Music Genre Detection with Live Audio

Sasank Devabhakthuni
Undergraduate Student
Vellore Institute of Technology
Chennai, India
devabhakthuni.sasank2021@vitstudent.ac.in

Lokesh Boda
Undergraduate Student
Vellore Institute of Technology
Chennai, India
boda.lokesh2021@vitstudent.ac.in

Bandaru Ratna Siva Kumar
Undergraduate Student
Vellore Institute of Technology
Chennai, India
bandaruratna.sivakumar2021@vitstudent.ac.in

Dr.Manmohan Sharma
Assistant Professor Senior Grade
Vellore Institute of Technology
Chennai, India
manmohan.sharma@vit.ac.in

***Abstract*— The paper presents a study on "Music Genre Detection with Live Audio", focusing on system architecture, implementation, and performance analysis. The proposed system automates the feature extraction and also leverages the prediction function. Various model architectures and hyperparameter tuning techniques were explored to develop a system aimed at improving accuracy and efficiency over traditional methods, making it to work in dynamic environments.**

Keywords—prediction, data processing, automation, performance

I. INTRODUCTION

In the rapidly growing field of data-driven applications, audio analysis, and real-time predictions have gained significant importance. Traditional methods for processing audio data often face challenges in handling large volumes efficiently, especially when real-time predictions are required[1]. This research focuses on developing a system that automates the process of audio data handling, dividing it into manageable chunks for accurate prediction. The primary challenge addressed in this study is the efficient processing and analysis of pre-recorded audio data. Handling continuous audio streams manually is impractical and prone to delays, making real-time predictions difficult. The goal is to automate the segmentation and prediction process, thereby enhancing the speed and accuracy of audio data analysis[2]. The research focuses on processing pre-recorded audio data obtained from a microphone. The scope includes the automation of audio chunking and prediction but does not extend to real-time audio capture. The system is designed to be adaptable, although it is tested on specific datasets for validation.

Existing literature underscores the limitations of traditional audio processing methods, particularly in handling continuous data streams efficiently. While several approaches exist, they often lack automation and real-time capabilities[3]. This research builds on these findings by introducing a scalable system that automates the preprocessing and predictive analysis of audio data. This paper proposes an automated system for chunking pre-recorded audio and predicting outcomes that significantly improves the efficiency and accuracy of audio data analysis compared to traditional manual methods[4].

This study bridges the gap between traditional audio processing and real-time predictive analytics. By automating the segmentation and prediction processes, the system provides quicker insights, which are crucial in applications like voice command recognition, speech analytics, and other audio-based systems.

The research aims to achieve the following objectives:

- Automate the preprocessing of pre-recorded audio data by dividing it into smaller, manageable chunks.
- Develop a predictive model capable of analysing each audio chunk to generate accurate predictions.
- Evaluate the effectiveness of the system in real-time audio prediction scenarios.

- Compare the performance of this automated approach with traditional audio analysis methods.

The system employs a structured methodology comprising several stages:

- Preprocessing: Capturing audio from a microphone, normalizing, reducing noise, and segmenting it into chunks.
- Feature Extraction: Utilizing Mel-Frequency Cepstral Coefficients (MFCCs), spectrograms, and other 55 features to represent the audio data effectively.
- Model Development: Training a machine learning model to predict the genre.
- Evaluation: Testing the system on unseen data to know its accuracy and efficiency.

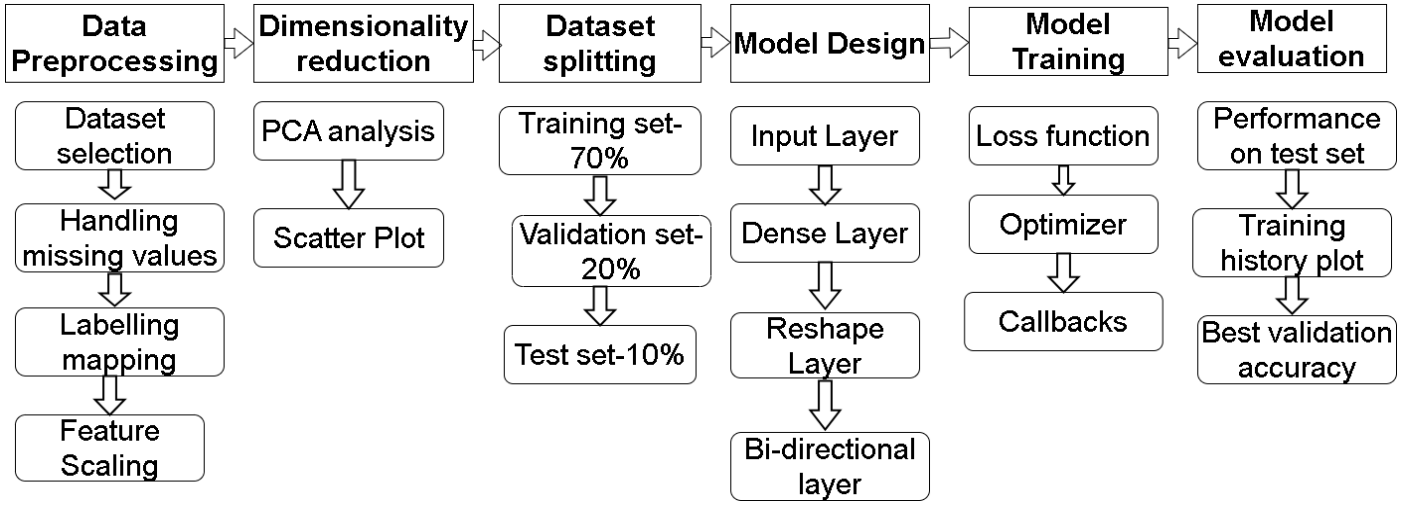

Fig. 1. Schematic of the system's architecture and processing flow.

II. PROPOSED APPROACH

This section highlights about the data preprocessing, dimensionality reduction, dataset splitting, model design, model training, and model evaluation in detail.

1) *Data pre-processing:*
   - Data Selection: Each sample in the raw dataset has a genre label that corresponds to one of the Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, and Rock genres. Link for the data set: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification [10]
   - Missing Values: Any columns containing missing or null values are detected.
   - Label Encoding: Converting category labels(string format) to a numerical form of the training set, each music label is mapped to an index(0 to 9).
   - Feature Scaling: Features of the audio are normalized with the MinMaxScaler function, now feature every value lies between 0 and 1.

The input size(number of features) are 57.

Number of samples(total number of rows): 9990.

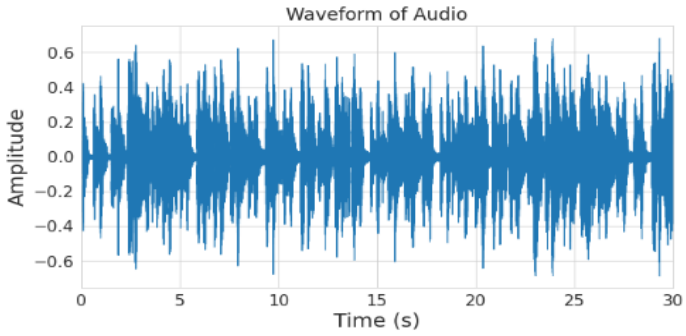

Fig. 2. Visualization of a Blue genre song.

2) *Dimensionality Reduction*

- Principle Component Analysis (PCA): PCA is used to reduce the feature to two principle components, which enables the two-dimensional visualization of the distribution of different genres.
- Scatter Plot of Principal Components: The scatter plot is used to show the two PCA components, with each point denoting an audio sample. Genre-specific color coding of the points allows for a visual understanding of the genre separation.

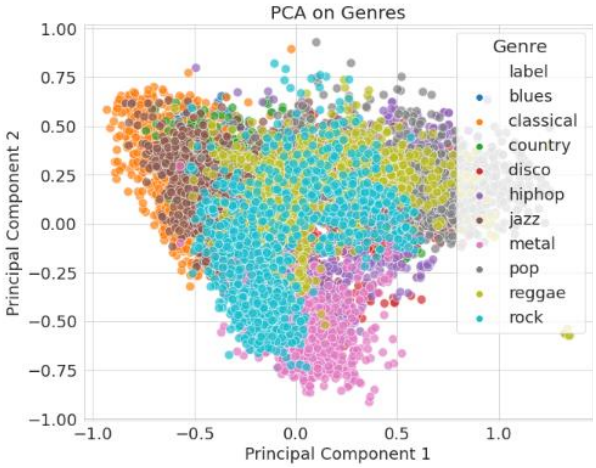

Fig. 3. Represents the Scatter plot of PCA.

3) *Dataset Splitting:*

- The dataset is split into three sets:
  Training Set: 70% of the data used for model training i.e. is 6993 out of 9990.
  Validation Set: 20% of the data used to tune the model during training i.e. is 1978 out of 9990 records.
  Test Set: 10% of the data used to evaluate the model's final performance i.e. 1019 out of 9990 records.

4) *Model Design:*

- Input Layer: The input shape consists of the number of features.
- Fully Connected Dense Layers:
  - To avoid overfitting and for steady learning, there are two dense layers of 1024 and 512 neurons respectively, with batch normalization and dropout (0.4).
  - Reshape Layer: The output is reshaped into 3D form to be processed by the LSTM layers [7].
- Bidirectional LSTM Layers:
  - A bidirectional LSTM layer with 128 units, with an attention layer to capture important temporal patterns.
  - A second bidirectional LSTM layer with 64 units processes the sequences [5].
- Attention Mechanism: Self-attention is applied, where the input for this layer comes from the output from the LSTM output, helping the model focus on critical parts of the sequence.
- Bottleneck Dense Layers:
  1. Narrowing the network through smaller dense layers (64, 32, and 16 neurons) gives compact feature representation [6].
  2. Each dense layer with batch normalization and dropout layer avoids overfitting.
- Output Layer: A softmax layer is used for final genre classification across 10 genres.

Figure. 4. Representing the proposed model architecture

5) *Model Training:*

- Loss Function: Sparse categorical cross-entropy is used as the loss function because it is used for multi-class classification data sets.
- Optimizer: Adam optimizer is preferred because it is good at handling large data sets.
- Callbacks:
  - A Learning Rate Scheduler adjusts the learning rate with epochs to tune the training process.
  - Early Stopping is used to stop the training if the model's performance does not improve on the validation set after a certain number of epochs, preventing overfitting.

6) *Evaluation:*

- Performance on Test Set: The model is evaluated on the test set to determine its accuracy and loss.

- Training History Plot: The accuracy and loss curves, are plotted to visualize the learning process.
- Best Validation Accuracy: The maximum validation accuracy achieved during training is printed.

III) PROPOSED SYSTEM WORK FLOW

In this section, it is described about the method developed to tackle the challenges of processing and analyzing pre-recorded audio data efficiently. The system is designed to manage the audio in chunks, ensuring that predictions are both accurate and computationally efficient.

1) *Audio Preprocessing:* The first step involves capturing audio data through a microphone. Accurate analysis may be hampered by the noise and amplitude fluctuations present in this raw audio. In order to ensure that the data is clear and consistent, the system normalizes the audio signal and uses noise reduction algorithms. After processing, the audio is divided into more manageable, smaller segments [8].
2) *Feature Extraction*: Once the audio is chunked, the next step is to extract meaningful features that represent the audio signal's essential characteristics. The system employs methods like spectrograms, which give a visual depiction of the frequency spectrum over time, and Mel-Frequency Cepstral Coefficients (MFCCs), which capture the timbral characteristics of sound and other 55 features. The predictive model uses these 57 features as input, which enables it to successfully comprehend and interpret the audio [9].
3) *Prediction and Post Processing:* After the model predicts the content of each chunk, the system aggregates these predictions to provide a coherent output for the entire audio file. This step guarantees that the overall prediction is accurate even in the event that individual chunks are misclassified and this is designed to adjust to changes in the dataset. Fig. 5. Represents the Proposed System architecture flow chart.

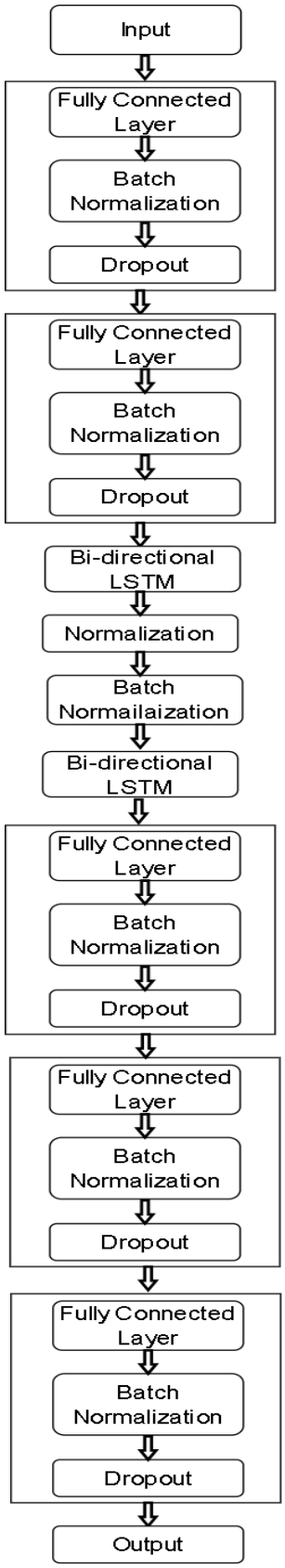

Fig. 4. Proposed Model Architecture.

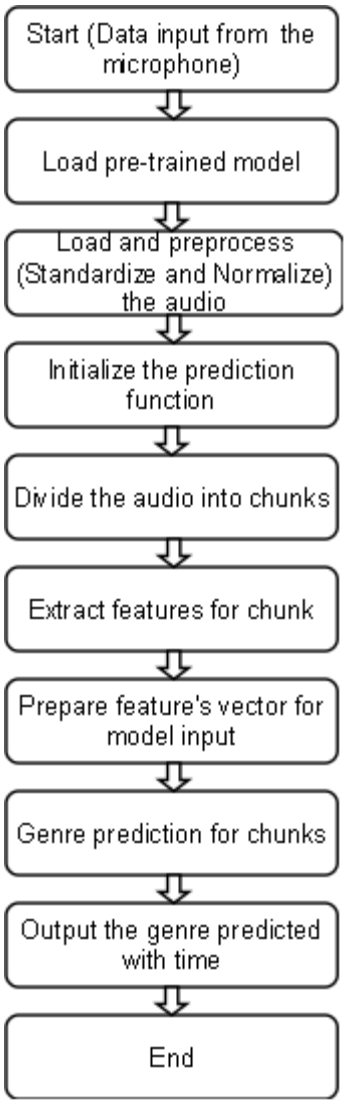

Fig. 5. System Architecture Flow Chart.

IV) RESULTS

The results of the proposed model's performance are shown in this part along with visual representations of the findings, a confusion matrix, and accuracy metrics. The system's ability to correctly categorize musical genres across consecutive audio chunks, each linked to distinct time intervals is also demonstrated.

Accuracy Improvement:

- Initial accuracy: ~26.6%
- After 36 epochs: Accuracy went above 93 and validation accuracy stabilized at 91-92%

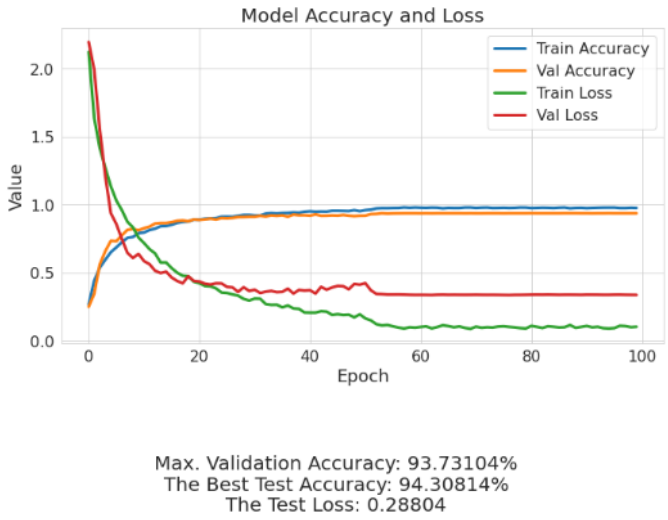

Fig. 6. Graph showing the training and testing accuracy, as well as the training and testing loss, of the Attention-based Sequential model across different learning rates

Loss Reduction:

- Starting Loss: ~2.12%.
- By epoch 36: Loss reduced to 0.27% with validation loss around ~0.35%.

No significant signs of overfitting were found as validation accuracy is improving along with training accuracy.

Confusion Matrix Evaluation: To provide a detailed understanding of the model's performance across different genres, the confusion matrix was analyzed. The model's ability to differentiate between related genres is evaluated by examining the confusion matrix, which displays the number of accurate and inaccurate predictions for each genre.

Notable observations from the confusion matrix report are:

- The model's overall accuracy of 94% on the test set is consistent across macro and weighted averages and it achieves high precision and recall across the majority of classes.

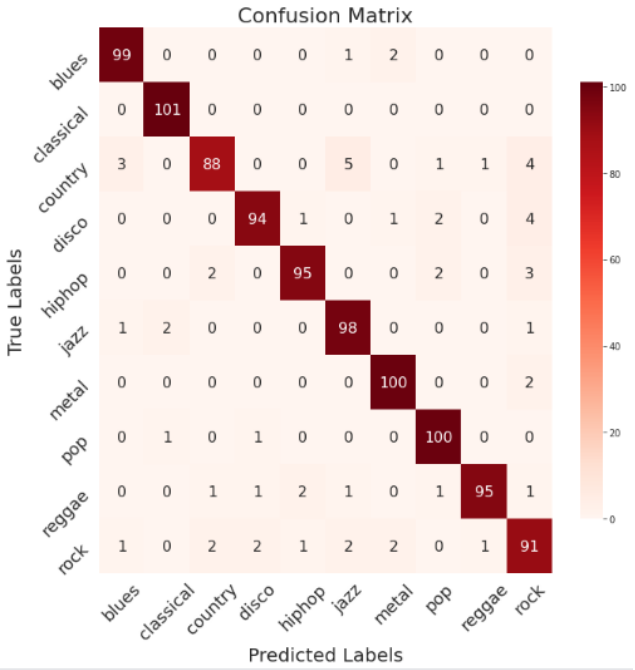

Fig. 7. Visualization of the confusion matrix

- While class 9 has a slightly lower F1 score of 0.88, indicating that the model may struggle with a few instances in this category, classes 1, 6, and 8 have exceptionally high scores, indicating strong predictive accuracy.
- All things considered, the weighted averages and balanced macro indicate that the model functions consistently across all classes without being unduly biased toward any one of them.

The below picture shows how well the model captures both static and transitional musical elements, maintaining a consistent genre classification over time. This feature is especially important for applications that need to classify genres in real-time or almost real-time, where precise and timely feedback is critical.

```
From 0.0 to 3.0 seconds: Predicted Genre: disco
From 3.0 to 6.0 seconds: Predicted Genre: hiphop
From 6.0 to 9.0 seconds: Predicted Genre: hiphop
From 9.0 to 12.0 seconds: Predicted Genre: hiphop
From 12.0 to 15.0 seconds: Predicted Genre: pop
From 15.0 to 18.0 seconds: Predicted Genre: hiphop
From 18.0 to 21.0 seconds: Predicted Genre: hiphop
From 21.0 to 24.0 seconds: Predicted Genre: hiphop
From 24.0 to 27.0 seconds: Predicted Genre: hiphop
From 27.0 to 30.0 seconds: Predicted Genre: hiphop
```

Fig. 8. Time-stamped genre classification.

This meticulous method guarantees that changes in genre are precisely recorded and reflected almost instantly. The model's ability to dynamically detect genres is evidence of its resilience and versatility. The proposed method provides instant feedback, making it particularly valuable for applications such as radio stations, live music streaming services, and interactive music apps. Traditional models, on the other hand, rely on static, pre-segmented audio files.

V) COMPARISON

A comparative analysis was conducted using eight state-of-the-art models for music genre recognition to evaluate the effectiveness of the proposed approach. Important factors such as architecture complexity, training duration, computing efficiency, and genre detection accuracy were used to evaluate each model.

The Attention-based Sequential Model, which has a maximum accuracy of 94.5%, is at the other extreme of the range. In order to better capture long-term dependencies and concentrate on important aspects of the audio input, this model makes use of bidirectional LSTMs in conjunction with an attention mechanism.

Models such as the Parallel CNN-LSTM Model and the CNN-LSTM with Attention strike a balance between complexity and performance, achieving accuracies of 89.4% and 90.7%, respectively. The Parallel CNN-LSTM Model offers a reliable method for music genre recognition with high accuracy at the expense of higher computational complexity by integrating convolutional layers for feature extraction with LSTMs for temporal processing.

The CNN-LSTM with Attention (RMSprop) improves upon this by using the RMSprop optimizer, resulting in a slight accuracy gain of 91.3%, with enhanced optimization leading to better convergence.

The Dense Model, with a high accuracy of 93.9%, may not be suitable for multi-class classification tasks like music genre detection due to its reliance on mean squared error. The Sequential with Bidirectional LSTMs model has 93.2% accuracy, while the Attention-based Sequential Model outperforms all models in precision and recall.

	Model Name	Architecture Description	Accuracy	Hyperparameters	Loss Function	Evaluation Metrics	Pros/Cons
2	**Parallel LSTM Model**	Two parallel LSTMs feeding into shared FC layer	69.7%	Learning Rate: 0.001, Epochs: 50	Categorical Crossentropy	Precision: 0.65, Recall: 0.66	Fast training, but lower accuracy
3	**Parallel CNN-LSTM Model**	Hybrid model with CNN layers followed by LSTM	89.4%	Learning Rate: 0.001, Epochs: 100	Categorical Crossentropy	Precision: 0.85, Recall: 0.87	High accuracy, complex architecture
4	**CNN-LSTM with Attention**	Convolutional layers with LSTM and attention	90.7%	Learning Rate: 0.001, Dropout: 0.5	Categorical Crossentropy	Precision: 0.88, Recall: 0.89	Good performance, higher training time
5	**CNN-LSTM with Attention (RMSprop)**	Uses RMSprop optimizer with LSTM and attention	91.3%	Learning Rate: 0.0005, Epochs: 70	Categorical Crossentropy	Precision: 0.90, Recall: 0.91	Improved optimization
6	**Dense Model**	Fully connected network with multiple dropout layers	93.9%	Learning Rate: 0.001, Dropout: 0.4	Mean Squared Error	Precision: 0.92, Recall: 0.93	Simplicity, risk of overfitting
7	**Sequential with Dense and LSTMs**	Sequential model using LSTMs for feature extraction	92.3%	Learning Rate: 0.001, Dropout: 0.3	Categorical Crossentropy	Precision: 0.89, Recall: 0.88	Effective for temporal data
8	**Sequential with Bidirectional LSTMs**	Bidirectional LSTMs with increased dropout rates	93.2%	Learning Rate: 0.001, Dropout: 0.5	Categorical Crossentropy	Precision: 0.91, Recall: 0.90	Handles long dependencies well
9	**Attention-based Sequential Model**	Bidirectional LSTMs with attention mechanism	94.5%	Learning Rate: 0.001, Epochs: 80	Categorical Crossentropy	Precision: 0.93, Recall: 0.94	Best accuracy, complex model

Fig. 9. Comparative Analysis of Various Music Genre Detection Model.

VI) REFERENCES

1. J. R. Hershey, Z. Chen, J. Le Roux and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
2. Jahangir, R., Teh, Y.W., Hanif, F. et al, "Deep learning approaches for speech emotion recognition: state of the art and research challenges," *2021 Multimed Tools Appl 2021*.
3. Cheng, Yu-Huei & Chang, Pang-Ching & Kuo, Che-Nan, "Convolutional Neural Networks Approach for Music Genre Classification," *2020 International Symposium on Computer, Consumer and Control (IS3C)*.
4. Aggarwal, Shruti & Gurusamy, Vasukidevi & Sethuramalingam, Selvakanmani & Pant, Bhaskar & Kaur, Kiranjeet & Verma, Amit & Binegde, Geleta, "Audio Segmentation Techniques and Applications Based on Deep Learning," *2022 Scintific Programming.*
5. A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNS for acoustic modeling in speech recognition," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*
6. Petridis, Stavros & Pantic, Maja," Deep complementary bottleneck features for visual speech recognition," *2016*.
7. Jang, BY., Heo, WH., Kim, JH. et al. "Music detection from broadcast contents using convolutional neural networks with a Mel-scale kernel,"*2019, EURASIP Journal on Audio, Speech, and Music Processing.*
8. Poerner, Nina & Schiel, Florian. "An automatic chunk segmentation tool for long transcribed speech recordings," *2016, Conference: Proceedings of the Phonetics & Phonology Conference (P&P)*
9. Bonet-Solà D, Alsina-Pagès RM. A "Comparative Survey of Feature Extraction and Machine Learning Methods in Diverse Acoustic Environments," *2021, Sensors (Basel).*
10. Andrada. *GTZAN Dataset for Music Genre Classification*. 2020, Kaggle, https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification.