Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2024) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

9th International Conference on Computer Science and Computational Intelligence 2025 (ICCSCI 2025)

Comparison of MFCC and Spectrogram Analysis in Improving the Accuracy of CNN Models for Song Clip Prediction

First Author[a], Second Author[a], Third Author[a], Fourth Author[a], Fifth Author[a]

[a]*School of Computer Science, Binus University, Jakarta, Indonesia*

Abstract

In automatic music recognition, choosing the right feature extraction method can make a big difference in how well a model performs. This study compares two popular techniques—Mel-Frequency Cepstral Coefficients (MFCC) and Spectrograms—to see which one works better with Convolutional Neural Networks (CNNs) for predicting short song clips. The audio data was first preprocessed, segmented, and augmented, and then features were extracted using both MFCC and Spectrogram approaches. Separate CNN models were trained on each type of input. While the MFCC-based model performed reasonably well, achieving 84.2% accuracy on the test set, the model trained with Spectrograms performed significantly better, reaching 94% test accuracy along with higher precision, recall, and F1 scores. These results suggest that Spectrograms are more effective at capturing the intricate patterns in audio, making them a stronger option for accurate song classification.

Click here and insert your abstract text.

Peer-review under responsibility of the scientific committee of the 10th International Conference on Computer Science and Computational Intelligence 2025

Keywords: MFCC; Spectrogram; CNN; Song Clip Prediction; Audio Classification; Feature Extraction; Deep Learning; Music Recognition

1. Introduction

In this digital era, automatic music recognition has been rising significantly, especially with the increasing number of music streaming services such as Spotify, SoundCloud, and Apple Music, as well as song identification tools like Shazam. Convolutional Neural Networks (CNNs) have demonstrated strong performance in audio classification, but the choice of feature extraction methods can significantly impact the model's accuracy. In this case, Mel-Frequency Cepstral Coefficients (MFCC) and spectrograms are the most used techniques, each with its own advantages and disadvantages. However, their effectiveness in improving CNN performance remains an area of research. This study aims to compare the performance, differences, and effectiveness of MFCC and spectrograms in enhancing the

Peer-review under responsibility of the scientific committee of the 10th International Conference on Computer Science and Computational Intelligence 2025

accuracy of CNN-based song clip prediction. While past research has explored these methods in voice recognition and music genre classification, limited studies have focused on comparing their effectiveness in identifying song clips. This experiment involves applying both MFCC and spectrogram methods to a dataset, training CNN models, and analyzing their accuracy. The goal of this research is to determine which technique is more effective and provides better predictive performance. The findings can contribute to optimizing classification techniques in real-world applications, such as music search engines.

2. Literature Review

Different spectral and rhythm features for CNN-based audio classification are examined, comparing mel-scaled spectrograms, mel-frequency cepstral coefficients (MFCCs), cyclic tempograms, and other chromagram-based features. The results show that mel-scaled spectrograms and MFCCs achieve the highest classification accuracy, with training accuracies of 94.06% and 93.88%, respectively, while features like cyclic tempograms and chromagrams yield lower performance. These findings emphasize the importance of selecting the most effective features to optimize CNN-based audio classification, particularly in applications such as speech recognition and music analysis [1].

SpectNet, a deep learning framework for audio classification that features a learnable spectrogram extraction layer. Unlike traditional spectrograms with fixed filterbank parameters, SpectNet employs a trainable front-end based on gammatone filters, allowing it to adaptively extract spectrogram features optimized for specific classification tasks. The study assesses SpectNet on two datasets—heart sound anomaly detection and acoustic scene classification—reporting a 1.02% improvement in MACC for heart sound classification and a 2.11% accuracy boost in acoustic scene classification. These findings underscore the benefits of learnable spectrogram features in enhancing CNN-based audio classification [2].

The performance of various machine learning models—CNN, VGG16, and XGBoost—in music genre classification is analyzed using Mel-Frequency Cepstral Coefficients (MFCC) and mel spectrograms. Evaluations on the GTZAN dataset show that XGBoost, when utilizing 3-second MFCC features, outperforms CNN and VGG16, achieving a 97% testing accuracy. This suggests that tabular representations of audio features can significantly enhance classification performance with tree-based models. Additionally, the findings highlight the importance of data segmentation in improving CNN-based classification accuracy, demonstrating that short-segment MFCCs are more effective than full-length mel spectrograms for genre classification[3].

The comparative effectiveness of Mel-Frequency Cepstral Coefficients (MFCC) and spectral representation in voice recognition tasks is analyzed by extracting features from a voice dataset and utilizing the K-Nearest Neighbors (K-NN) classification algorithm. The findings indicate that MFCC significantly outperforms spectral representation in terms of accuracy, achieving a training accuracy of 84.18% and a testing accuracy of 74.71%. These results highlight MFCC's superior capability in capturing both the frequency and temporal characteristics of human speech, making it a more effective feature extraction approach for voice recognition than spectral representation[4].

The combination of Convolutional Neural Networks (CNN) with various Recurrent Neural Network (RNN) variants, including LSTM, Bi-LSTM, GRU, and Bi-GRU, has been explored for music genre classification using the GTZAN dataset. When comparing the performance of Mel-frequency cepstral coefficients (MFCCs) and Mel-spectrograms, the latter generally outperformed MFCCs, with the CNN+Bi-GRU hybrid model achieving the highest accuracy of 89.30%. This highlights the effectiveness of integrating spatial and temporal modeling in deep learning for music analysis and suggests that Mel-spectrograms are a powerful feature for genre classification[5].

In the study of raga classification within Indian classical music, researchers explored how well MFCC and Mel spectrogram techniques performed when used to train a CNN model. Both methods were applied to a diverse dataset of five Hindustani ragas. While MFCC delivered reliable results, the Mel spectrogram stood out by achieving higher accuracy in training, validation, and testing. This suggests that spectrograms—thanks to their image-like

representation—are better at capturing the rich tonal patterns and timing details that define musical pieces. These insights highlight the growing appeal of using visual audio features in deep learning, especially for recognizing and classifying complex musical structures like ragas[6].

3. Research Methodology

This study employs a systematic experimental approach to evaluate MFCC and spectrogram features for audio classification. The methodology consists of four key phases: (1) Research Design, (2) Dataset Collection * Preprocessing, (3) Feature Extraction, (4) Data Splitting, (5) CNN Model Architecture, (6) Model Evaluation, and (7) Comparative Analysis between MFCC and Spectrogram. Each phase is designed to ensure reproducible and objective results.

3.1. Research Design

The research compares two techniques of extracting audio features—MFCC and Spectrogram analysis—to see which one helps CNN predict songs more accurately from short audio snippets. It follows data preparation, feature extraction, CNN model training, testing, and comparison of the result.

3.2. Dataset Collection & Preprocessing

The data has short song clips saved in MP3 format. The clips are taken from diverse sources and converted to WAV format for feature extraction and training the model. The data preprocessing is done in the following steps.

- MP3 to WAV Conversion

  MP3 files are converted to WAV format with metadata like the artist's name and song title intact. These modified files are stored in an organized way, and corresponding labels are stored for training.
- Audio Segmentation

  Each WAV file is segmented into 5-second clips, making sure multiple segments of the songs to enhance diversity of the data availabke and improve the model of the training.
- Data Augmentation

  Several augmentation methods are used to make the dataset richer and more generalizable, such as:
  1. Time Stretching: Altering the playback speed without altering the pitch.
  2. Pitch Shifting: Changing the pitch upward or downward to create variations.
  3. Noise Addition: Including background noise to render the model more robust.

3.3. Feature Extraction

MFCC Extraction: From each segmented audio file, MFCC features are extracted to retain its frequency and temporal dynamics. The features are preprocessed and labeled for training the CNN model.

Spectrogram Extraction: Spectrograms are generated by converting audio signals into time-frequency images. These features are used instead of MFCCs to train the CNN model.

3.4. Data Splitting

To train and evaluate the models effectively, the dataset is divided as below:

- Training Set: 60%
- Validation Set: 20%
- Test Set: 20%

3.5. CNN Model Architecture

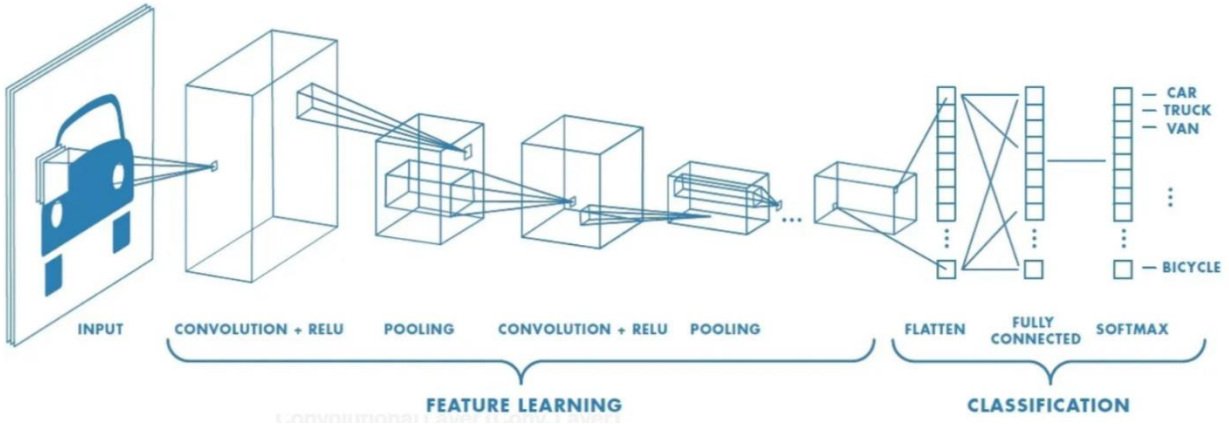

Fig. 1 CNN Layer Architecture

The CNN model follows a structured design for optimal performance:

- Input Layer: Processes extracted features from MFCC and Spectrogram.
- Convolutional Layers: Extract patterns and relevant features from the input.
- Pooling Layers: Reduce dimensionality while saving key information.
- Batch Normalization: Stabilizes the learning process and improves training efficiency.
- Fully Connected Layers: Perform classification based on extracted feature maps.
- Output Layer: Uses softmax activation to classify song clips into each category.

Avoid hyphenation at the end of a line. Symbols denoting vectors and matrices should be indicated in bold type. Scalar variable names should normally be expressed using italics. Weights and measures should be expressed in SI units. All non-standard abbreviations or symbols must be defined when first mentioned, or a glossary provided.

3.6. Model Evaluation

To fully evaluate the performance of our CNN-based song clip prediction model is, we will be using a mix of normal and deep learning performance measures/metrics. We will evaluate it based on:

- Model Accuracy & Loss: Evaluating training and validation accuracy/loss across epochs.
- Precision: Indicates the proportion of correctly predicted positive instances out of all positive predictions, reflecting the model's reliability in classification tasks.
- Recall: Measures the model's ability to identify all relevant instances of each class, representing its effectiveness in capturing true positives.
- F1 Score: Provides a balanced evaluation by combining precision and recall into a single metric, particularly useful in cases of class imbalance.

3.7. Comparative Analysis: MFCC vs. Spectrogram

To determine the superior feature extraction method, two separate models are trained:

- CNN trained with MFCC features
- CNN trained with Spectrogram features

Performance metrics are analyzed to conclude which technique yields better accuracy and efficiency in predicting song clips.

4. Result and Discussion

4.1. CNN model using MFCC

The CNN model is built with 13 layers and has a total of 94,588 parameters, taking up about 369 KB of space. It starts with three convolutional blocks (Conv2D), each followed by max-pooling and batch normalization layers. As the model goes deeper, it reduces the output shape from (18, 9, 24) down to (2, 1, 48), while learning more detailed features. The first convolutional layer has 624 parameters, and the deeper layers expand to 28,848 and 57,648 parameters, capturing increasingly complex patterns.

After flattening the features, the model uses two dense layers with 6,208 and 780 parameters to handle classification. Dropout is applied here to help prevent overfitting. Interestingly, out of the total parameters, 94,348 (about 368.5 KB) are trainable, while only 240 (about 960 bytes) are non-trainable, coming from the batch normalization layers. This shows the model is efficiently designed to focus on learning useful features. Overall, this compact yet powerful architecture strikes a good balance between complexity and efficiency, making it a solid fit for audio classification tasks.

The model consists of a deep architecture composed of layers. The convolutional layers become deeper to discover increasingly complex patterns. The fully connected layers towards the end gather the discovered features and classify them into classes. Dropout is utilized to make the model more effective, preventing it from fitting the training data too closely.

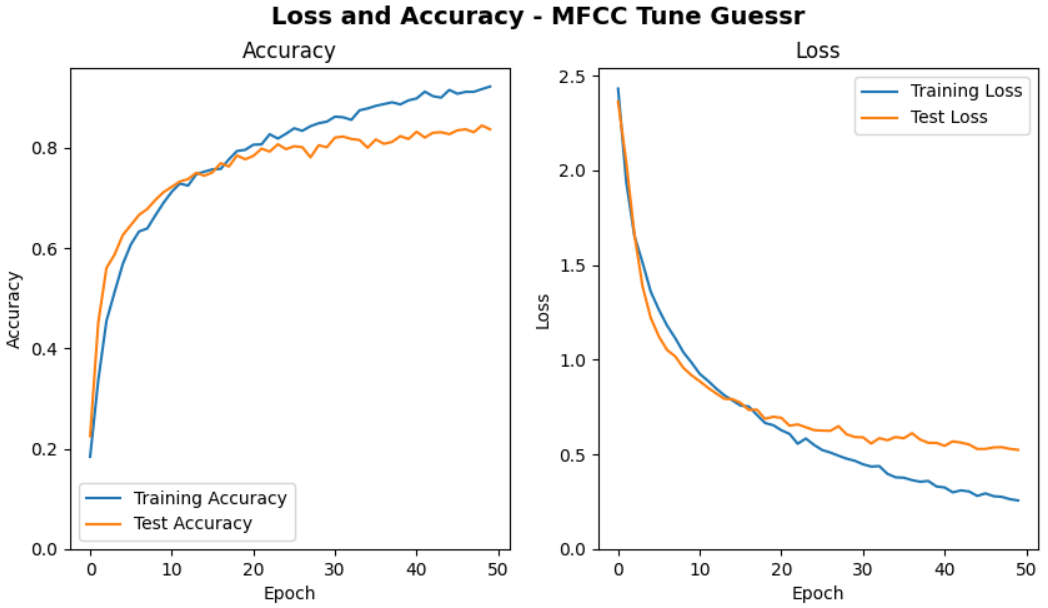

Fig. 2 MFCC Loss Accuracy

The behavior of the CNN model was also examined using training and validation accuracy/loss plots. The accuracy plot reveals a consistent upward trend in both training and validation accuracy, ultimately converging to approximately

84.2% on the test data after 50 epochs. The minimal gap between training and validation accuracy suggests that the model exhibits strong generalization abilities when exposed to unseen data, indicating it is not simply memorizing the training examples but rather learning to generalize to new, unseen data.

The loss plot shows a steady decline in both training and validation loss, which demonstrates that the model is learning effectively over time. After a certain number of epochs, the validation loss plateaus, signaling that the model has reached an optimal level of learning and is no longer improving significantly, indicating that it has adequately learned the patterns in the data.

The closeness of the training and validation accuracy curves, which do not show much divergence, further suggests that overfitting is not a serious concern. In the case of overfitting, the training accuracy would continue to increase while the validation accuracy would stay constant or even decrease. Here, the accuracy curves remain quite close, which signifies the model is not overfitting to the training data.

Underfitting, which would be reflected in both training and validation accuracy being low, is not present in this model. On the contrary, the accuracy achieved is relatively high, which indicates that the model is learning appropriately and capturing the underlying patterns in the data.

The final evaluation of the model on the test dataset resulted in a Test Loss of 0.5257 and a Test Accuracy of 84.2%. These results show that the CNN model is successfully able to learn meaningful representations from MFCC features and has been properly trained, with minimal overfitting, leading to a solid performance on the test data.

4.2. CNN Model using Spectrogram

The CNN model is built with 13 layers and packs around 3.86 million parameters (about 14.7MB), making it much stronger than our baseline version. It starts with three convolutional layers (ranging from 624 to 57,648 parameters each) that gradually extract richer audio features, with pooling and batch normalization layers helping to streamline and stabilize the process. After that, the network flattens the data and feeds it into a large dense layer with over 1.1 million parameters, allowing for deep pattern recognition, followed by dropout to prevent overfitting and a final layer for classification. Out of all the parameters, only 240 are fixed, while 1.29 million are actively trainable, supported by an advanced optimizer managing another 7.57 million settings. This high capacity lets the model perform highly detailed audio analysis — though it does require more computing power to fully unlock its potential.

The model is a deep architecture composed of multiple layers. At the beginning of the model, several convolutional layers (Conv2D) are used to extract basic features from the input data. As the data passes through deeper layers, the model becomes capable of recognizing increasingly complex patterns. These convolutional layers are accompanied by MaxPooling2D layers to reduce the spatial dimensions of the data while retaining the essential features.

In between the convolutional layers, BatchNormalization is applied to stabilize and accelerate the learning process by normalizing the activations. Towards the end, the model utilizes a Flatten layer to convert the multi-dimensional outputs into a one-dimensional vector, which is then processed by Dense layers. These fully connected layers gather the features discovered by the convolutional layers and make final class predictions.

To prevent overfitting, Dropout layers are strategically placed in the model. These layers randomly set a fraction of the input units to zero during training, forcing the model to generalize better and avoid memorizing the training data.

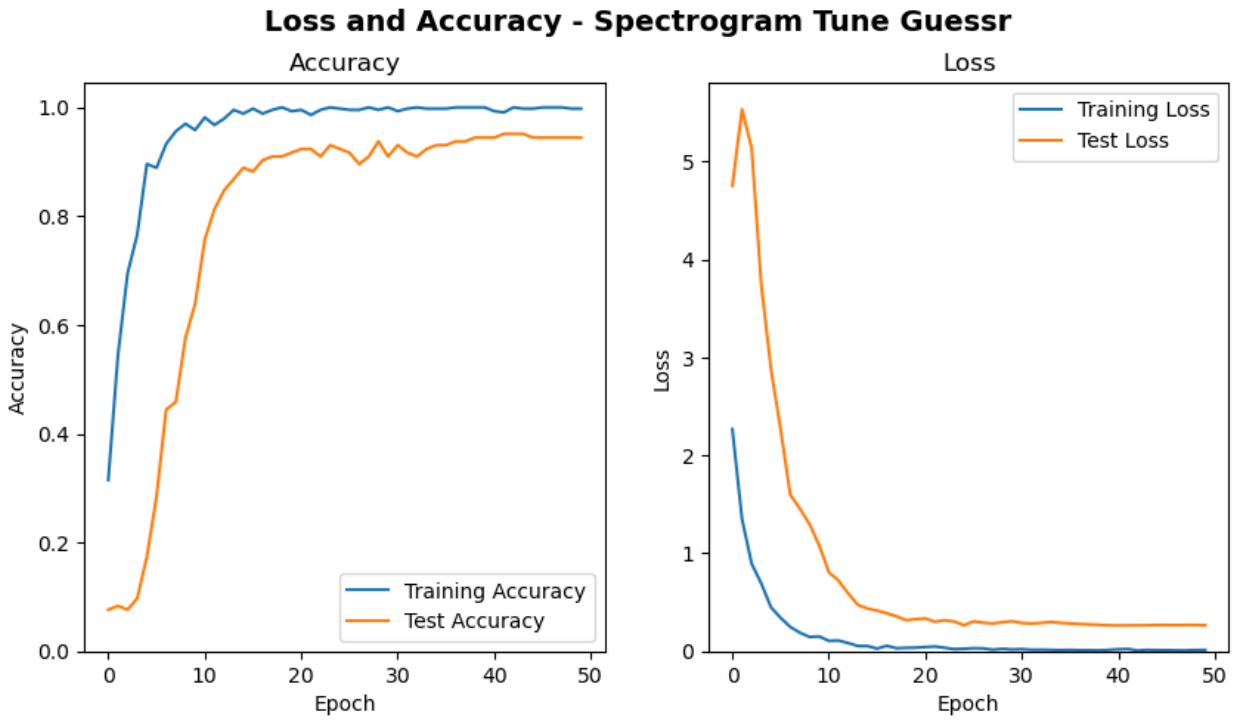

Fig. 2 Spectogram Loss Accuracy

The accuracy plot indicates that the model achieved an impressive overall accuracy of 94%, which demonstrates its ability to classify the test data correctly. This high accuracy reflects the model's robust learning, capturing meaningful features from the Spectrogram input.

The precision, recall, and f1-score are also calculated for each class. The precision of the model is consistently high across all classes, suggesting that when the model predicts a class, it is highly accurate. This indicates that the model was able to classify these classes with no false positives.

The recall, which measures the ability of the model to identify all relevant instances of each class, is also strong. Most classes have a recall of 1.00, meaning the model successfully identified nearly all the true instances.

The macro average precision, recall, and f1-score are all 0.95, 0.94, and 0.94 respectively, further confirming the model's overall strong performance across all classes. The weighted average precision, recall, and f1-score of 0.95, 0.94, and 0.94 suggest that the model performed well even when accounting for the class distribution.

In terms of the loss, the training and test losses demonstrate the effectiveness of the model. The loss curves show that both the training and validation loss steadily decreased and converged after around 50 epochs, with minimal divergence between them, indicating that the model is effectively learning and generalizing without significant overfitting or underfitting.

In conclusion, the results indicate that the CNN model using Spectrogram input has been properly trained, with strong accuracy and little overfitting. The model is capable of distinguishing between various music classes, achieving high precision, recall, and f1-scores across all classes.

4.3. Comparison Between MFCC and Spectrogram

Table 1. MFCC and Spectrogram comparison.

Stats	MFCC	Spectrogram
Final Accuracy	0.85	0.94
Precision	0.86	0.95
F1	0.86	0.94
Recall	0.86	0.94

The CNN model using MFCC has a total of 94,588 parameters, with 94,348 trainable parameters and 240 non-trainable parameters. It consists of convolutional layers (Conv2D), fully connected layers (Dense), and dropout layers for regularization. MaxPooling2D layers are used to downsample the spatial dimensions, and BatchNormalization is applied to stabilize and accelerate the training process.

In contrast, the CNN model using Spectrogram has a much larger architecture with 3,859,094 parameters. Of these, 1,286,284 are trainable parameters, and 240 are non-trainable. This model also uses convolutional layers (Conv2D), MaxPooling2D layers for downsampling, BatchNormalization for normalization, Flatten to reshape the data, Dense layers for classification, and Dropout for regularization. The larger number of parameters in the Spectrogram model allows it to potentially capture more complex patterns and features, but it also requires careful attention to avoid overfitting.

In terms of training behavior, the MFCC model achieved a test accuracy of 84.2% after 50 epochs, with a test loss of 0.5257. The model exhibited minimal overfitting, as seen in the training and validation accuracy plots, which converged, showing strong generalization. The Spectrogram model, on the other hand, achieved a significantly higher test accuracy of 94%. The loss curves for both training and validation loss steadily decreased and converged after 50 epochs, indicating the model's effective learning and generalization. This model demonstrated robust learning, suggesting it captured more detailed features from the Spectrogram input.

In conclusion, the Spectrogram model outperforms the MFCC model in terms of overall test accuracy, precision, recall, and F1-scores. The Spectrogram method's ability to capture more detailed information from the audio data enables it to perform better, especially for more complex classification tasks. While the MFCC model remains an effective and simpler approach for scenarios with limited computational resources or data, the Spectrogram model offers superior performance for tasks requiring higher accuracy and the ability to distinguish between more complex patterns.

References

[1] F. Wolf-Monheim, "Spectral and Rhythm Features for Audio Classification with Deep Convolutional Neural Networks," arXiv preprint arXiv:2410.06927, 2024.

[2] M. I. Ansari and T. Hasan, "SpectNet: End-to-end audio signal classification using learnable spectrograms," arXiv preprint arXiv:2211.09352, 2022.

[3] Y. Meng, "Music Genre Classification: A Comparative Analysis of CNN and XGBoost Approaches with Mel-frequency cepstral coefficients and Mel Spectrograms," arXiv preprint arXiv:2401.04737, 2024.

[4] R. A. N. Diaz, N. L. G. P. Suwirmayanti, and K. Budiarta, "PERBANDINGAN KUALITAS PENGENALAN SUARA UNTUK EKSTRAKSI FITUR MENGGUNAKAN MFCC DAN SPECTRAL," Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika, vol. 6, no. 1, pp. 58–63, 2024.

[5] M. Ashraf et al., "A hybrid cnn and rnn variant model for music classification," Applied Sciences, vol. 13, no. 3, p. 1476, 2023.

[6] D. Joshi, J. Pareek, and P. Ambatkar, "Comparative study of Mfcc and Mel spectrogram for Raga classification using CNN," Indian J Sci Technol, vol. 16, no. 11, pp. 816–822, 2023.

Author Contributorship

XXX: Conceptualization, Methodology, Software Development, Data Preprocessing, Feature Extraction, Model Training, Formal Analysis, Writing – Original Draft Preparation; XXX: Literature Review, Data Augmentation, Model Evaluation, Visualization, Writing – Review & Editing, Writing – Original Draft Preparation; XXX**:** Experimental Design, Hardware/GPU Resource Management, Statistical Validation, Results Interpretation, Writing - Discussion Section, Manuscript Proofreading, Writing – Original Draft Preparation; XXX: Writing – Review & Editing, Supervision, Project administration, Funding acquisition. XXX: Writing – Review & Editing, Supervision, Project administration, Funding acquisition.

Data Availability

The authors: We demonstrate our commitment to data openness and transparency. To facilitate further research, we have made the used in their study publicly available.

The data used by the author can be opened via the link below:

- https://github.com/AryoBaskoro/tune-guessr