network sketched for handling designed arrays of data like interpretations. CNNs are especially acceptable at understanding patterns within the input image, like lines, gradients, circles, or perhaps eyes and faces CNN is capable of operating directly on a rare image and doesn't need any pre-processing. A convolutional neural network may be a feed headfirst neural network, rarely with up to twenty. The depth of a convolutional neural network takes place from a chosen quiet layer called the convolutional layer. CNN includes many convolutional layers gathered on cover of each other, all capable of identifying more than complex shapes. Through three or four convolutional layers it's possible to recognize handwritten numerals and with 25 layers it's likely to distinguish human faces. The program for this sphere is to stimulate machines to view the planet by way humans prepare, observe it in an extremely similar fashion and smoothly use the information for a mess of duty like image and video recognition, image examination and organization, media restoration, reference schemes, tongue processing, etc.

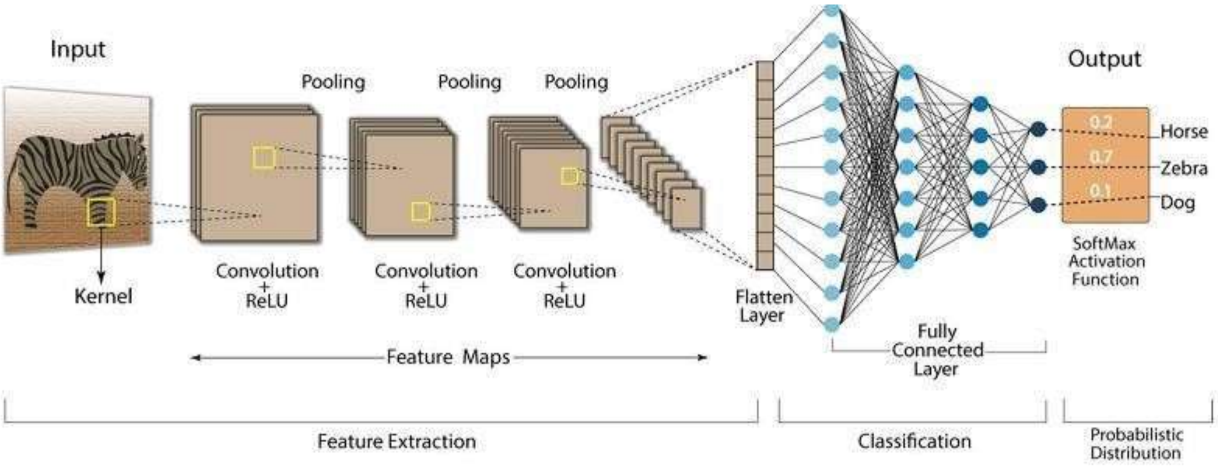


Figure 3: CNN Layer

2.2 Techniques

- Feature Extraction: Extracting audio features such as Mel-Frequency Cepstral Coefficients (MFCCs)[10], chroma features, and spectrograms to represent the audio signal.
- Deep Learning Models: Utilizing Convolutional Neural Networks (CNNs) for spatial pattern recognition in spectrograms and Recurrent Neural Networks (RNNs)[11] or Long Short-Term Memory (LSTM)[12] networks for capturing temporal dependencies.
- Data Augmentation: Enhancing the dataset with transformations like pitch shifting, time stretching, and adding noise to create diverse audio variations.
- Transfer Learning and Ensemble Methods: Applying pre-trained models and combining multiple models to improve classification accuracy and robustness.
- Evaluation and Regularization: Using metrics like precision, recall, F1 score, and cross-validation for model assessment, and applying regularization techniques such as dropout and batch normalization to prevent overfitting.

2.3 Tools

- **Librosa:** A Python library for audio and music analysis, useful for feature extraction and audio processing unit.
- **TensorFlow:** An open-source machine learning framework widely used for building and training deep learning models.
- **Keras:** A high-level neural networks API that runs on top of TensorFlow, making it easier to design and train deep learning models.
- **PyTorch:** An open-source deep learning framework known for its flexibility and dynamic computation graph, suitable for developing complex neural network models.
- **Scikit-learn:** A Python library for machine learning that provides tools for data preprocessing, model training, and evaluation.
- **Jupiter Notebook:** An interactive computing environment that allows for easy prototyping, visualization, and sharing of code and results.
- **Pandas:** A Python library for data manipulation and analysis, useful for handling and preprocessing audio datasets.
- **Matplotlib and Seaborn:** Visualization libraries in Python used for plotting and visualizing audio features, model performance, and results.

2.4 Methods

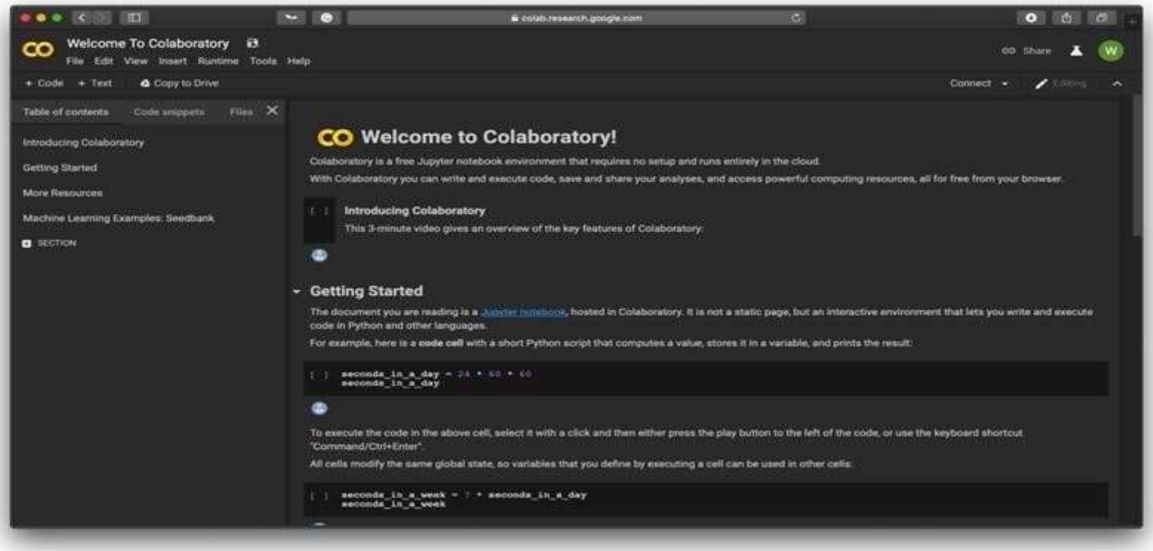
- **Spectrogram Analysis**: Converting audio signals into visual representations (spectrograms) to capture frequency and amplitude variations over time, which are then used as input for machine learning models.

- **Mel-Frequency Cepstral Coefficients (MFCCs)**: Extracting MFCCs, which represent the short-term power spectrum of sound, to capture the timbral aspects of music crucial for genre differentiation.
- **Convolutional Neural Networks (CNNs)**: Applying CNNs to learn an and classify spatial patterns from spectrogram images, effectively recognizing local features and structures in audio data.
- **Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) Networks**: Utilizing RNNs or LSTM networks to model temporal dependencies and sequence information in audio signals, capturing the dynamic characteristics of music.
- **Data Augmentation**: Enhancing the training dataset by applying transformations such as pitch shifting, time stretching, and adding noise, to improve model robustness and generalization.
- **Transfer Learning**: Using pre-trained models on related tasks and fine-tuning them for music genre classification, leveraging existing knowledge to improve performance with limited data.

III. METHODOLOGY

3.1 Input

Google Collab is a web-based notebook environment that allows users to write, run and share code in a collaborative online setting. It is a free cloud-based service offered by Google, based on Jupiter Notebooks. The platform provides users with a Python environment and access to free resources such as GPUs and TPUs. Google Collab allows you to create and edit Jupiter notebooks in your browser without having to install anything on your computer. This makes it an ideal choice for people who want to work on data science and machine learning projects, but may not have the necessary hardware or software. With Collab, you can easily import datasets, write code, and collaborate with others on projects. You can also use Collab to store and share your notebooks with others through Google Drive or GitHub [13]. One of the standout features of Collab is its ability to run code on Google's powerful servers, which makes it possible to train large machine learning models using powerful GPUs and TPUs. Additionally, Collab provides built-in integration with many popular machine learning frameworks such as TensorFlow, PyTorch, and Keras.


colab.research.google.com
Welcome To Colaboratory
File Edit View Insert Runtime Tools Help
Share
W
+ Code + Text Copy to Drive
Connect
Editing
Table of contents Code snippets Files
Introducing Colaboratory
Getting Started
More Resources
Machine Learning Examples: Seedbank
SECTION

Welcome to Colaboratory!

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.

Introducing Colaboratory

This 3-minute video gives an overview of the key features of Colaboratory:

Getting Started

The document you are reading is a Jupyter notebook, hosted in Colaboratory. It is not a static page, but an interactive environment that lets you write and execute code in Python and other languages.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ]  seconds_in_a_day = 24 * 60 * 60
     seconds_in_a_day
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter".

All cells modify the same global state, so variables that you define by executing a cell can be used in other cells:

```
[ ]  seconds_in_a_week = 7 * seconds_in_a_day
     seconds_in_a_week
```

Figure 4: Google Collab

3.2 Steps For Execution

- Executing a music genre classification project involve several key steps. First, gather labelled audio tracks datasets like GTZAN [14] or FMA.
- Preprocess these audio files by converting them to a consistent format, normalizing them, and segmenting them, if necessary, along with reducing noise and trimming silence.
- Extract important features such as MFCCs, chroma features, spectral contrast, and spectrograms using tools like Librosa.
- Enhance the dataset through data augmentation techniques like pitch shifting, time stretching, and adding noise. Choose and design appropriate model architectures, such as CNNs or RNNs/LSTMs, using frameworks like TensorFlow, Keras, or PyTorch.
- Split the dataset into training, validation, and test sets, and train the model, optimizing with suitable loss functions while monitoring progress.
- Evaluate model performance using metrics like accuracy, precision, recall, F1 score, and confusion matrix, and ensure robustness through cross-validation.
- Fine-tune hyperparameters to further improve performance. Deploy the trained model for real-time genre classification, ensuring it has a user-friendly interface.
- Finally, continuously monitor and maintain the model's performance, updating it with new data to adapt to evolving music trends and maintain accuracy.

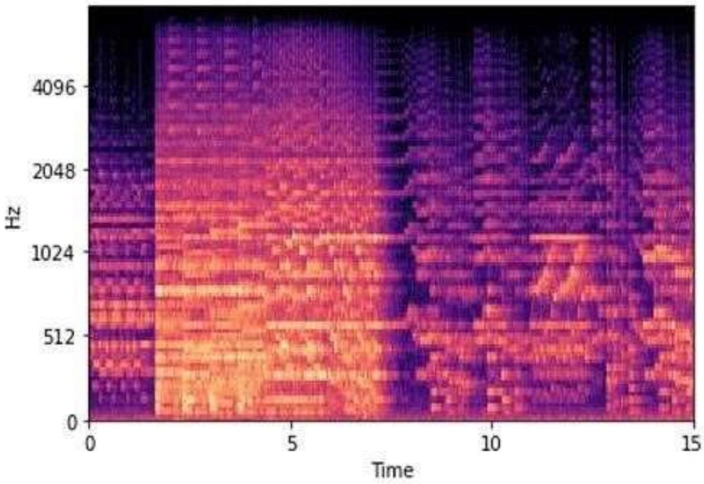
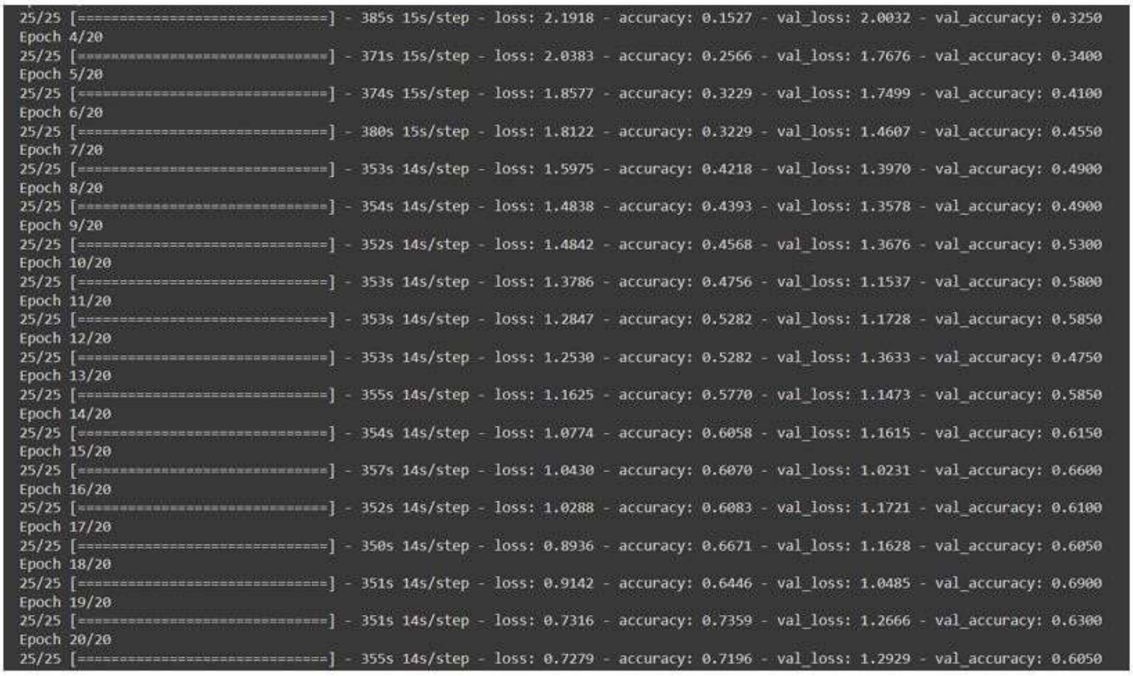


Figure 5: Output Screen1

3.3 Output

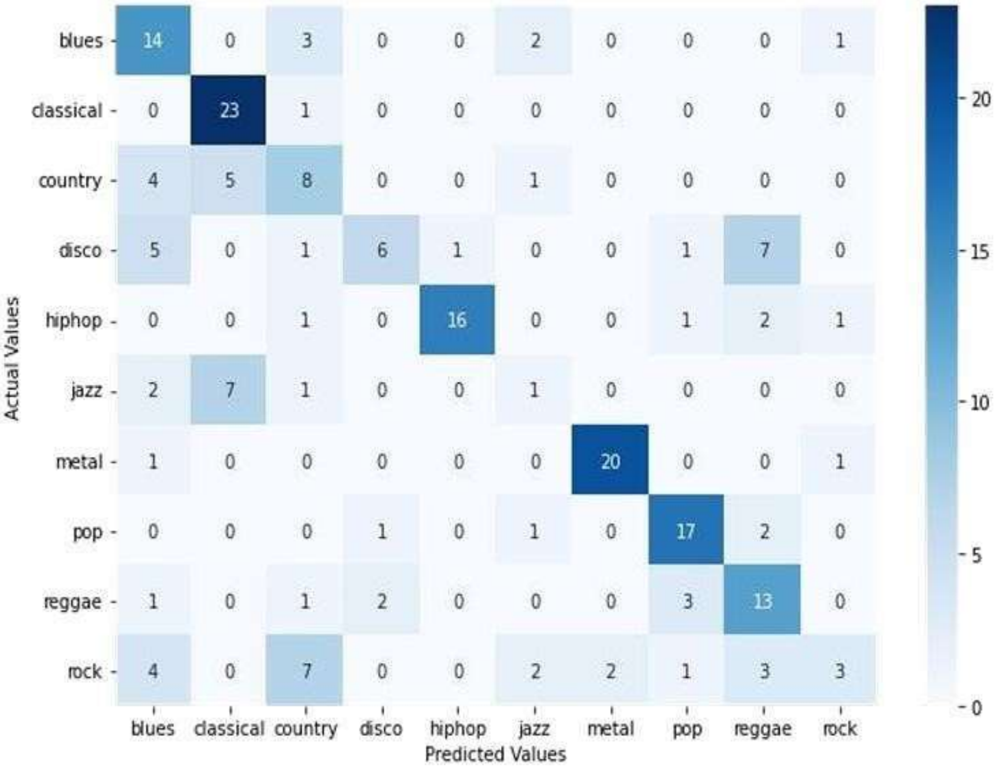


```
25/25 [==============================] - 385s 15s/step - loss: 2.1918 - accuracy: 0.1527 - val_loss: 2.0032 - val_accuracy: 0.3250
Epoch 4/20
25/25 [==============================] - 371s 15s/step - loss: 2.0383 - accuracy: 0.2566 - val_loss: 1.7676 - val_accuracy: 0.3400
Epoch 5/20
25/25 [==============================] - 374s 15s/step - loss: 1.8577 - accuracy: 0.3229 - val_loss: 1.7499 - val_accuracy: 0.4100
Epoch 6/20
25/25 [==============================] - 380s 15s/step - loss: 1.8122 - accuracy: 0.3229 - val_loss: 1.4607 - val_accuracy: 0.4550
Epoch 7/20
25/25 [==============================] - 353s 14s/step - loss: 1.5975 - accuracy: 0.4218 - val_loss: 1.3970 - val_accuracy: 0.4900
Epoch 8/20
25/25 [==============================] - 354s 14s/step - loss: 1.4838 - accuracy: 0.4393 - val_loss: 1.3578 - val_accuracy: 0.4900
Epoch 9/20
25/25 [==============================] - 352s 14s/step - loss: 1.4842 - accuracy: 0.4568 - val_loss: 1.3676 - val_accuracy: 0.5300
Epoch 10/20
25/25 [==============================] - 353s 14s/step - loss: 1.3786 - accuracy: 0.4756 - val_loss: 1.1537 - val_accuracy: 0.5800
Epoch 11/20
25/25 [==============================] - 353s 14s/step - loss: 1.2847 - accuracy: 0.5282 - val_loss: 1.1728 - val_accuracy: 0.5850
Epoch 12/20
25/25 [==============================] - 353s 14s/step - loss: 1.2530 - accuracy: 0.5282 - val_loss: 1.3633 - val_accuracy: 0.4750
Epoch 13/20
25/25 [==============================] - 355s 14s/step - loss: 1.1625 - accuracy: 0.5770 - val_loss: 1.1473 - val_accuracy: 0.5850
Epoch 14/20
25/25 [==============================] - 354s 14s/step - loss: 1.0774 - accuracy: 0.6058 - val_loss: 1.1615 - val_accuracy: 0.6150
Epoch 15/20
25/25 [==============================] - 357s 14s/step - loss: 1.0430 - accuracy: 0.6070 - val_loss: 1.0231 - val_accuracy: 0.6600
Epoch 16/20
25/25 [==============================] - 352s 14s/step - loss: 1.0288 - accuracy: 0.6083 - val_loss: 1.1721 - val_accuracy: 0.6100
Epoch 17/20
25/25 [==============================] - 350s 14s/step - loss: 0.8936 - accuracy: 0.6671 - val_loss: 1.1628 - val_accuracy: 0.6050
Epoch 18/20
25/25 [==============================] - 351s 14s/step - loss: 0.9142 - accuracy: 0.6446 - val_loss: 1.0485 - val_accuracy: 0.6900
Epoch 19/20
25/25 [==============================] - 351s 14s/step - loss: 0.7316 - accuracy: 0.7359 - val_loss: 1.2666 - val_accuracy: 0.6300
Epoch 20/20
25/25 [==============================] - 355s 14s/step - loss: 0.7279 - accuracy: 0.7196 - val_loss: 1.2929 - val_accuracy: 0.6050
```

Figure 6: Output Screen2

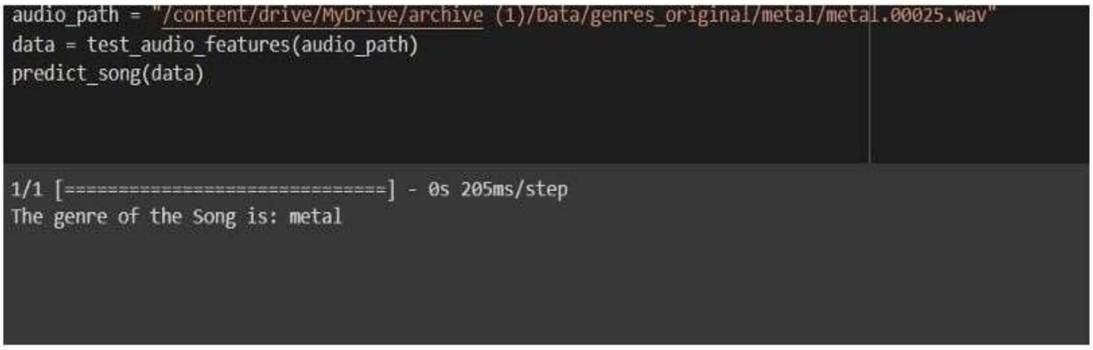
IV. RESULT

Executing a music genre classification project involves collecting and preprocessing audio data, extracting features, and selecting an appropriate model architecture such as CNNs or RNNs. After training the model with augmented data and validating its performance using metrics like accuracy and F1 score, the model is fine-tuned through hyperparameter optimization. Once the model achieves satisfactory performance, it is deployed for real-time genre classification. Continuous monitoring and periodic retraining ensure the model remains accurate and robust in production. This comprehensive approach enhances music organization, personalized recommendations, and overall user experience on music platforms.



	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	14	0	3	0	0	2	0	0	0	1
classical	0	23	1	0	0	0	0	0	0	0
country	4	5	8	0	0	1	0	0	0	0
disco	5	0	1	6	1	0	0	1	7	0
hiphop	0	0	1	0	16	0	0	1	2	1
jazz	2	7	1	0	0	1	0	0	0	0
metal	1	0	0	0	0	0	20	0	0	1
pop	0	0	0	1	0	1	0	17	2	0
reggae	1	0	1	2	0	0	0	3	13	0
rock	4	0	7	0	0	2	2	1	3	3

Figure 7: Result Screen1



```
audio_path = "/content/drive/MyDrive/archive (1)/Data/genres_original/metal/metal.00025.wav"
data = test_audio_features(audio_path)
predict_song(data)
```

```
1/1 [==============================] - 0s 205ms/step
The genre of the Song is: metal
```

Figure 8: Result Screen2

V. DISCUSSION

The music genre classification project aims to automate the categorization of music tracks into predefined genres using machine learning and deep learning techniques. The project begins with collecting a diverse dataset of labelled audio tracks from sources like GTZAN or FMA. This dataset undergoes preprocessing to ensure uniformity in format and quality, followed by segmentation of audio files if necessary. Feature extraction is a crucial step where audio features such as Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, and spectrograms are derived using tools like Librosa. These features encapsulate essential aspects of the audio signal, facilitating effective model training. The choice of model architecture significantly impacts classification performance. Convolutional Neural Networks (CNNs) are employed to analyses spectrograms, capturing spatial patterns, while Recurrent Neural Networks (RNNs) or Long Short Term Memory (LSTM) networks are used to model temporal dependencies in the audio. Data augmentation techniques such as pitch shifting, time stretching, and adding noise enhance the dataset, making the model more robust. Model training involves splitting the dataset into training, validation, and test sets, optimizing the model with appropriate loss functions and optimizers, and monitoring its performance to prevent overfitting. Evaluation metrics like accuracy, precision, recall, and F1 score, along with cross validation, ensure the model's robustness and reliability. Hyperparameter tuning further refines the model, optimizing parameters such as learning rate and batch size. Upon achieving satisfactory performance, the model is deployed for real-time genre classification. Continuous monitoring and periodic retraining are essential to maintain accuracy and adapt to new data trends.

VI. CONCLUSION

In conclusion, music genre classification using Convolutional Neural Networks (CNNs) has proven to be a promising approach for automatic music genre classification. CNNs are able to learn effective features directly from the raw audio signals and can capture characteristics of music signals, which are essential for music genre classification. Various studies have shown that CNN-based models can achieve high accuracy rates for music genre classification, surpassing traditional feature-based models. However, the performance of CNNs depends on various factors such as the size of the dataset, the quality of the audio data, the architecture of the CNN model, and the choice of hyperparameters. Overall, music genre classification using CNNs is a promising field of research that has the potential to revolutionize the way music is organized and curated in various applications such as music recommendation systems, music libraries, and music streaming services.

VII. FUTURE SCOPE

The feature scope of a music genre classification project encompasses several critical aspects. It begins with robust audio data ingestion, supporting diverse formats and handling large-scale datasets or streaming inputs effectively. Preprocessing steps include normalization, segmentation into uniform clips, and noise reduction to ensure data quality. Feature extraction involves deriving essential audio features like MFCCs, chroma features, and spectrograms, essential for capturing both timbral and structural aspects of music. Data augmentation techniques such as pitch shifting and noise addition enhance dataset diversity and model robustness. Model architecture choices range from CNNs for spatial pattern recognition in spectrograms to RNNs/LSTMs for modelling temporal dependencies, with hybrid models combining these approaches for optimal performance. Training involves validation and test splits, cross-validation for model validation, and evaluation metrics such as accuracy and F1 score. Hyperparameter tuning optimizes learning rates and batch sizes to refine model performance. Deployment ensures real-time classification capability, supported by a user-friendly interface and integration with music platforms for personalized recommendations. Continuous monitoring and periodic retraining maintain model accuracy and adaptability to evolving music trends, providing valuable analytics and insights into genre distribution and listener preferences.

VIII. ACKNOWLEDGEMENT

Kandhati Tulasi Krishna Kumar: Training & Placement Officer with 15 years' experience in training & placing the students into IT, ITES & Core profiles & trained more than 9,500 UG, PG candidates & trained more than 350 faculty through FDPs. Authored 5 books, Guided 40+ papers in international journals for the benefit of the diploma, pharmacy, engineering & pure science graduating students. He is a Certified Campus Recruitment Trainer from JNTUA, did his Master of Technology degree in CSE from VTA and in process of his Doctoral research. He is a professional in Pro-E, CNC certified by CITD He is recognized as an editorial member of IJIT (International Journal for Information Technology & member in IAAC, IEEE, MISTE, IAENG, ISOC, ISQEM, and SDIWC. He published articles in various international journals on Databases, Software Engineering, Human Resource Management and campus Recruitment Training.

Ms. Bharathi Erothi is pursuing her final semester MCA in Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Deep Learning Ms. Bharathi has taken up her PG project on Music Genre Classification and published the paper in connect to the project under the guidance of K. Tulasi Krishna Kumar, associate professor, SVPEC.

IX. REFERENCES

9.1 Book References

[1] A book on Classification. Music and Books on Music: M: Music; ML: Literature of Music; MT: Musical Instruction and Study by O. G. Sonneck_and_Herbert Putnam_linked: http://surl.li/guycky

[2] A book on Lyrics on Several Occasions by Ira Gershwin linked: http://surl.li/vocatq

[3] A book on Music Production | 2024+ Edition: The Professional Studio Guide for Producers, Songwriters, Artists & Audio Mastering Engineers by Tommy Swindali_linked: http://surl.li/jtopkq

[4] A book on Music Notebook Journal by Michelle J. Ciesielka linked: http://surl.li/unmjto

[5] A book on Mastering Audio: The Art and the Science by Bob Katz_linked: http://surl.li/lqpeya

[6] A book on Music Theory[15] by Nicolas Carter linked: http://surl.li/tmotjm

[7] A BOOK ON MASTER THE RECORDING STUDIO BY STERLING SKYE LINKED: HTTP://SURL.LI/QCTKOR

[8] A BOOK ON CHAMBER MUSIC [16] BY KUIN ROGER LINKED: HTTP://SURL.LI/DJHJER

9.2 Web Reference

[9] A Web-Based Music Genre Classification for Timeline Song Visualization and Analysis by Jaime Ramirez Castillo in IEEE linked: https://ieeexplore.ieee.org/abstract/document/9333553

[10] A Music genre classification and music recommendation by using deep learning by A. Elbir linked: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/el.2019.4202

[11] A web reference on Music Genre Classification GitHub linked: https://github.com/crgoku7/MusicGenreClassification

[12] A web reference on Music Genre Classifier using Machine Learning by geeks for geeks linked: https://www.geeksforgeeks.org/music-genre-classifier-using-machine-learning

9.3 Article Reference

[13] An article reference on Deep Learning Based Music Genre Classification Using Spectrogram by Athulya KM
linked: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3883911

[14] An article on A Machine Learning Approach to Automatic Music Genre Classification by Alessandro L. Koerich linked: https://link.springer.com/article/10.1007/BF03192561

[15] An article on Music Genre Classification using Machine Learning Techniques by Hareesh Bahuleyan linked: https://arxiv.org/abs/1804.01149

[16] An article on Music genre classification with taxonomy by Tao Li linked: https://ieeexplore.ieee.org/abstract/document/1416274

[17] An article on Multimodal deep learning for music genre classification by Nieto Caballero and Oriol
linked: https://repositori.upf.edu/browse?type=author&value=Nieto+Caballero%2C+Oriol

[18] A paper reference on Genre Classification in Music using Convolutional Neural Networks by Sandeep Kumar Dash linked: https://link.springer.com/chapter/10.1007/978-981-99-73392_33