

Musical Guitar Tuning using Machine Learning Approach for Custom Data using FFT and MFCC

Shashidhar R¹, Yash P Jain², Rahul V Tirlapur³, Prakash Srivastava⁴, Chander Prabha⁵ and Niranjana Athreya M⁶

^{1-3,6}Department of Electronics and Communication Engineering, JSS Science and Technology University, Mysuru, India
Email: shashidhar.r@jssstuniv.in, yashpjain3@gmail.com, rahulvtirlapur@gmail.com, mniranjana18@gmail.com

⁴Department of Computer Science & Engineering, Graphic Era (Deemed to be University), Dehradun, India.
Email: prakash2418@gmail.com

⁵Chitkara University Institute of Engineering and Technology, Chitkara University Punjab, India
Email: prabhanice@gmail.com

Abstract— The proposed model explores the application of machine learning to automate the tuning process for musical instruments, focusing especially on guitars. This approach involves extracting detailed audio features from the guitar using Mel-Frequency Cepstral Coefficients (MFCC) and Fast Fourier Transform (FFT). MFCCs capture the timbral characteristics of the audio signal by analyzing its short-term power spectrum. This method provides a detailed representation of the phonetic attributes essential for musical note identification. By using MFCC, It can effectively capture the nuances of each string's sound and tone quality. FFT transforms the audio signal from the time domain into the frequency domain, permitting us to examine its frequency components and harmonics. This transformation is very crucial for distinguishing between the fundamental frequencies and detecting deviations from standard tuning. The features extracted from MFCC and FFT are then apply random forest machine learning algorithms to train models for detecting tuning nuances. These models are designed to recognize deviations from the expected pitch and provide corrective adjustments, ensuring that each string is tuned precisely. system processes the audio signal to maintain its original quality, making adjustments to align each string perfectly with its intended pitch. This model demonstrates the combining of MFCC and FFT with machine learning techniques and offers a simple solution for automated guitar tuning. The Accuracy vs number of trees was found to be 92.11% and that of Loss vs number of trees is 7.89%. The overall tuning accuracy of all the strings was found to be 98.27%.

Keywords—Fast Fourier Transform (FFT), Power Spectral Density (PSD), Musical Tuning, Mel-Frequency Cepstral Coefficients (MFCC),MATLAB.

I. INTRODUCTION

The human auditory system is capable of hearing between 20 Hz and 20 kHz, processing complex sound mixtures and perform auditory scene analysis, allowing us to navigate environments by distinguishing voices or instruments. However, it often falls short in detecting fine pitch differences, crucial for musicians who need perfectly tuned notes. Musicians face challenges in tuning instruments accurately due to factors like personal fatigue, environmental noise, and individual hearing sensitivity. Untuned instruments can lead to disharmony in ensembles, reduced sound quality, and increased difficulty in performing complex pieces.

This lack of precision can cause stress and frustration for musicians, impacting their confidence and performance. To aid musicians, electronic tuners and software applications are used, relying on precise estimation of a soundwave's fundamental frequency.

Power Spectral Density (PSD) estimation techniques are essential for determining the frequency components in a signal. Common methods include the Welch, periodogram, covariance, and Burg methods, each with advantages and limitations.

The Fast Fourier Transform (FFT) is a powerful process underlying many PSD techniques, efficiently calculating the Discrete Fourier Transform of a signal. It converts signals from the time domain to the frequency domain, revealing frequency content for filtering and analysis.

Major Contributions to the anticipated work are as:

- Develop the custom musical data using Guitar.
- Insert the audio file into the MATLAB in mp3 format and Read the audio file using the read instruction
- Select the required frequency for tuning as a fundamental frequency and preprocess the audio signal using FFT
- Develop the machine learning algorithm to train and test the custom data.
- Compare the anticipated result with existing outcome.

II. RELATED WORK

A. Developments in Musical Signal Tuning

The main focus was on comparing two similar audio datasets, testing the impact of noise removal on model performance. Among the various machine learning models tested, the Artificial Neural Network (ANN) model achieved the highest accuracy, around 91%, making it the most effective for our purposes [1]. Nihat Pamuk in her paper discussed the frequency element rate of the indicator impending from the atmosphere is engaged in actual time and matched with the occurrence standards of the records in the databank organized previously in the LabVIEW program.

As an outcome of the association, the expanse of the identified occurrence from the focus frequency of the note in the record is visually testified to the customer [2].

B. Overview of various techniques

This paper showed that the Welch Power Spectral Density Investigation is employed to identify the essential frequency of the note, which is then associated with a chord frequency chart to find the instrument's tuning status—sharp if the note's frequency exceeds the chord's fundamental frequency and flat if it falls below [3]. Using frequency analysis, the note that corresponds to the frequency result is obtained. The resulting analysis, which is the frequency relationship between the notes, is displayed. The characteristics of this analysis are then concluded [4].

The author discussed the fundamental frequency of a particular guitar string and then compared the frequency with the standard frequency [5]. Discussed the Arduino uno to automatically tune a Yamaha FX310 guitar with stepper motors, The Arduino calculates the signal frequency and, if it is incorrect, adjusts the corresponding string's tuning peg using a 28BYJ 48 stepper motor controlled by a PNP transistor [6].

The Strong Karplus Algorithm, used in Easy Tuner, simulates plucked string sounds by looping a short waveform through a delay filter [7].

Modern smartphones using the YIN algorithm offer accurate pitch detection and a Java-based app compares frequencies to musical notes, calculating deviations and providing real-time tuning feedback through a graphical interface [8]. The Fast Fourier Transform (FFT) procedure simplifies the complex process of performing the Discrete Fourier Transform (DFT), enabling efficient transformation of time-domain data into the frequency domain [9]. This paper discusses the current study builds upon previous research, employing a modified FFT algorithm to analyze Latotou tones and compare them against reference tones from a guitar, achieving correct identification of instrument tones through signal processing on a computer [10].

This paper reviews the Fourier transform changing the signal from the time area to the frequency domain and vice versa [11]. The feature extraction of sound by extraction of sound by recognizing the timbre of the sound [12]. A comprehensive review has considered an embedded structure, using a typical microcontroller 8051, to attain ordinary position frequencies for three dissimilar octaves. The production sign is habituated to attain nigh sinewave and replicated into sound using an acoustic amplifier [13]. A pre-recorded tabla sound that can be adjusted for pitch, tempo, and rhythm to meet the musician's needs [14,15]. The tuning process provides both audio and visual feedback [16,17]. Recognition of the tune for local languages [18,19].

III. METHODOLOGY

A stepwise enlightenment of the anticipated process is agreed sideways with the mathematical demonstration of the augmented model intended for training and testing the custom database. Figure 1 shows the proposed model of the system. Figure 1 depicts the step-by-step procedure of the proposed model in a very modest way. The block diagram shows both the training and testing parts of the proposed method.

A. Block diagram

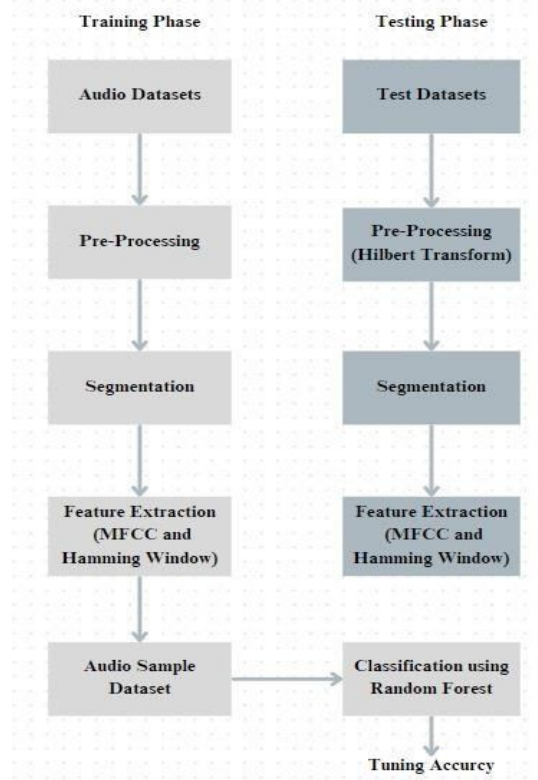


Figure1. Diagram of Projected Technique

B. Procedure and algorithm

Audio Data Input: Collect MP3 audio files from specific subfolders for each of the six guitar strings. Organize these files to ensure that each subfolder corresponds to a particular string and contains multiple samples.

Pre-processing: Convert all audio files to mono to simplify processing. Ensure each sample is trimmed to a time duration of 2 seconds.

Feature Extraction: Hamming window is used to extract Mel-Frequency Cepstral Coefficients (MFCCs) from each pre-processed audio segment. It captures key spectral features of the audio, which are essential to differentiate between different guitar strings.

$$\omega(t) = 0.54 - 0.46 \cos(2\pi t / N - 1) \quad (1)$$

$$\text{MFCC}(a_i) = \text{DCT}(\log(\text{Melspectrogram}(a_i))) \quad (2)$$

Frequency Analysis: Bandpass filter is applied to isolate the fundamental frequency of each audio sample. Fast Fourier Transform is used to translate the audio signal to a frequency domain, identifying the important frequency corresponding to the guitar string's pitch.

Dataset Preparation: Extracted MFCC features are combined to the corresponding string labels.

Split this dataset into training and testing sets with an 80/20 ratio to simplify effective model training and calculation.

Model Training: Train the model using Random Forest, using different numbers of trees, ranging from 1 to 100. Evaluate training accuracy and test accuracy on a separate test set to understand how well the model generalizes to new data.

Model Evaluation: Plot the training and validation accuracy plots to identify the best possible number of trees for the Random Forest model. Analyze these plots to find the best configuration for achieving high classification accuracy.

Performance Analysis: Generate and analyze a confusion matrix using the Random Forest model with the highest number of trees.

The confusion matrix provides a detailed view of classification performance, highlighting how well the model differentiates between different guitar strings.

Output: Report the final accuracy of the tuning system based on the best-performing model. This output summarizes the effectiveness of the tuning procedure and provides insights into its precision and reliability.

Architecture

Audio Data Input

Step 1: Audio files are loaded from specified directories.

Let AudioData be the collection of audio {ai} for each guitar string i ∈ {1,2,...,6}.

Convert stereo to mono if necessary.

Mono audio: ai = mean(ai, axis=2) (average across stereo channels).

Preprocessing

Step 2: Normalize audio length to 2 seconds.

If Length(ai) > fs × audioDuration, then

ai = ai[1: fs × audioDuration] (truncate).

If Length(ai) < fs × audioDuration, then

ai = pad(ai, fs × audioDuration) (pad with zeros).

Feature Extraction

Step 3: Apply the Hamming window and extract MFCCs.

Window function:

$$\omega(t) = 0.54 - 0.46\cos\left(\frac{2\pi t}{N-1}\right) \quad (1)$$

where t is the time index and N is the window length.

➤ Compute MFCCs using:

$$MFCC(ai) = DCT(\log(MelSpectrogram(ai))) \quad (2)$$

where DCT is the Discrete Cosine Transform and MelSpectrogram is derived from FFT.

Frequency Analysis

Step 4.1: Apply a bandpass filter to isolate fundamental frequencies.

Mel filter bank

$$f_{mel} = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \quad (3)$$

Step 4.2: applying discrete cos transform (DCT) to convert a sequence of Mel filter bank energies Sm into the cepstral coefficients Cn.

Discrete cos transform

$$C_n = \sum_{m=0}^{M-1} S_m \cdot \cos\left(\frac{\pi n(m+0.5)}{M}\right) \quad (4)$$

Step 4.3: Perform FFT to obtain frequency domain representation.

➤ FFT:

$$X(f) = \sum_{n=0}^{N-1} e^{-j\frac{2\pi f n}{N}} x(n) \quad (5)$$

Where X(f) is the frequency domain signal and x(n) is the time domain signal.

Data Set Representation

Step 5.1: Combine MFCC features with labels.

➤ Create a dataset {(MFCCs, labels)}.

Step 5.2: Split the dataset into training and testing sets.

- Training set: $\{(X_{train}, Y_{train})\}$ (80% of data)
- Testing set: $\{(X_{test}, Y_{test})\}$ (20% of data).

Model Training

Step 6.1: Train Random Forest models with varying numbers of trees.

- Bootstrap Sampling:

$$D_i = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \quad (6)$$

where D_i is the dataset for the i -th tree, sampled with replacement from the original dataset D .

- Decision Tree Prediction:

$$h_i(x) = Tree_i(x) \quad (7)$$

where $h_i(x)$ is the prediction of the i^{th} decision tree for input x .

- Random Forest Prediction: For classification:

$$\hat{y} = \{h_1(x), h_2(x), \dots, h_n(x)\} \quad (8)$$

where m is the number of trees and mode refers to the most frequent prediction. For Regression:

$$\hat{y} = \sum_{i=1}^m h_i(x) \quad (9)$$

where $\hat{y}$ is the final prediction by averaging the predictions of all trees.

Step 6.2: Calculate out-of-bag (OOB) accuracy for training data

- OOB Accuracy:

$$TrainAccuracy = \frac{\sum(OOBPredictions == T_{train})}{length(T_{train})} \times 100 \quad (10)$$

Model Evaluation

Step 7.1: Predict labels for the testing set using trained models.

- Predictions:

$Y_{pred} = \text{predict}(\text{ModelT}, X_{test})$

Step 7.2: Compute validation accuracy

- Validation accuracy:

$$Accuracy_T = \frac{\sum(Y_{pred} == Y_{test})}{length(Y_{test})} \times 100 \quad (11)$$

Step 7.3: Plot training and validation accuracies.

Performance Analysis

Step 8.1: Generate a confusion matrix for the model with the highest number of trees.

- Confusion Matrix: A matrix performs the amounts of accurate positives, accurate negatives, false positives, and false negatives.

Step 8.2: Plot and analyze the confusion matrix.

Outcome

Step 9.2: Report the final tuning accuracy.

- Final accuracy:

$FinalAccuracy = ValAccuracy_{max}$

Where $ValAccuracy_{max}$ is the accuracy of the model with the optimal number of trees.

IV. OUTCOME AND DISCUSSIONS

In outcome of the anticipated work displayed the training and testing using epoch details.

Figure 2 shows the loss, which measures the error in the model's predictions, changes over 100 epochs during the training and validation phases. Initially, both the training and validation losses are high, indicating that the model is making significant errors in its predictions. However, as training progresses, the loss decreases sharply within

the first 20 epochs, stating that the model is learning and improving its accuracy as shown in Figure 2. After this point, the loss stabilizes and decreases more slowly, eventually levelling off. The close alignment between the training loss and validation loss throughout the epochs suggests that the model is generalizing well to new data and is not overfitting, meaning it maintains good performance on both the training and validation datasets.

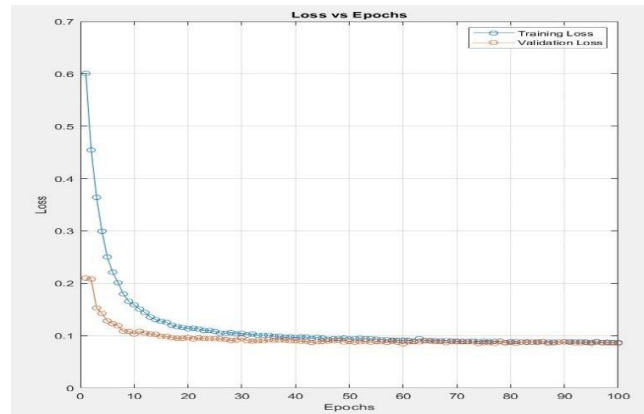


Figure 2. Model Loss vs Epochs

Figure 3 shows how the accuracy of the model's predictions on both the training and validation data improves over 100 epochs. The accuracy starts low, reflecting the model's initial difficulty in making correct predictions. However, similar to the loss, the accuracy improves rapidly in the first 20 epochs as the model learns from the data. After reaching around 80-90% accuracy, the rate of improvement slows, and the accuracy stabilizes, indicating that the model has learned effectively and is making correct predictions at a high and consistent rate.

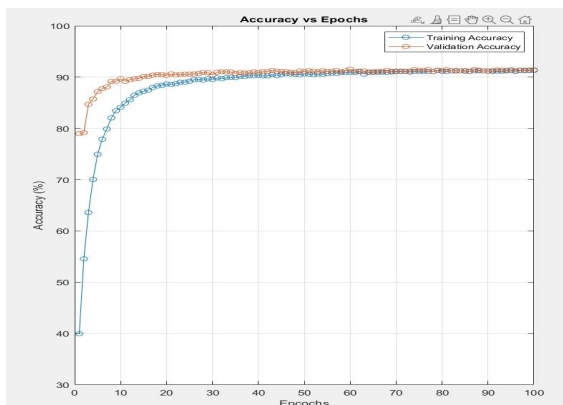


Figure 3. Model Accuracy vs Epochs

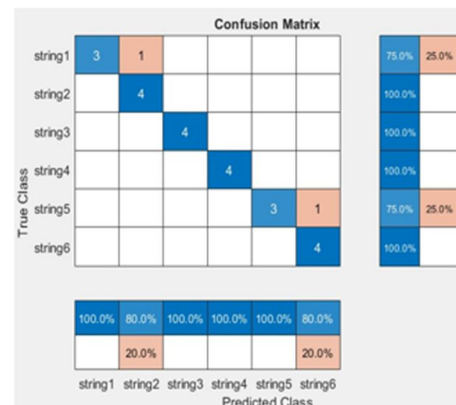


Figure 4. Confusion matrix of the proposed method

Figure 4 represents the confusion matrix, which shows how well the model has predicted each guitar string class by comparing true class labels (rows) with predicted labels (columns). The diagonal values represent correct predictions, while rows of the corresponding columns indicate errors. This model has correctly classified most strings, with minor misclassifications which are, 'string1' was incorrectly predicted as 'string2' once, and 'string5' was misclassified as 'string3' once. Overall, this matrix has made strong predictions of the strings testing samples, with the majority of predictions falling on the diagonal, indicating accurate string identification and total test samples add up to 24 on adding all the diagonal elements.

TABLE I. CLASSIFICATION MATRIX OF THE PROPOSED METHOD

String	Class	Precision	F1-Score	Recall	provide Support
String 1	String1	1	0.8571	0.7500	4
String 2	String2	0.8000	0.8889	1	4
String 3	String3	1	1	1	4
String 4	String4	1	1	1	4
String 5	String5	1	0.8571	0.7500	4
String 6	String6	0.8000	0.8889	1	4

TABLE II. COMPARISON TABLE OF THE PROPOSED METHOD WITH

Method	Random Forest	Random Forest
Custom Dataset	Pre-recorded Audio Samples [1]	Custom dataset with audio samples recorded from each guitar string individually
Accuracy	92.05%	92.11%
Error Rate	7.95%	7.89%

The above two tables give insight into the classification matrix and the Comparison table. Table I give the division of strings and their attributes. Precision, F1-score, and Recall of each string is almost 1. Support for each string is a perfect 4 as 4 strings have been taken for testing out of 20 samples for each string. Table II provides a comparison of the existing method to the proposed method and it is seen that the accuracy of the proposed method is almost the same as that of the existing method.

V. CONCLUSION AND FUTURE SCOPE

The anticipated work successfully demonstrates a modern approach to guitar string tuning by blending signal processing with machine learning. It processes audio samples from guitar strings and uses MFCCs to extract key features that help in distinguishing between different strings. By applying a Random Forest classifier to these features, it can classify the string types with high accuracy. The process is started by using the custom data set, gathering the audio samples, normalizing the duration of each audio sample, and using a Hamming window to extract feature, then data collected is divided into two proportions of specified ratio for training and testing to evaluate the performance of the proposed method. The model is trained using many trees and the accuracy is measured using confusion matrix and feature matrix. This approach shows the improvement in the Guitar tuning process when the traditional digital processing is combined with Machine learning techniques. The high accuracy shows the system's efficiency.

The future scope of this work is in exploring alternative machine learning techniques and features such as Chroma features, Spectral Contrast, or using deep learning to provide better insights and accuracy. CNN or RNN can be used as they can capture more complex audio patterns which can better and detailed analysis and might help to fine-tune the output, different types of guitar samples and frequencies can be incorporated to make the model more generalized so that the application increases. Limitations of the model are that it is limited only to one single music instrument changes are needed to make it more versatile. A better model is required for better output with better accuracy.

REFERENCES

- [1] Gourisaria, M.K., Agrawal, R., Sahni, M. et al. Comparative analysis of audio classification with MFCC and STFT features using machine learning techniques. *Discover Internet Things* 4, 1 (2024). <https://doi.org/10.1007/s43926-023-00049-y>
- [2] Nihat Pamuk (2022) "Identifying Different Musical Instrument Sounds Using Fourier Analysis in LabVIEW", *SAR Journal*. Volume 5, Issue 4, Pages 175-182, ISSN 2619-9955, DOI: 10.18421/SAR54-02 December 2022.
- [3] Reymore L and Lindsey DT (2025) Color and tone color: audiovisual crossmodal correspondences with musical instrument timbre. *Front. Psychol.* 15:1520131. doi: 10.3389/fpsyg.2024.1520131.
- [4] B. Deo Kumar, A. Kushwaha, A. Kumar and A. Agarwal, "Design & Implementation of Digital Guitar Tuner Using MATLAB," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2021, pp. 547-549, doi: 10.1109/ICACITE51222.2021.9404728.
- [5] S. S. Tippannavar, R. Shashidhar, H. r. Sathvik, S. Varun, G. v. Punith and H. g. Nikshep, "Text Independent Speaker Recognition and Classification using KNN Algorithm", 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), pp. 1199-1205, 2022.
- [6] Soeharwinto, L. Tampubolon, T. H. Nasution and Fahmi, "Design of Automatic Guitar Tuner using 8-bit Microcontroller," 2023 7th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM), Medan, Indonesia, 2023, pp. 311-314, doi: 10.1109/ELTICOM61905.2023.10443179.
- [7] Dahlan Abdullah, Kevin Angkasa, H Hartono and Herman Fithra (2019) "Designing Guitar Tuning Software Using Karplus Strong Algorithm", *Journal of Physics: Conference Series* 1364 (2019) 012027 doi:10.1088/1742-6596/1364/1/012027
- [8] Arnie Gedanken Branta, Emanuel Marques Martins, Isabela Quandt, Solivan Arantes Valente (2019) "Development of a triple input musical instrument tuner using Yin algorithm", *International Journal of Advanced Engineering Research and Science (IJAERS)*, [Vol-6, Issue12, Dec- 2019], ISSN: 2349-6495(P) | 2456-1908(O).
- [9] Jannereth, E., & Esch, L. (2021). Analyzing Timbres of Various Musical Instruments Using FFT and Spectral Analysis. *Journal of Student Research*, 10(1). <https://doi.org/10.47611/jsrshs.v10i1.1292>.

- [10] Kang, S. (2025). Adaptations, Code-Switching, and Novelty with Cultural Integrity: Musicians Performing and Learning Musical Instruments in Different Musical Traditions. *Journal of Research in Music Education*, 0(0). <https://doi.org/10.1177/00224294251317170>.
- [11] E. Palantei, H. Hamid and Dewiani, "A Modified-FFT Algorithm for Determining the Basic Frequency Tone of Latotou Traditional Music Instrument," 2019 International Conference on Electrical Engineering and Informatics (ICEEI), Bandung, Indonesia, 2019, pp. 41-45, doi: 10.1109/ICEEI47359.2019.8988865.
- [12] Palantei, Elyas & Hamid, Helson & Dewiani, Dewiani. (2019). A Modified-FFT Algorithm for Determining the Basic Frequency Tone of Latotou Traditional Music Instrument. 41-45. 10.1109/ICEEI47359.2019.8988865.
- [13] K P, D. (2020). A Standard Reference Audio Tone Generator for Tuning Indian Classical Musical Instruments. *International Research Journal on Advanced Science Hub*, 2(10), 1-3. doi: 10.47392/irjash.2020.180
- [14] U. K. Roy, "Composing recorded Tabla sound to accompany musicians," 2015 Annual IEEE India Conference (INDICON), New Delhi, India, 2015, pp. 1-6, doi: 10.1109/INDICON.2015.7443114..
- [15] B. Bozkurt, "A System for Tuning Instruments Using Recorded Music Instead of Theory-Based Frequency Presets," in *Computer Music Journal*, vol. 36, no. 3, pp. 43-56, Sept. 2012, doi: 10.1162/COMJ_a_00128.
- [16] Shashidhar R, D. Balivada, S. D N, K. Hirebasur Krishnappa and R. M, "Music Emotion Recognition using Convolutional Neural Networks for Regional Languages," 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIIE), Ballari, India, 2023, pp. 1-7, doi: 10.1109/AIKIIIE60097.2023.10390450.
- [17] N. Pavitha, V. Pungliya, A. Raut, R. Bhonsle, A. Purohit, A. Patel, et al., "Movie recommendation and sentiment analysis using machine learning", *Global Transitions Proceedings*, vol. 3, no. 1, pp. 279-284, 2022.
- [18] B. P. A. Prabhu, T. Sharma, R. Dani, and M. S. G. Prasad, "A novel approach to video summarization using AI-GPT and speech recognition," in *Lecture Notes in Networks and Systems*, Singapore: Springer Nature Singapore, 2024, pp. 201–209.
- [19] S. Balla, M. S. Guru Prasad, S. Balla, K. Madhu Krishna, J. Kiran and M. Ram, "Comparative Analysis of Face Emotion Detection based on Deep Learning Techniques," 2023 World Conference on Communication & Computing (WCONF), RAIPUR, India, 2023, pp. 1-5, doi: 10.1109/WCONF58270.2023.10235127..