Automated Raaga Recognition In Carnatic Music Using CNN-BiLSTM Model With Attention Mechanism

K Hemavardhan Reddy
Computer Science- Artificial Intelligence Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
hvardhan1437@gmail.com

KN Lakshmi
Computer Science - Artificial Intelligence Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
lakshminandakumar30092003@gmail.com

A Venkata Satya
Computer Science - Artificial Intelligence Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
satyavenkata46@gmail.com

KV Vamshidhar Reddy
Computer Science - Artificial Intelligence Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
kotavenkatavamshidharreddy7@gmail.com

Dr Jyothish Lal G
Professor Computer Science - Artificial Intelligence Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
g_jyothishlal@cb.amrita.edu

***Abstract*—In this project, we have developed a deep learning based approach for identifying and classifying 10 ragas from vocals in Carnatic Music using a hybrid CNN-BiLSTM model enhanced with an attention mechanism. The pipeline begins with data preprocessing, which includes filtering, source separation, downsampling, silence removal, data augmentation, Segmentation (10s with 50 percent overlap), and dataset balancing across ragas. The data preprocessing is followed by feature extraction, where three important pitch-based features—pitch contour, pitch velocity, and pitch acceleration — are extracted. The extracted features are then stacked in the form of a 2D matrix, where each row represents a time stamp and each column corresponds to a feature. This matrix is the input to a CNN-LSTM-attention model. The attention mechanism enables the model to focus on temporally significant parts of the audio. The model achieved a notable classification accuracy of 98.76 . The dataset used in this study was manually extracted from live concerts and recorded performances from YouTube and other online platforms. It consists of 10 Raagas, with multiple singers.**

***Index Terms*—Raaga, Filtering, Source separation, Downsampling, silence removal, Data augmentation, Segmentation, Dataset Balancing.**

I. Introduction

Music plays a significant role in the lives of human beings. For centuries, man has tried to shatter the barriers of his mind through the beauty of art, finding expression, solace, and inspiration, and music has always played a significant role in it, adding colors to a barren heart. Around the world, music exists in diverse forms. Few notable ones include the Arabic and Turkish classical, *Folj, Gnawa, Gqom, Erhu, Pinpeat,* and so on. In India, the two major systems of classical music are Hindustani, which is prominent in the northern part of India, and Carnatic music, which is primarily practiced in the southern, Dravidian-speaking region. Indian classical music is a deeply expressive and complex art form with centuries of cultural evolution. Both Carnatic and Hindustani music are characterized by Raagas. A Raaga is a set of rules that define the melodic frameworks for improvisation and composition. [1] It uniquely defines a set of musical notes and their allowed arrangements to form melodies to evoke certain emotions. Within Indian classical musical systems, a Raaga has the power to create very specific emotions in one's mind. [2] A range of emotions, such as joy, sadness, happiness, romance, yearning, devotion, and more, can be expressed through Raagas. Some Raagas are seasonal; they enhance the listener's mood through association with a particular season, such as spring or monsoon. [2]

Today, the beauty of traditional music is shadowed by the pop world. Connecting Carnatic music with technology not only helps in uplifting the forgone colors of Carnatic music but also creates a certain sort of curiosity in both youngsters' and adults' minds. Many Music Information Retrieval systems present today mainly focus on semi-classical and film music genres. There is a necessity to automate the raaga identification procedure as it becomes an enormous help to both professionals and lovers of Carnatic music. Each individual

would have a certain sort of affinity towards a particular Raaga. Thus a music recommendation system based on automated raaga recognition helps in suggesting raaga-specific songs. Carnatic music has seven swaras or *Saptaswara* which include *Sa* (*Shadja*), *Re*(*Rishabha*), *Ga*(*Gandhara*), *Ma*(*Madhyama*), *Pa*(*Panchama*), *Dha*(*Dhaivata*), and *Ni*(*Nishada*). *Sa* is the tonic or the base pitch. The base pitch Sa differs between individuals. The swara Sa is known as "*adhara shadja*". Except for *shadja sa* and *Panchama Pa*, the seven basic notes can be further divided into 16 notes (the Melakarta system used 16 named note positions or *Swarastanas*, even if some are sonically identical, i.e., same pitch.) The combination of these notes forms the Raagas in the Carnatic music system [3] The *Melakarta* and *Janya* Raga scheme, originally developed by *Venkatamakhi*, categorizes the raagas into two categories: *Melakarta* Ragas or the parental raagas and the *Janya* raagas or the child raagas that are based on *Melakarta* Raaga. [4] There are a total of 72 *Melakarta* Raaga, each of which is *sampoorna* (Uses all the seven notes (swaras) in both *Arohana* (Ascending) and *Avarohana* (descending) scales) and *Non-Vakra* (The notes are used in straight linear sequence). [4] While the *Melakarta* Raagas follow a strict set of rules, *Janya* Raagas are more flexible and diverse. They are the Raagas that evolved naturally through years of musical practice and tradition, and not through theoretical design. [4] The *Janya* Raagas can be further classified based on Swaras used as *Upanga* Raga (Uses only *Swaras* from parent *Melakarta* raga. E.g., *Hamsadhwani*.), *Bhashanga* raga (Borrows one or more *Swaras* (*anya swaras*) from a different *Melakarta* Raga .E.g., *Kharaharapriya*) and based on Structure as *Vakra* Raga (Follows a non-linear pattern in the *arohana* and *avarohana* .E.g., *Kambhoji*) and Non- *Vakra* Raga (Notes are straight sequence, similar to *Melakarta* Raga .E.g., *Mohanam*). [4] In this work, we have come up with a methodology for classifying a combination of 10 *Melakarta* and *Janya* Raaga.

II. Dataset

For this research work, we have created our own dataset "*SwararaagaSudha*". It consists of 10 Raagas (both *Melakarta* and *Janya*), which include *Ananda Bhairavi, Darbari Kanada, Hamsadhwani, Kalyani, Kharaharapriya, Mayamalavagowla, Mohanam, Neelambari, Shankarabharanam,* and *Thodi*. Over one hour of audio files from different female and male artists were downloaded from YouTube and other online platforms for each raagas. The audio files were further augmented and a total of 2 hour 28 minutes of audio were created for each raaga (after balancing). In total, *Swararaaga Sudha* contains a total of 24.67 hours of audio files after augmentation.

Raaga	Total Number of Tracks	Hours	Female Arti
AnandaBhairavi	10	57min 19sec	6
DarbariKannada	10	58min 29sec	4
Hamsadhwani	9	1 hour 1 min 5 sec	5
Kalyani	9	1 hour 1 min	2
Kharaharapriya	7	1 hour 3 min 5 sec	3
Maayamalawagowla	7	1 hour 5 min	3
Mohanam	7	1 hour 3 min 50 sec	6
Neelambari	8	1 hour 28 sec	3
Shankarabharanam	6	1 hour 5 min 3 sec	4
Thodi	6	1 hour 4 min 54 sec	2

TABLE I
DATASET DESCRIPTION

III. Literature Review

Recent advancements in machine learning and deep learning have significantly improved the automatic recognition of ragas in Indian classical music. Various studies explored different architectures, datasets, and performance metrics to address the complexity and subtleties of this task. Siji John et al. implemented a CNN-based model using TensorFlow for classifying ragas from the CompMusic dataset. The model employed a sequential architecture involving convolutional, pooling, and fully connected layers, achieving an accuracy of 94% for five typical ragas under consideration [8]. Another notable contribution by Devansh P Shah et al. proposed a hybrid CNN-LSTM architecture, which utilizes spectrogram images to extract spatial features through CNNs and captures temporal dependencies using LSTM layers. This model reported an accuracy of 98.98% on the CompMusic dataset, demonstrating the effectiveness of temporal modeling in raga recognition tasks [9]. Traditional machine learning methods have also been explored. Gopala Krishna Koduri et al. adopted a k-Nearest Neighbors (kNN) framework using a custom dataset of 170 tunes across 10 ragas. This method achieved a classification accuracy of 76.5%, highlighting the limitations of non-deep approaches in capturing intricate raga characteristics [10]. Multimodal learning has been explored by Stella et al., who proposed a deep neural network that integrates both audio signals and metadata for Hindustani raga classification. Their system attained 98.22% accuracy using the CompMusic Art Indian music dataset, indicating the value of multi-source information for complex tasks like raga identification [11]. Bhagyalakshmi R et al. utilized a Hidden Markov Model (HMM) trained using the Baum-Welch algorithm for raga classification using the GTraagDB dataset. This probabilistic approach yielded 90% accuracy, showing that temporal sequence modeling without deep architectures can still produce competitive results [12]. Additionally, Parampreet Singh et al. employed a CNN-LSTM model for multi-class classification of 12 ragas, achieving an F1-score of 0.89 using their own curated dataset. Their work emphasizes the importance of explainability in deep learning models for Indian art music analysis [13]. Collectively, these studies underline a growing shift towards deep learning methods, particularly hybrid CNN-LSTM frameworks, due to their superior ability to handle the temporal and spectral intricacies inherent in Indian classical music. The performance improvements across models and

datasets validate the ongoing transition from traditional methods to more sophisticated neural approaches in the domain of raga recognition.

IV. Methodology

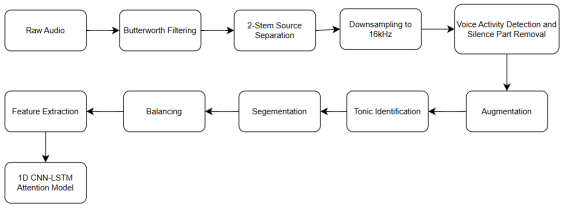

Fig. 1. Block diagram of the proposed CNN–BiLSTM–Attention architecture for rāga classification.

In this section, we will be discussing about the pipeline and the procedures that we have implemented in our work.

A. Data Preprocessing

1) Butterworth Filtering: We started by applying a bandpass filter to clean up the audio signals. For this purpose, we used a 5th-order Butterworth bandpass filter with a cutoff frequency range of 50 Hz and 5000 Hz. The vocal fundamental frequencies (swara range) typically lie between 80 Hz to 800 Hz. But important harmonics content and gamakas can extend up to 4000 to 5000 Hz. The frequencies below 80 Hz and those above 5 kHz are typically considered noise, such as electrical hum, microphone rumble, etc. Using a Butterworth filter helps in analyzing the musically relevant frequency band and improves the quality of feature extraction.

$$|H(j\omega)|^2 = \frac{1}{1 + (\omega/\omega_c)^{2n}}, \tag{1}$$

where ω is the angular frequency in radians per second, ω_c is the cutoff frequency, and n is the order of the filter.

2) Source Separation: The Butterworth filtered audio signals are then made to undergo source separation, where the vocals alone are extracted from the filtered audio. We have done this using HTDemucs (Hybrid Architecture Transformer Blocks Deep Extractor For Music Sources), introduced by Facebook AI Research. HTDemucs is the latest version in the Demucs family used for source separation, i.e., separating different instrumental and vocal components. In our work, we have done a two-stem source separation where the vocals are extracted from the audio files and saved in a separate directory. The Demucs architecture consists of Convolutional layers which extract the local time-frequency features, Bi-directional LSTM's which take into consideration temporal dependencies and the transformer blocks for capturing long term temporal context and improving the source separation quality. [5]

3) Downsampling: For the purpose of removing silence from audio files using the WebRTC Voice Activity Detection algorithm, the audio files should have a sampling rate of 16 kHz for efficient silence detection. But our audio files, after source separation, have a sampling rate of 44.1 kHz. So we downsampled our audio into 16 kHz using the librosa.resample function.

$$\mathbf{x}_{\text{res}} = \text{Resample}(\mathbf{x}, f_s^{\text{in}}, f_s^{\text{out}}), \tag{2}$$

where $\mathbf{x}$ is the original audio signal, $f_s^{\text{in}} = 44{,}100$ Hz is the input sampling rate, and $f_s^{\text{out}} = 16{,}000$ Hz is the target sampling rate.

4) Silence removal using Voice Activity Detection: Even though HTDemucs isolates only the vocal stem, the duration of the audio files will be preserved. This leads to the vocal track files having silent parts. Silent parts can also be caused due to the singer pausing for breaths or because of low vocal energy. These sections in the audio files are useless and can introduce noise during the training process, thus misleading the model. So it becomes a necessity to remove these silent parts. In our work, we have used WebRTC VAD, developed by Google. It classifies short audio frames as either voiced or unvoiced based on energy, spectral content, and temporal patterns. WebRTC was initially designed for speech detection, but it worked well on our Carnatic dataset. The audio files were divided into 30-ms frames, and then individually classified as voiced or unvoiced. We have used medium aggressiveness (level 2), which offers a balanced trade-off between too much trimming and preserving soft vocal phrases. The frames identified as voiced regions are then concatenated, thus returning audio files that do not contain the silent part. [6]

5) Data Augmentation: Data diversity is crucial for deep learning based tasks. Given the limited size of our dataset, we have applied data augmentation techniques to expand the dataset and introduce variations in pitch, timing, and background noise while preserving the raaga's melodic identity. The techniques of pitch shifting, time stretching, and additive noise were used for this purpose, as shown in previous work using augmentation for raga classification [7] We applied three augmentation techniques to each preprocessed clip. i.e., a pitch shift of +2 semitones, time stretching factor of 1.1, and addition of white noise with noise factor 0.005 to each preprocessed vocal clip. The augmented files are then saved alongside the original clip, increasing the dataset size fourfold, significantly improving diversity.

$$\mathbf{x}_{\text{pitch}}(t) = \text{PitchShift}(\mathbf{x}(t), \Delta p), \quad \Delta p = +2 \text{ semitones}, \tag{3}$$

$$\mathbf{x}_{\text{stretch}}(t) = \text{TimeStretch}(\mathbf{x}(t), \alpha), \quad \alpha = 1.1, \tag{4}$$

$$\mathbf{x}_{\text{noise}}(t) = \mathbf{x}(t) + \epsilon(t), \quad \epsilon(t) \sim \mathcal{N}(0, \sigma^2), \quad \sigma = 0.005, \tag{5}$$

Each augmented signal is saved with 'pitch', 'stretch', and 'noise' in its label, so as to separate them during model training part, along with the original, resulting in a fourfold increase in the dataset size:

$$\mathcal{D}_{\text{aug}} = \{\mathbf{x}, \mathbf{x}_{\text{pitch}}, \mathbf{x}_{\text{stretch}}, \mathbf{x}_{\text{noise}}\}. \tag{6}$$

6) Tonic Estimation and Audio Segmentation Using TorchCREPE: As mentioned earlier, the *Shadja* or *Sa* is the tonic or the foundational pitch from which all other Swaras are derived. This can vary over singers. For instance female singers usually will have a higher tonic compared to male singers. Thus it is essential to first determine the tonic frequency before segmentation to normalize the pitch scale for each audio clip. For this purpose, we used TorchCREPE, a GPU-accelerated pitch estimator, to determine the tonic of each audio file. The audio is converted to mono and loaded at 16 kHz. It is further reshaped and passed through the prediction model. We used a hop size of 10 ms for finer temporal resolution. Both the pitch and periodicity or confidence are estimated. Pitch values with a periodicity greater than 0.5 are considered reliable. The final step involves estimating the tonic by creating a histogram of confident pitch values and selecting the mode or the most frequent pitch value. The audio files are then segmented into 10 seconds with a 50% overlap between consecutive segments. The segmented files are then saved with their tonic frequency (extracted before) in their label. For example, `Mohanam_tonic=131.2_pitch_seg0.wav`, `Mohanam_tonic=131.2_trimmed_seg0.wav` ,`Mohanam_tonic=131.2_noise_seg0.wav` and so on. This is done so that during feature extraction, the tonic frequency remains the same for all segments derived from the same audio. The tonic is computed as the mode of the confident pitch values:

$$f_{\text{tonic}} = \text{mode}\left(f_p^{(i)} \mid c_p^{(i)} > 0.5\right), \tag{7}$$

where $f_p^{(i)}$ and $c_p^{(i)}$ are the i-th pitch and confidence values, respectively. Only frames with confidence > 0.5 are considered reliable.

Let f_s be the sampling rate and $L = T \cdot f_s$ be the segment length in samples. Then, the k-th segment $\mathbf{x}_k$ is defined as:

$$\mathbf{x}_k = \mathbf{x}[kL/2 : kL/2 + L], \quad k = 0, 1, 2, \ldots \tag{8}$$

7) Balancing: In order to have an unbiased training, it is necessary to balance the data across Raagas. For this purpose, we have come up with a duration-based dataset balancing which would iterate through the entire raaga folder, finding the one with the least time. Then, for raagas with data more than the minimum, we selected only enough segments, in sorted order, to match the target duration.

The Raaga with the least amount of time is Kharaharapriya with 23440 sec.

B. Feature Extraction

In our work, we have focused mainly on three pitch-based features, i.e., pitch contour, pitch velocity, and pitch acceleration. CREPE model, which is well-suited for Indian classical music due to its ability to detect continuous pitch glides, is used here. We have used Viterbi decoding to ensure smoother pitch contours and a 10-ms step size for high temporal resolution.

Raaga	Duration (sec)
Anandabhairavi	23540
Darbari Kanada	24640
Hamsadhwani	23450
Kalyani	24340
Kharaharapriya	23440
Mayamalawagowla	26880
Mohanam	24860
Neelambari	23700
Shankarabharanam	30990
Thodi	26450

TABLE II
TOTAL DURATION OF RAAGAS AFTER AUGMENTATION

1) Pitch Contour: Pitch contour is the continuous representation of how fundamental frequency changes over time during musical performance. [8] After extracting the pitch contours, it is then normalised into cent value in order to ensure raaga for person-independent raaga classification.

$$p(t) = 1200 \cdot \log_2\left(\frac{f_0(t)}{f_{\text{tonic}}}\right), \tag{9}$$

where $p(t)$ is the pitch in cents, relative to the tonic.

2) Pitch Velocity: Measures the speed at which pitch changes over time. In Carnatic music, how we sing a note is as important as the note itself. Two Raagas may use the same set of Swaras, but the speed and style of transitions between them .i.e., the pitch velocity are what that make the raaga unique. E.g. The graceful oscillation in *kamboji* vs the quick flicks in *Shankarabharanam*.

$$v(t) = \frac{dp(t)}{dt}, \tag{10}$$

where $v(t)$ is the pitch velocity in cents per second.

3) Pitch Acceleration: Pitch acceleration measures how quickly the singer's voice speeds up or slows down, moving from one note to another. It helps in understanding the energy and expression in which notes are sung, especially in *Gamakas*. For Example, in *Neelambari Raaga*, a phrase might begin with a faster glide, then slow and settle down. It is measured by taking the gradient of pitch velocity.

$$a(t) = \frac{dv(t)}{dt} = \frac{d^2p(t)}{dt^2}, \tag{11}$$

where $a(t)$ is the pitch acceleration, representing the second-order temporal dynamics of pitch. The extracted features are then stacked into a 2D matrix where the rows represents time steps (at 10ms resolution) and coloumns represents features. The dimension of the matrix is (1001, 3).

C. Model

1) Dataset preparation and spliting: Since we have augmented our dataset it is important to ensure that testing and validation set doesn't contain augmented data. For achieving this, as mentioned earlier, we have saved our augmented features with labels containing words 'pitch', 'stretch', and 'noise' for pitch shifted, time stretched and noise added features. So now for training validating and testing the model, we have created three different folders, for train test and val,

and made sure only non augmented data gets saved in test and validation after a 70% train, 15% val and 15% test split.

Train	21,264
Test	1,130
Val	904

TABLE III
TOTAL NUMBER OF FILES FOR TRAIN,TEST AND VAL

After spiltting the data the labels (.ie., the name of raaga) are then converted into one hot encoded vectors.

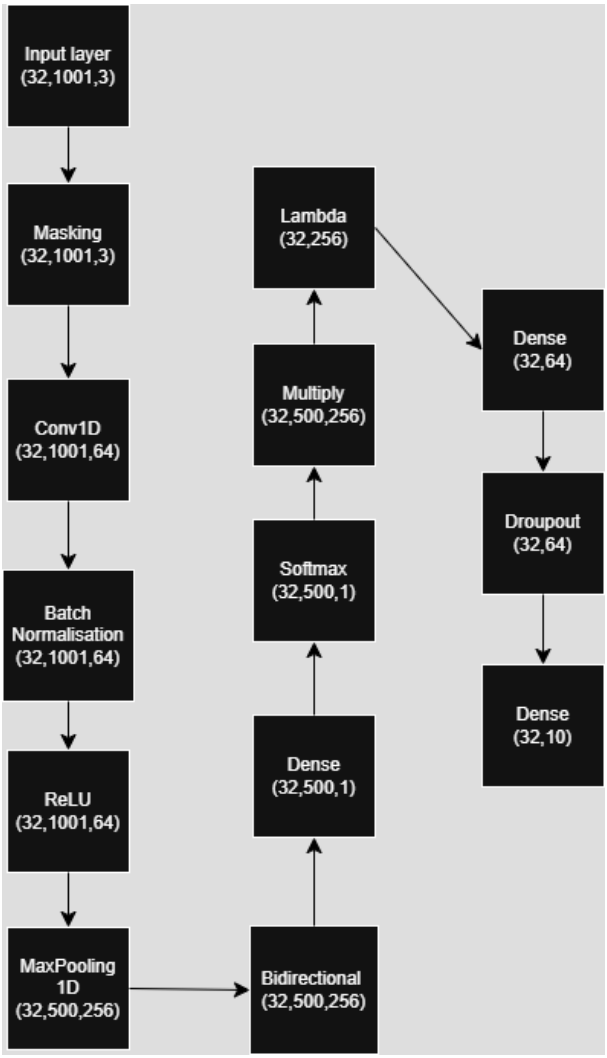

Fig. 2. Block diagram of the proposed CNN–BiLSTM–Attention architecture for rāga classification.

2) Model Description: Our propsed model is a hybrid deep learning model to capture both spectral features and temporal dependies of our features. It contains 1D CNN whose output is given as input to a Bidirectional LSTM layer 128 units in each direction. This helps our model to capture both forward and backword temporal dependencies.

To focus on the most informative time steps, we have employed an attention mechanism. The attention mechanism calculates a context vector by assigning weights to each of the time step of the BiLSTM output. The attention scores are calculated by using a non-linear transformation and then a softmax normalization.

$$\mathbf{A} = \text{softmax}\left(\tanh(\mathbf{W}_a \cdot \mathbf{H})\right), \tag{12}$$

$$\mathbf{C} = \sum_{t=1}^{T} A_t \cdot \mathbf{H}_t, \tag{13}$$

The final layer of our model consists of a fully connected Dense Layer with 64 units followed by a ReLU activation layer. We have used a dropout value of 0.3 to prevent overfitting. The final dense layer with the softmax activation gives us the final output class probabilities for our 10 Raagas.

3) Training: For training purpose we have compiled our model using Categorical cross-entropy loss functionand Adam optimizer with a learning rate of 0.001. The training of the model is done for 100 epochs with a batch size 32 and callbacks EarlyStopping with a patience of 20 epochs (which monitors validation loss) and ModelCheckpoint to save the model with the best validation accuracy. The best validation accuracy was found at the 100th epoch and equals to 99.558 with validation loss 0.0199. The training accuracy is 97.97 and training loss is 0.0608. Thus the model turned out to be an efficient model with no overfitting.

V. RESULTS

Our model showed promising results with a testing accuracy of 98.76 and testing loss of 0.0451,when tested on completely unseen data. The precision, recall, f1-score, and support values are given below.

Testing Accuracy	98.76
Testing Loss	0.0451

TABLE IV
TESTING ACCURACY AND LOSS

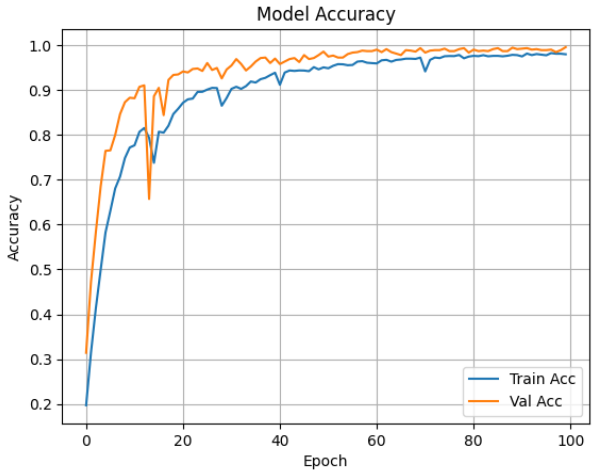

Fig. 3. Training and validation accuracy across epochs

Raaga	Precision	Recall	f1-Score	Support
Ananda Bhairavi	0.98	1.00	0.99	108
Darbari Kanada	0.98	0.97	0.98	116
Hamsadhwani	0.98	1.00	0.99	120
Kalyani	0.99	0.99	0.99	112
Kharaharapriya	0.98	0.99	0.99	120
Mayamalavagowla	0.98	1.00	0.99	114
Mohanam	1.00	0.98	0.99	113
Neelambari	0.98	0.98	0.98	122
Shankarabharanam	0.99	0.98	0.98	95
Thodi	1.00	0.97	0.99	110

TABLE V
CLASSIFICATION REPORT

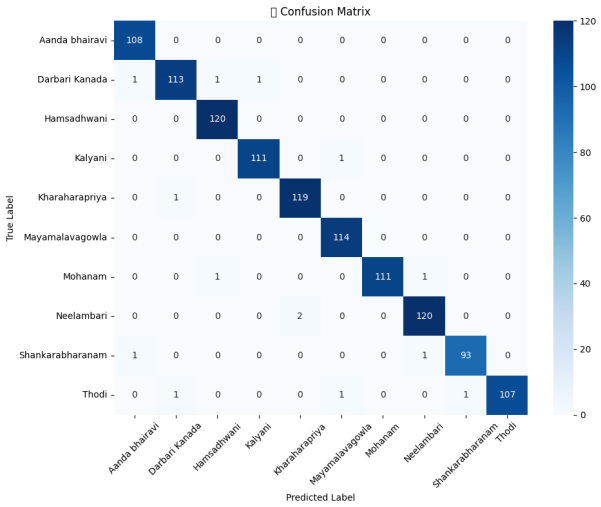

Fig. 4. Confusion Matrix of Ten Raagas

VI. CONLUSION

In our project, we tried to understand how we can use deep learning models to recognize and classify Raaga Carnatic from vocal recordings. We have included different data preprocessing and cleaning steps, which gave us good results. Our model is a hybrid model made up of 1D CNN for local feature learning, BiLSTMs for capturing temporal patterns, and an attention mechanism to focus on the most relevant parts of the audio. From the results our model proved tp be effective in Raaga classification. We can further extend this work for covering all Raagas.Our work shows that technology, when guided by musical understanding, can be a powerful tool to preserve and promote rich traditions like Carnatic music.

ACKNOWLEDGMENT

We would like to express our heartily thanks to Dr Jyothish Lal Sir for guiding us and making us efficient enough to achieve this. We thank our college, Amrita Vishwa Vidyapeetham, for giving us the opportunity for quality education and knowledge. We hope we will be able to come up with more efficient ideas and projects under Dr Jyothish Lal Sir's Guidance. We also like to pay our tribute to Shree Swathi Thirunal Rama Varma, whose compositions inspired us a lot in building our project.

REFERENCES

[1] https://www.britannica.com/art/Hindustani-classical-music
[2] https://www.indianclassicalmusic.com/what-is-raag
[3] P. Govindarajan, S. M. Mothi and A. Tesfahun, "An AI Model for Recognization of Raagas in Indian Classical, Carnatic Music – A Review Article," 2023 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Mysuru, India, 2023, pp. 110-119, doi: 10.1109/CCEM60455.2023.00025.
[4] The journal of the music academy devoted to the advancement of the science and art of music vol lxii (1991)how many janya ragas are there ? P. Sriram and Valavanur N. Jambunathan.
[5] Défossez, A., Usunier, N., Bottou, L., & Bach, F. (2021). Hybrid Transformer Demucs: Removing vocals from music with transformers. Meta AI Research. Available: https://github.com/facebookresearch/demucs
[6] Google, "WebRTC Voice Activity Detector (VAD)," WebRTC.org, 2011. [Online]. Available: https://webrtc.googlesource.com/src/+/refs/heads/main/common_audio/vad
[7] P. Mututhan, K. T. Sreekumar, K. I. Ramachandran, and C. Kumar, "Data Augmentation for Improving the Performance of Raga (Music Genre) Classification Systems," in Proc. IEEE Int. Conf. on Emerging Smart Computing and Communication Technologies (ICESC), Aug. 2024, pp. 1407–1412. doi: 10.1109/ICESC60852.2024.10690091.
[8] S. John, M. S. Sinith, S. R. S and L. P. P, "Classification of Indian Classical Carnatic Music Based on Raga Using Deep Learning," 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Thiruvananthapuram, India, 2020, pp. 110-113, doi: 10.1109/RAICS51191.2020.9332482.
[9] D. Shah, N. M. Jagtap, P. T. Talekar, and K. Gawande, "Raga Recognition in Indian Classical Music Using Deep Learning," in *Artificial Intelligence in Music, Sound, Art and Design*, Lecture Notes in Computer Science, vol. 12691, pp. 248–263, Springer, 2021. doi: 10.1007/978-3-030-72914-1_17
[10] K. G. Koduri and S. Gulati, "A Survey of Raaga Recognition Techniques and Improvements to the State-of-the-Art," Research Report, Jan. 2011. [Online]. Available: https://www.researchgate.net/publication/264885437
[11] Stella *et al.*, "Multimodal Deep Learning Architecture for Hindustani Raga Classification," *ProQuest*, [Online]. Available: https://www.proquest.com/docview/2841538396
[12] Bhagyalakshmi R et al., "Machine Learning Based Indian Raga Classification and Detection," in Proc. IEEE, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10673605
[13] Parampreet Singh et al., "Explainable Deep Learning Analysis for Raga Identification in Indian Art Music," arXiv preprint, arXiv:2406.02443, 2024. [Online]. Available: https://arxiv.org/abs/2406.02443