Audio Separation and Classification of Indian Classical Instruments from Monophonic and Polyphonic Audio Clips

Dr R. Srinivasan
Department of Information Technology
Sri Sivasubramaniya Nadar College of Engineering
Kalavakkam, Chennai
srinivasanr@ssn.edu.in

Abhishek Rajagopal
Department of Information Technology
Sri Sivasubramaniya Nadar College of Engineering
Kalavakkam, Chennai
abhishek2210225@ssn.edu.in

Hayden C
Department of Information Technology
Sri Sivasubramaniya Nadar College of Engineering
Kalavakkam, Chennai
hayden2210218@ssn.edu.in

***Abstract*—Indian classical music often features multiple instruments playing together, as well as complex melodic patterns and intricate instrumental interplay. Due to overlapping frequency spectra, different tonal characteristics, and the lack of labeled data for traditional Indian instruments, computational analysis of this music faces challenges. Through two related goals: single-instrument classification and multi-source audio segregation, this study investigates algorithmic approaches to research Indian classical instruments.**

Classifying solo instrument excerpts from the IMID collection into groups like flute, veena, sitar, sarod, tabla, violin, guitar, piano, bass, and percussion is the first phase of this research. A comparison between traditional and feature-learning approaches for instrument identification is made easier by testing a variety of machine learning and neural network architectures, including Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and a pretrained PANN model.

Following this, the research is focused on analyzing polyphonic audio clips present in the Sanidha data set. Polyphonic compositions present in the dataset were split into three distinct layers: Vocal tracks, Mridangam and an instrumental segment consisting of a Violin and Tampura. Advanced separation frameworks such as an SCNet, BS-Roformer, and HTDemucs were trained and compared to assess their efficacy in extracting individual components from these musical compositions.

Hence, by addressing both the classification of isolated audio clips into distinct instruments and mixed audio decomposition, the study offers a thorough exploration of various methods to separate and classify widely used Indian musical instruments present in complex compositions. The outcomes support the advancement of analytical tools for music research, specifically for underrepresented Indian Instruments and preservation and archiving of Indian compositions.

***Keywords*—Indian Classical Music, Music Information Retrieval, Source Separation, Instrument Classification, Monophonic Classification, Polyphonic Audio Analysis, IMID Dataset, Sanidha Dataset, Support Vector Machines (SVM), Convolutional Neural Networks (CNN), Pretrained Audio Neural Networks (PANN), SCNet, BS-Roformer, HTDemucs.**

I. Introduction

Indian classical music majorly encapsulates two styles, namely Carnatic and Hindustani. These traditions have ornate melodic structures called ragas, and rhythmic cycles called talas, with compositions that often have many instruments performing simultaneously. Such compositions have a polyphonic nature, wherein instruments occupy overlapping frequency spectra and display various tonal attributes, and present significant challenges to algorithmic analysis and automated identification. Accurate separation and isolation of instruments in these recordings are essential to academic research, digital research and preservation, content-driven retrieval, and intelligent audio processing.

The identification of instruments and separation of audio using automation systems has been studied in depth in the context of Western music. Techniques no longer rely on handcrafted feature extractors and classifiers, but advanced neural networks, with convolutional networks (CNNs), recurring structures, and pre-trained audio embeddings, such as PANNs. These approaches have achieved great levels of success in isolating and classifying instruments in thick musical settings. The presence of large labeled datasets, including MusDB, MedleyDB and MusicNet, has played a key role in enabling these developments by allowing holistic training programs and analysis of large scale sound.

On the other hand, the number of computational studies of Indian classical music is relatively small. One of the major obstacles lies in a lack of carefully annotated data on indigenous instruments. Besides, Indian classical recordings often have complex polyphonic effects, impromptu improvisational structures, and broad sonic characteristics that make them difficult to automatically partition. Although some of the previous studies have analyzed individual instrument passages or even monophonic fragments, there is still rather limited work on solutions that consider both the recognition and separation of multi-layered scenarios.

In order to address these challenges, the study will look at computational methods on scrutiny of Indian classical instruments on the basis of two related tasks: monophonic instrument classifications and polyphonic sound separation. Initially, single-instrument samples from the IMID collection are employed to develop and assess classification systems

targeting instruments like flute, veena, sitar, sarod, tabla, violin, guitar, piano, bass, and percussion. Diverse categorization techniques, including Support Vector Machines (SVM), Convolutional Neural Networks (CNN), as well as Pretrained Audio Neural networks (PANN) are investigated to evaluate their effectiveness in the classification of instrumental timbre on single recordings.

The next stage is focused on polyphonic sound analysis using the Sanidha dataset. In this case, modern music separation models, including SCNet, BS-Roformer, and HTDemucs, are evaluated based on their ability to separate multilayered recordings into three main elements: mridangam, vocal overheads, and an auxiliary instrumental layer, which consists of violin and tanpura. These frames are compared in terms of separation and stability in case of dealing with complex Indian classical amalgamations of audio.

Exploring not only monophonic classification but also polyphonic source segregation provides this study with a broader analysis of machine learning in understanding Indian classical music in both single and mixed listening conditions. The offered methodology supports the process of narrowing the uses of analytical tools to process, catalog, and conserve the aural Indian heritage in the digital form and addresses the challenges connected with non-Western musical systems.

II. Literature Survey

A. Music Instrument Recognition

IRMAS Dataset: One of the most widely used criteria of assessment of systems recognizing musical instruments is the IRMAS (Instrument Recognition in Musical Audio Signals) collection. It is comprised of 6,705 training and 2,874 evaluation audio snippets, 3 seconds long and 11 different types of Western instruments. This dataset is often trained using acoustic feature representations (e.g. Mel-spectrograms and Mel-Frequency Cepstral Coefficients (MFCCs)) together with computational models (e.g. Support Vector Machines (SVMs) and Convolutional Neural Networks (CNNs)). These strategies have recorded good performance of single-instrument and multi-instrument recognition issues. This catalog has played a significant role in defining numerous methods of reference when it comes to the classification of the instruments[1].

OpenMIC-2018 Dataset: The OpenMIC-2018 is a massive repository that is specialized in the identification of different instruments in music recordings. It consists of about 20,000 audio segments, 10 seconds long and poorly labelled 20 types of instruments. In comparison to data sets that have an accurate annotation, OpenMIC gives segment level tags that show the existence of an instrument, but not the timing information. This is what makes it especially handy to learn weakly supervised learning methods and attention-based models. Neural network models based on CNNs, specifically spectral representation learning models, have been shown to achieve higher accuracy on this dataset, specifically in terms of detecting simultaneous instruments in polyphonic audio.[4].

MedleyDB Dataset: MedleyDB is a multi-track archive of about 120 professional recordings, with more extensive annotations such as melody notes and instrument on/off. All of the tracks contain isolated stems, which are especially handy in activities like breaking down sound sources, identifying instruments, and transcription of multiple instruments. Due to its high-quality recording and carefully edited annotations, MedleyDB has been used extensively in training and evaluating deep learning models on music information retrieval tasks, especially in the Western music domain[5].

IMID Dataset: To address the lack of publicly available datasets in terms of Indian classical instruments, Indian Musical Instrument Dataset (IMID) was created. This data set contains the recording of different traditional instruments, which are used regularly, including flute, veena, sitar, sarod, tabla, violin, guitar, piano, bass, and drums. Most of the recordings are monophonic, with one instrument in each clip. IMID is a valuable instrument in creating instrument classification systems that would be specifically applied to Indian musical situations. It helps in the analysis of machine learning models detecting the acoustic peculiarities of Indian instruments, which in most cases differ significantly with western ones.[8].

B. Music Source Separation

HTDemucs: HTDemucs or Hybrid Transformer Demucs is a hybrid of convolutional neural networks with transformer-based attention. The model functions across both time and frequency domains, enabling it to grasp long-term dependencies while preserving detailed spectral accuracy. HTDemucs has already demonstrated remarkable performance on conventional benchmarks, such as MUSDB18, making it one of the best solutions to extracting single components, such as vocals, drums, bass, and other instruments of mixed audio[6].

BS-Roformer: The BS-Roformer model uses a transformer-based method of separating musical sources through time-frequency analysis. It integrates band-splitting techniques that segment spectrograms into distinct frequency bands, facilitating specialized learning in varied spectral ranges. Based on rotary positional embeddings and attention mechanisms, this model captures temporal patterns in audio signals. Empirical results reveal strong performance in isolating complex musical mixtures, specifically excelling in modeling lengthy dependencies within audio datasets[12].

SCNet: The SCNet model is a neural network architecture, which is specifically designed to address audio separation problems. It has structured convolutional layers and fine feature extraction, to enhance the accuracy of spectral representation. These models have the expertise of separating overlapping musical components, which can be useful in situations where two or more instruments are in similar frequency bands[13].

MUSDB Dataset: MUSDB is a common source separation benchmark that is used in musical source separation studies. It includes 150 professional recordings and single stem elements, and each piece of composition has four categories, including vocals, percussion, bass and the rest of the instrumentation. It provides mixed and isolated tracks, which facilitate monitored

model training and evaluation. MUSDB is considered as a standard benchmark and assesses the effectiveness of modern separation systems, such as Demucs, based on performance metrics, such as Signal-to-Distortion Ratio (SDR). The dataset, which is mostly made of western music, has been utilized in influence of the development and also benchmarking of the state-of-the-art source separation models which can then be reused to other musical situations[14].

Sanidha Dataset: The Sanidha dataset consists of studio-quality multitrack recordings of Carnatic music performances. Each recording provides separate stems for individual musical components, which significantly reduces the issue of microphone bleed between instruments. The dataset includes stereo stems such as vocals, mridangam, violin, tanpura, and some tracks also have ghatam. Because of its clean multitrack structure and representation of traditional Indian classical music ensembles, the Sanidha dataset provides an important resource for training and evaluating source separation models specifically for Indian classical music applications[9].

III. Proposed Methodology

This study investigates two complementary tasks in the computational analysis of Indian classical music: (1) monophonic instrument classification and (2) polyphonic source separation. These tasks are explored independently in order to analyze instrument characteristics in isolated recordings as well as complex polyphonic mixtures.

A. Monophonic Instrument Classification

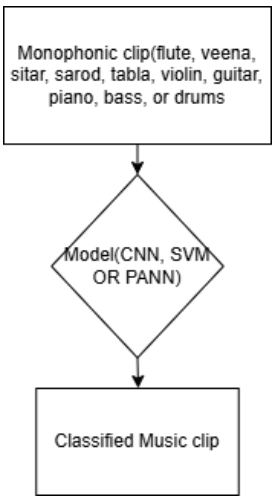

Fig. 1. Classification pipeline.

Objective: Identify the instrument present in monophonic audio clips using machine learning and deep learning models.

IMID Dataset: The Indian Musical Instrument Dataset (IMID) contains recordings of individual instruments including flute, veena, sitar, sarod, tabla, violin, guitar, piano, bass, and drums. Each audio clip contains a single instrument, allowing models to learn distinctive timbral characteristics without interference from other sources.

Input: Monophonic audio clips corresponding to individual instruments including flute, veena, sitar, sarod, tabla, violin, guitar, piano, bass, and drums.

Output: Predicted instrument labels for each input audio clip.

Model Choices:

Support Vector Machine (SVM): A classical machine learning model that performs classification using handcrafted spectral features or learned audio embeddings.

Convolutional Neural Network (CNN): A deep learning architecture capable of learning hierarchical spectral patterns directly from time-frequency representations such as log-Mel spectrograms.

Pretrained Audio Neural Network (PANN): A model trained on large-scale audio datasets that produces robust audio embeddings capturing both spectral and temporal information. These embeddings are used to improve classification performance, especially when working with relatively small datasets.

B. Polyphonic Source Separation

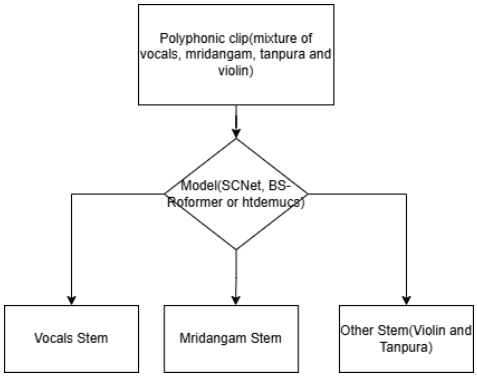

Fig. 2. Classification pipeline.

Objective: Separate polyphonic Carnatic music recordings into distinct audio stems corresponding to major musical components.

Sanidha Dataset: The Sanidha dataset consists of studio-quality multitrack recordings of Carnatic music performances. Each recording provides clean stems for different musical components, making it suitable for training and evaluating source separation models in Indian classical music contexts.

Input: Polyphonic Carnatic music recordings containing multiple instruments and vocals.

Output: Separated audio stems consisting of three components:

- Vocals
- Mridangam
- Instrumental stem containing violin and tanpura

Model Choices:

SCNet: A deep neural network architecture designed for audio source separation that focuses on learning structured time-frequency representations to effectively isolate overlapping sound sources.

BS-Roformer: A transformer-based model that utilizes band-splitting techniques and attention mechanisms to capture long-range temporal dependencies in audio signals, enabling improved separation of complex musical mixtures.

HTDemucs: A hybrid architecture combining convolutional layers with transformer-based attention mechanisms to perform high-quality music source separation in both time and frequency domains.

IV. Dataset Description

A. IMID Dataset

IMID is a curated collection of recordings specifically aimed at supporting automatic recognition of Indian classical instruments. The dataset contains 10,050 audio tracks, comprising about 33 hours of audio, occupying a storage space of 9.9 GB. There are 10 musical instruments represented: Flute, Veena, Sitar, Sarod, Tabla, Violin, Guitar, Piano, Bass, and Drums.

For the IMIDataset, the authors collected several Indian musical instruments' audio from various sources. This dataset was then preprocessed, eliminating singers, canceling noise, and trimming each audio clip to within 10 s, thus making the data more interpretable and easier to analyze. The dataset is fine-tuned to ensure that each instrument is suitably represented to eliminate biases. Preparation procedures guaranteed uniformity and high-quality data for later model computations. By canceling the voices and performing noise cancellation, the authors could isolate the sounds produced by the musical instruments, and they became much easier to identify. Finally, the files were renamed and rearranged into classes correlating to the monophonic clip being played.

B. Sanidha Dataset

The Sanidha dataset is a recently introduced dataset designed specifically for computational research in Carnatic music. It provides studio-quality multitrack recordings of Carnatic music performances, making it particularly suitable for tasks such as music source separation and audio analysis.

The dataset contains recordings of 5 Carnatic concerts each consisting of different songs, where individual instruments and vocals are captured through separate recording channels. These recordings were conducted in controlled studio environments where musicians performed in acoustically isolated spaces, significantly reducing the problem of audio bleed between instruments. This setup enables the creation of clean individual stems for different musical components. The ensemble recordings typically include vocals, violin, mridangam, tanpura and some songs included the ghatam. For the purposes of this research, we have excluded the ghatam stem as it is not included in all the songs.

The audio in the dataset is recorded in high-quality WAV format with a sampling rate of 44.1 kHz and 16-bit resolution. In addition to the audio recordings, the dataset also provides supplementary metadata and high-definition videos of the performances, enabling potential multimodal research involving both audio and visual information. Such multimodal data can support further studies in music performance analysis and musician interaction.

Because of its clean multitrack structure and minimal overlap between sources, the Sanidha dataset provides an important benchmark for training and evaluating deep learning models for music source separation in Indian classical music contexts. It enables researchers to create realistic polyphonic mixtures while still having access to the original isolated instrument stems for supervised learning and evaluation.

V. Experimental Results

A. Monophonic Instrument Classification

1) CNN-Based Classification: Dataset: The experiments were performed on the IMID dataset, which contains 10,050 one-minute audio tracks of 10 Indian musical instruments: Flute, Veena, Sitar, Sarod, Tabla, Violin, Guitar, Piano, Bass, and Drums. The dataset was divided into an 80-20% train test split.

Audio Preprocessing and Feature Extraction:

- Each audio clip was standardized by shortening it to 3 seconds and resampling it to 22,050 Hz.
- Log-Mel Spectrograms with 128 Mel bands and a hop length of 512 samples were extracted from each audio clip to give as an input to the CNN model.
- The spectrograms were normalized(zero-mean and unit variance) and were either padded or truncated to a fixed time dimension of 130 frames to provide a uniform input size for the CNN model.

Model Architecture:

A Convolutional Neural Network (CNN) was implemented with the following architecture:

- Three convolutional blocks with 32, 64, and 128 filters respectively each followed by batch normalization, ReLU activation, and 2x2 max-pooling.
- A fully connected layer with 256 neurons and dropout (0.3) for overfitting control.
- Output layer with ten neurons (one for each instrument) and softmax activation for multi-class classification. The flattening size of the CNN was calculated while training from the Mel spectrogram input dimensions.

The flattening size of the CNN was dynamically calculated based on the Mel spectrogram input dimensions.

Performance Metrics:

- *Validation Accuracy:* The model was able to get a near perfect validation score of 99.42% on the validation set.
- *Precision, Recall, and F1-Score:* Each instrumental sound class reached metric values above 0.98, with multiple classes even attaining the maximum value of 1.0, thus suggesting that the model was highly accurate and was able to learn from the audio clips in the training set.

Confusion Matrix: A heatmap of the confusion matrix confirmed high validation accuracy across all instrument classes, with occasional misclassifications for similar sounding instruments.(e.g., guitar occasionally misclassified as sitar or violin.).

TABLE I
CNN CLASSIFICATION METRICS

Instrument	Precision	Recall	F1
Bass	0.99	1.00	1.00
Piano	0.98	0.99	0.99
Flute	0.99	1.00	1.00
Sarod	0.98	0.98	0.98
Sitar	1.00	1.00	1.00
Guitar	1.00	0.96	0.98
Drum	1.00	0.99	0.99
Violin	1.00	1.00	1.00
Veena	0.98	0.98	0.98
Tabla	1.00	1.00	1.00

*Observations:*It was observed that the validation accuracy is very high but the model fails to generalize well with data from outside the dataset. This shows that the model was able to learn the intricate features that constitutes indian classical music instruments but since the dataset is too ideal, it is unable to generalize to the noise that accompanies real world data.

2) SVM-Based Classification: Dataset: The experiments were conducted on the IMID dataset with the removal of redundant bass and percussion classes for the purpose of better classification on non-percussive instruments.

Audio Preprocessing and Feature Extraction: The following Audio Preprocessing steps were undertaken before the experiment.

- The dataset was initially stratified by reserving 5% of the available tracks for final validation.
- The remaining dataset was divided into 80% training and 20% testing, stratified by class to maintain a balanced representation.
- Each audio clip was resampled to 22,050 Hz.

Furthermore, frame level features were obtained over time and each aggregated by their mean and standard deviation. The primary features considered are as follows:

- *Mel-Frequency Cepstral Coefficients (MFCCs):* MFCCs are a set of features that characterize a sound's timbre, or tone color. They are obtained from the audio signal's short-term power spectrum. This spectrum is mapped onto the Mel scale, a perceptual scale of pitches that listeners perceive to be equidistant from one another. The coefficients that result from this mapping, which simulates the non-linear frequency resolution of the human auditory system, provide a useful representation for describing the spectral envelope.
- Δ *MFCCs:* Δ MFCCs, or first-order derivatives, capture the rate of change of the MFCCs over time. They are computed by taking the difference between the MFCC vectors in successive frames. This feature set encodes the temporal dynamics of the sound and describes how the timbre evolves within a given note or segment. By measuring the slope of the MFCC trajectory, information about the modulation and movement in the spectral envelope is obtained.
- *Chroma STFT:* Chroma STFT (Short Term Fourier Transform) captures the harmonic content or chordal structure of a musical clip. It maps the full-frequency spectrum into 12 distinct pitch classes (C, C#, D, ..., B), independent of the octave in which they occur. This property makes chroma features highly robust to variations in timbre and transposition, focusing purely on which notes are active at a given moment. The representation is often visualized as a 12-dimensional vector where each component corresponds to the energy of one of the pitch classes
- *Tonnetz:* Tonnetz features are numerical representations derived from the Tonnetz (tonal network) that help quantify harmonic relationships in a piece of music. They represent the tonal relationship between chords and pitches.
- *Spectral Statistics:* A set of features collectively known as Spectral Statistics describes the distribution and characteristics of energy within the frequency spectrum.
  - *Spectral Centroid:* Represents the center of spectral energy and indicates the brightness of the sound, with higher values suggesting more high-frequency content.
  - *Spectral Bandwidth:* Measures the width of the frequency spectrum around the spectral centroid. Quantifies the degree of dispersion of the spectrum.
  - *Spectral roll off:* Defines the frequency below which 85–95% of the total spectral energy lies.
  - *Zero-Crossing Rate (ZCR):* Measures the rate at which the signal changes sign, i.e., how often the signal crosses the zero amplitude axis.

Performance Metrics:

- *Validation Accuracy:* The model achieved the best validation accuracy of 99.85%, indicating near-perfect performance on the internal test set based on the seven instrument classes.
- *Precision, Recall, and F1-Score:* Each instrument class achieved values above 0.99, with several classes reaching 1.0, demonstrating excellent recognition across all instruments.

TABLE II
SVM CLASSIFICATION METRICS

Instrument	Precision	Recall	F1
Piano	1.00	1.00	1.00
Flute	0.99	1.00	0.99
Sarod	0.99	1.00	0.99
Sitar	1.00	1.00	1.00
Guitar	1.00	1.00	1.00
Violin	1.00	0.99	0.99
Veena	1.00	0.99	0.99

Confusion Matrix: A heatmap of the confusion matrix confirmed high validation accuracy across all instrument classes, with minimal misclassifications.

Observations: Despite its high validation accuracy, the model does not consistently generalize data outside the dataset. However, due to its strong performance within the dataset

and its lightweight nature in comparison to traditional deep learning models, it can be considered useful for classifying monophonic clips of Indian instruments especially when better datasets are available.

3) PANN-Based Classification: Pretrained Audio Neural Networks (PANNs) are a family of deep learning CNN models that were originally trained for general audio classification and was introduced by Kong et al. (2020) in the paper "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition". PANNs, particularly the Cnn14 architecture, are trained on the AudioSet dataset which is a large-scale dataset that contains over 2 million audio clips that are 10 seconds long,across 527 audio event classes, allowing the models to learn rich spectral and temporal embeddings in various different noise settings. The pretrained embeddings produced can thus capture high-level audio features, making them suitable for tasks such as indian music instrument recognition especially when the training is limited to a small dataset, when we want a model that generalizes accurately to real world data.

Dataset: The experiments were performed on the IMID dataset, which contains 10,050 one-minute audio tracks of 10 Indian musical instruments: Flute, Veena, Sitar, Sarod, Tabla, Violin, Guitar, Piano, Bass, and Drums. The dataset was split into 80% training, 10% validation and 10% testing sets.

Audio Preprocessing and Feature Extraction:

- The audio clips were resampled to 32 kHz as required for the PANN model.
- Mono conversion was applied by averaging the stereo channels.
- Since we have a small and probably idealistic dataset, data augmentation was applied using Pitch Shift (±2 semitones) and Time Shift (±0.2 seconds) to improve model generalization.
- PANN's Cnn14 model was used to extract 2048-dimensional embeddings from each audio clip.
- The embeddings represent high-level spectral-temporal features learned from the AudioSet dataset.

Model Architecture and Fine-Tuning

- The extracted embeddings produced by the PANN model were used as input to a custom Multi-Layer Perceptron (MLP):
  - Input layer: 2048 neurons (embedding size)
  - Hidden layers: Fully connected layers, each with ReLU activation and dropout (0.5)
  - Output layer: 10 neurons (one per instrument) with softmax activation.

Performance Metrics

- *Validation Accuracy:* The model achieved a best validation accuracy of 92.23% from amongst the epochs.
- *Test Accuracy:* The overall test accuracy was 93%, confirming reliable performance on unseen data from within the dataset.
- *Precision, Recall, and F1-Score:* Most instrument classes achieved high metrics, with precision and recall above 0.80 for all instruments, and F1-scores ranging from 0.75 (Sitar) to 1.0 (Tabla), demonstrating effective recognition across the majority of Indian classical instruments.

TABLE III
PANN CLASSIFICATION METRICS

Instrument	Precision	Recall	F1
Flute	0.94	0.92	0.93
Veena	0.80	0.92	0.85
Sitar	1.00	0.60	0.75
Sarod	0.81	0.91	0.86
Tabla	1.00	1.00	1.00
Violin	0.92	0.97	0.95
Guitar	0.95	0.67	0.79
Piano	0.93	0.99	0.96
Bass	0.97	1.00	0.98
Drum	0.98	1.00	0.99

Confusion Matrix:

The confusion matrix indicated strong overall performance, with occasional misclassifications in instruments with similar timbral characteristics, like our custom CNN model. (e.g., Sitar and Guitar).

Observations

- The PANN embeddings effectively captured high-level features from Indian classical instruments despite being pretrained on a Western-centric dataset (AudioSet).
- Although this architecture has a lower accuracy than both the CNN and SVM models trained, we observed that it generalized better with data from outside the dataset..
- Some instruments with overlapping spectral characteristics (Sitar vs Guitar, Veena vs Sarod) still exhibited lower recall, suggesting that further fine-tuning or instrument-specific embeddings could improve performance.

B. Polyphonic Source Separation

1) SCNet-Based Source Separation: Dataset: The experiments were conducted using the Sanidha dataset to fine-tune the SCNet model which was already trained on 150 western songs from the MUSDB dataset consisting of 4 stems: drums, vocals, bass and other. Songs from concerts 1, 4, and 5 were used to fine-tune the SCNet model, amounting to 24 songs. Concerts 2 and 3 were used for testing as the songs from these concerts had the least amount of bleed or leakage between the isolated stems, constituting to 9 songs. To fine-tune this model, we have mapped drums to mridangam, bass to silence, western vocals to classical vocals and the other stem to the violin and tanpura stem.

Audio Preprocessing and Feature Extraction:

- All audio mixtures underwent resampling to 44,100 Hz and were handled as dual-channel stereo signals.
- The recordings got split into uniform segments of 261,120 samples each, roughly 5.9 seconds long, to maintain a steady input size for model training.
- Time-frequency representations were derived via the Short-Time Fourier Transform (STFT), employing an

FFT size of 4096, a window length matching the FFT size, and a hop length of 1024 samples.
- The spectrograms obtained served as the key time-frequency input for training the separation framework.
- These spectrograms were split into three frequency bands following preset ratios (0.230, 0.370, 0.400), enabling the network to acquire focused representations for distinct spectral zones.
- To enhance generalization, data augmentation methods were utilized, such as volume scaling of individual stems between 0.5 and 1.5, as well as stem combinations using mixup with applied chances of 0.2, 0.02, and 0.002.
- Further variability was introduced via channel-level modifications like rearranging channels, arbitrary polarity flips, and occasional signal inversions.
- Simulated MP3 compression artifacts, with bitrates varying from 32 to 320 kbps, were introduced for compression-based augmentation.
- Additional processing techniques, including pitch adjustments, injected Gaussian noise, and temporal stretching, were selectively incorporated to strengthen resilience against pitch shifts, noise interference, and tempo fluctuations.
- Occasionally, pedalboard-based effects like reverb, chorus, phaser, distortion, resampling, and bit-crushing were applied to emulate variations typical of real-world recording scenarios and acoustic environments.

Model Architecture:

The SCNet source separation model was implemented with the following architecture:

- A spectral encoder operating across multiple frequency ranges, which divides the spectrogram into three distinct bands, processing each with dedicated convolutional filters of dimensions 3, 4, and 4 respectively.
- A layered convolutional encoder where feature channels expand incrementally from 4 to 64, 128, and finally 256, permitting the extraction of progressively intricate spectral structures.
- Multiple convolutional blocks, each containing 3 successive layers, to derive hierarchical time-frequency representations across the segmented bands.
- An intensive processing unit composed of 8 sequential layers, designed to capture extended temporal relationships within the audio waveform.
- A decoder module that reassembles the isolated sources from encoded features while generating masking predictions for each target component, specifically drums (mapped to mridangam), vocals, bass (mapped to silence), and the other stem(mapped to violin and tanpura).

Performance Metrics

- *Pretraining Performance (MUSDB Dataset):* The SCNet architecture attained stem-specific SDR metrics of 9.23 dB for bass, 11.51 dB for drums, 11.05 dB for vocals, and 7.41 dB for other instrumentation.

TABLE IV
SCNET SDR ON MUSDB DATASET

Source	SDR (dB)
Bass	9.23
Drums	11.51
Vocals	11.05
Other	7.41

TABLE V
FINETUNED SCNET SDR ON SANIDHA DATASET

Source	SDR (dB)
Vocals	13.15
Mridangam	4.30
Violin + Tanpura (Other)	2.07

- *Fine-Tuned Performance (Sanidha Dataset):* After fine-tuning on the Sanidha dataset, the model achieved an SDR of 13.15 dB for vocals, 4.30 dB for mridangam, and 2.07 dB for the combined violin and tanpura stem.
- *Source Separation Quality:* High SDR values for vocal streams demonstrate robust isolation of dominant melodic content, while lower SDR values for mridangam and accompaniment stems imply increased challenges in distinguishing overlapping spectral profiles among secondary instruments.

Separation Analysis:

The separation outcomes suggest the model successfully extracted the vocal track, yielding the highest SDR values. Percussive elements like mridangam showed decent separation quality, whereas the combined stem of violin and tanpura had lower SDR scores because of overlapping harmonics and steady drone parts.

Observations

- The SCNet framework exhibited solid generalization when adapted from a Western music collection (MUSDB) to an Indian classical set (Sanidha).
- Vocals scored top SDR due to their distinct spectral and temporal traits in the mix.
- Mridangam isolation fared reasonably well, though rhythmic instruments often compete for similar frequency ranges, complicating the process.
- The weakest SDR appeared in violin and tanpura separation, hampered by lingering harmonic overlap and the sustained drone from the tanpura.

2) BS Roformer-Based Source Separation: Dataset: The experiments were conducted using the Sanidha dataset to fine-tune the BS-Roformer model which was originally trained on the MUSDB dataset containing 150 Western songs with four stems: drums, vocals, bass, and other. Songs from concerts 1, 4, and 5 were used for fine-tuning the model, amounting to 24 songs, while concerts 2 and 3 were used for testing, comprising 9 songs with minimal stem leakage. For adapting the model to Indian classical music, the original MUSDB stem mapping was modified such that the drums stem corresponded

to the mridangam, the bass stem was mapped to silence, the vocals stem represented classical vocals, and the other stem represented the combined violin and tanpura accompaniment.

Audio Preprocessing and Feature Extraction:

- All audio mixtures underwent resampling to 44,100 Hz and were handled as stereo signals featuring dual channels.
- The recordings were split into uniform segments of 261,120 samples (roughly 5.9 seconds) to ensure consistent input dimensions throughout model training.
- Time-frequency representations were derived via the Short-Time Fourier Transform (STFT), employing an FFT size of 2048, a window length of 2048 samples, and a hop length of 441 samples.
- The resulting complex spectrogram contained 1025 frequency bins, serving as the input representation for the transformer-based framework.
- This spectrogram was partitioned into several frequency bands based on predetermined groupings, allowing the model to develop specialized representations for distinct spectral regions.
- To enhance model robustness, data augmentation methods were utilized, incorporating random loudness adjustments within a 0.5 to 1.5 range and mixup-based augmentation, where stems from identical instrument classes were merged with specified probabilities.
- Additional augmentations such as channel shuffling, random polarity inversion, pitch shifting, Gaussian noise injection, and time stretching were applied selectively to increase robustness to pitch, noise, and tempo variations.

Model Architecture:

The BS-Roformer source separation model was implemented with the following architecture:

- A band-splitting spectrogram encoder that segregates the input spectrogram into multiple frequency bands, processing each independently to extract frequency-specific features.
- A transformer-based backbone comprising 8 stacked layers with a model dimension of 384, designed to learn intricate representations of time-frequency patterns in the audio signal.
- Multi-head self-attention mechanisms with 8 attention heads and a head dimension of 64, facilitating the capture of long-range dependencies across both temporal and spectral dimensions.
- Separate transformer modules operating along the time and frequency axes to effectively model temporal dynamics and spectral relationships in the audio signal.
- A mask estimation module consisting of two layers that predicts spectral masks for each target source.
- A decoder module that applies the predicted masks to the mixture spectrogram and reconstructs the separated audio stems corresponding to drums (mapped to mridangam), vocals, bass(mapped to silence) and an other stem(mapped to the violin and tanpura stem).

TABLE VI
BS-ROFORMER SDR ON MUSDB DATASET

Source	SDR (dB)
Bass	8.48
Drums	11.61
Vocals	11.08
Other	7.44

TABLE VII
FINE-TUNED BS-ROFORMER SDR ON SANIDHA DATASET

Source	SDR (dB)
Vocals	12.52
Mridangam	4.62
Violin + Tanpura (Other)	2.24

Performance Metrics

- *Pretraining Performance (MUSDB Dataset):* The BS-Roformer model's individual stem scores were 8.48 dB for bass, 11.61 dB for drums, 11.08 dB for vocals, and 7.44 dB for other instruments.
- *Fine-Tuned Performance (Sanidha Dataset):*Following fine-tuning on the Sanidha dataset, the model produced an SDR of 2.08 dB for the combined violin and tanpura stem, 4.32 dB for mridangam, and 12.52 dB for vocals.
- *Source Separation Quality:*While lower SDR values for mridangam and accompaniment stems suggest more difficulty in separating rhythm and background instruments with overlapping frequency characteristics, higher SDR values for vocals indicate strong separation performance for prominent melodic sources.

Separation Analysis:

The model successfully isolated the vocal stem with the highest SDR score, according to the separation results. The combined violin and tanpura accompaniment proved more difficult because of overlapping harmonic structures and continuous drone components, whereas percussion sources like mridangam achieved moderate separation quality.

Observations

- After being refined from a Western music dataset (MUSDB) to an Indian classical music dataset (Sanidha), the BS-Roformer architecture showed strong generalization capability.
- Due to their prominent spectral and temporal characteristics within the mixture, vocals attained the highest SDR.
- While rhythmic instruments frequently share overlapping frequency bands with other instruments, making separation more challenging, meridangam separation was somewhat successful.
- Because of their sustained harmonic content and spectral overlap, especially from the tanpura's continuous drone, the violin and tanpura stem had the lowest SDR.

3) HTDemucs-Based Source Separation: Dataset: The experiments were conducted using the Sanidha dataset to fine-tune the HTDemucs model which was originally trained on

the MUSDB dataset containing 150 Western songs with four stems: drums, vocals, bass, and other. Songs from concerts 1, 4, and 5 were used for fine-tuning the model, amounting to 24 songs, while concerts 2 and 3 were used for testing, comprising 9 songs with minimal stem leakage. For adapting the model to Indian classical music, the original MUSDB stem mapping was modified such that the drums stem corresponded to the mridangam, the bass stem was mapped to silence, the vocals stem represented classical vocals, and the other stem represented the combined violin and tanpura accompaniment.

Audio Preprocessing and Feature Extraction:

- All audio mixtures underwent resampling to 44,100 Hz and were handled as stereo signals featuring dual channels.
- To keep the input size constant during model training, the audio recordings were divided into fixed-length segments of 261,120 samples, or roughly 5.9 seconds.
- To obtain detailed spectral representations of the audio signal, time-frequency representations were extracted using the Short-Time Fourier Transform (STFT) with an FFT size of 4096 and hop length of 1024 samples.
- To stabilize optimization and enhance model convergence, the resulting spectrogram representations were normalized during training.
- The model can simultaneously capture the temporal and spectral characteristics of musical mixtures thanks to the hybrid architecture, which processes both waveform and spectrogram representations.
- To increase the robustness of the model, data augmentation techniques were used, such as random loudness scaling of stems within a range of 0.5 to 1.5 to simulate variations in recording conditions.

Model Architecture:

The HTDemucs source separation model was implemented with the following architecture:

- A hybrid encoder-decoder architecture that captures complementary temporal and spectral information by processing both frequency-domain spectrogram representations and time-domain waveforms.
  extract hierarchical audio representations, a convolutional encoder with an initial channel size of 48 gradually increases feature dimensionality across several layers.
- To maintain fine-grained temporal and spectral details during reconstruction, a U-Net-style structure with four layers of depth and skip connections between encoder and decoder stages is used.
- To model long-range dependencies across time and frequency representations, a Cross-Transformer module with five transformer layers and eight attention heads is used.
- Residual dilated convolution blocks, which enhance separation performance for overlapping audio sources and further improve intermediate representations.
- A decoder module that reconstructs the separated audio signals and predicts masks for each target source corresponding to mridangam (mapped from drums), vocals, bass (mapped to silence), and other stem(,apped to violin and tanpura stem).

TABLE VIII
HTDEMUCS SDR ON MUSDB DATASET

Source	SDR (dB)
Bass	11.76
Drums	10.88
Vocals	8.24
Other	5.74

TABLE IX
FINE-TUNED HTDEMUCS SDR ON SANIDHA DATASET

Source	SDR (dB)
Vocals	8.38
Mridangam	2.94
Violin + Tanpura (Other)	1.55

Performance Metrics

- *Pretraining Performance (MUSDB Dataset):* The HTDemucs model achieved individual stem scores of 11.76 dB for bass, 10.88 dB for drums, 8.24 dB for vocals, and 5.74 dB for other instruments.
- *Fine-Tuned Performance (Sanidha Dataset):* After fine-tuning on the Sanidha dataset, the model achieved an SDR of 8.38 dB for vocals, 2.94 dB for mridangam, and 1.55 dB for the combined violin and tanpura stem.
- *Source Separation Quality:* Although the model performed fairly well on the MUSDB dataset, the SDR values following fine-tuning show that it was still difficult to separate complex Indian classical mixtures, especially for accompaniment instruments with overlapping spectral characteristics.

Separation Analysis: Vocal extraction yielded the highest SDR values, likely due to the distinct spectral and rhythmic qualities of voice. Percussive elements, such as mridangam, exhibited lower separation accuracy, as their transient nature often coincided with harmonic instruments in shared frequency bands.The most challenging separation occurred for violin and tanpura stems, where prolonged drone characteristics and harmonic interference caused significant overlap.

Observations

- Adaptation from Western datasets like MUSDB to Indian classical music presented moderate success for the HTDemucs framework.
- Vocals retained superior isolation owing to their spectral dominance and transient clarity.
- Mridangam separation suffered due to rhythmic and spectral interactions with melodic sources.
- The lowest SDR corresponded to violin and tanpura separation, hindered by sustained frequencies and harmonic blending.

VI. Inferences

From the custom CNN architecture trained on the monophonic audio clips from the IMID dataset, we observe that the model gives an extremely high validation accuracy of 99.42% on the validation set. This suggests that the model learns to extract the appropriate features from the log mel-spectograms generated from the training set. However, we also observe that the model fails to generalize well with real world monophonic audio clips. This reveals that the model fails to classify less than ideal, noisy clips, although it is fully capable of learning these features given a more robust dataset.

The SVM model was trained on the IMID dataset by manually separating 5% of the dataset for testing. Out of the 342 testing audio clips, only 1 audio clip was wrongly classified suggesting that a simple and classical machine learning model such as SVM is able to learn the complex patterns within monophonic Indian instrument clips. It is considerably more lightweight than the CNN architecture used and slightly more accurate. However, like the CNN model, it is unable to generalize to noisy, real world audio clips and requires a better more robust dataset to effectively train the model

The PANN model trained on the IMID dataset achieved the lowest accuracy out of the other 2 models that were trained. However, this model generalizes well with noisy real world audio clips as it is able to capture better feature embeddings from the IMID dataset used as it is trained on a huge dataset based on real world audio clips. this is the inference for monophonic classification, i want u to add inference for polyphonic source separation to this and also u do relevant research and highlight which model is best in terms of accuracy and computation trade off. i was surprised that bs roformer performed so well, see if its fit to mention that given that its a new model that is still being explored. also explain how it works well for indian classical

TABLE X
Performance Comparison of Source Separation Models

Model	Vocals	Mridangam	Violin + Tanpura
SCNet	**13.15**	4.30	2.07
BS-Roformer	12.52	**4.62**	**2.24**
HTDemucs	8.38	2.94	1.55

For the task of polyphonic source separation, three state-of-the-art architectures were evaluated: SCNet, BS-Roformer, and HTDemucs. All three models were originally pretrained on the MUSDB dataset and subsequently fine-tuned on the Sanidha dataset containing Indian classical music recordings.

From the experimental results, it is observed that SCNet achieved the best vocal separation performance on the Sanidha dataset, obtaining an SDR of 13.15 dB. The BS-Roformer model achieved a slightly lower SDR of 12.52 dB for vocals but performed marginally better in separating the mridangam and accompaniment stems. In contrast, HTDemucs achieved lower SDR scores overall, with 8.38 dB for vocals, 2.94 dB for mridangam, and 1.55 dB for the violin and tanpura stem.

These results suggest that SCNet and BS-Roformer are better suited for separating Indian classical music mixtures compared to HTDemucs. SCNet benefits from its multi-band spectral encoder and hierarchical convolutional structure, which enables it to learn specialized representations across different frequency regions. This is particularly beneficial for Indian classical music, where instruments such as the tanpura produce continuous harmonic drones while melodic instruments such as the violin occupy overlapping spectral regions.

Interestingly, the BS-Roformer model showed marginally better performance to the SCNet architecture employed in this research. The band-splitting transformer model permits the model to process the various spectral bands separately and with transformer attention to capture long range dependant temporal and frequency fatigue. This design is particularly suited to Indian classical music, in which rhythmic patterns from the instruments like mridangam intermingle with drone accompaniments. These are learned by the model by using transformer-based architecture which is able to understand and capture time dependencies better than convolutional architectures.

The other notable observation is that BS-Roformer has good separation performance even with its computational complexity. Transformer-based architectures are generally computationally intensive with the attention mechanisms used to model long-range dependencies. However, the band-splitting strategy is used to cope with this complexity by dividing the spectrogram to be broken down into smaller frequency bands, which make it possible to. model to process individual bands but still capture. global relations between signal. This enables the model to scale well with a maintained separation quality across different sources.

Although considered a robust baseline in western music source separation HTDemucs performed worse when applied to the Indian classical music mixtures of this study. One of the possible reasons is that HTDemucs is overreliant on waveform-domain processing over spectrogram features, which might not reflect the subtle spectral differences required to separate highly overlapping harmonic sources such as violin and tanpura.

On the whole, the findings show that SCNet offers the best vocal separation accuracy on the Sanidha dataset, and BS- Roformer provides comparable performance in vocals and marginally better performance on the two other stems. Although being relatively unexplored, BS-Roformer performs very well in the domain of Indian classical music, comparable to the performance of SCNet on a number of separation metrics. Its long-range time-frequency modelling capability renders it especially promising with respect to complex music traditions like the Indian classical music, in which sus- stained drones, fancy melodic decorations, and rhythmic percussion coexist in the same mixture.

VII. Conclusion

The study presents a detailed framework of studying Indian classical music based on the joint work of the monophonic instrument classification and polyphonic source separation. The paper compares various machine learning and deep learning models that are typically used in the analysis of Western music and examines their performance when used on Indian classical music datasets.

In the case of monophonic instrument classification, Ten instrument classes (flute, veena, sitar, sarod, tabla, violin, guitar, piano, bass and drums) of IMID dataset were experimented. Three models were tested: a custom Convolutional Neural Network (CNN) and a classical Support Vector Machine (SVM) and a pretrained model based on PANN-based embedding. Both CNN and SVM models obtained very high validation rates in the dataset that show that they can learn characteristic timbral information on clean monophonic samples. But both models were presented as having limited levels of generalization, on noisy real-world audio snippets. Conversely, the PANN-based method has slightly lower accuracy on the dataset but a superior generalization capability owing to the rich audio representations acquired under large scale pretraining of various real-world audio data. This points to the relevance of pretrained audio embeddings in the context of limited or idealized data. It also demonstrates that models like PANN which was trained using huge and diverse datasets can easily be adopted to specific classification tasks even with minimal finetuning.

The next step in the study was focused on polyphonic source separation with the use of the Sanidha corpus, a series of recordings of Indian classical performances. Three state-of-the-art frameworks were evaluated which include SCNet, BS-Roformer, and HTDemucs, which have been pretrained on MUSDB collection after which they are fine-tuned to isolate mridangam, vocal tracks, and complementary stems, including violin and tanpura. Experimental results show that SCNet achieved higher vocal separation effectiveness with SDR of 13.15 dB, as compared to BS-Roformer which had slightly better separation of mridangam and certain accompaniment parts. The results of HTDemucs provided relatively lower values of SDR, indicating that the waveform-based models may have problems with the separation of highly intertwined harmonic sources common in Indian classical music.

The implication of these results is that both SCNet and BS-Roformer are potentially usable frameworks in order to repurpose Western-trained separation models in the Indian classical settings. SCNet takes advantage of its multiband convolutional architecture, allowing custom windowed processing of individual frequency bands, whereas BS-Roformer uses transformer-based attention to capture long-term and long-spectrum interactions. These properties can be particularly relevant in Indian classical music, where percussive rhythms are intermingled with melodic extempore over a constant drone sound produced by instruments such as the tanpura in an acoustic space.

Taken together, this work demonstrates that modern audio machine learning systems could be successfully reused in the analysis of Indian classical music, despite the fact that they were originally trained on Western-biased samples. Combining monophonic instruments classification with polyphonic separation, the methodology proposed will contribute to a higher level of computational understanding of this musical tradition. Future applications include automated music recognition, preservation of classical versions in a digital form, and instructional tools that can be used to make learning easier.

References

[1] J. J. Bosch et al., "A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals," in Proc. ISMIR, 2012, pp. 559–564.

[2] J. Liu and L. Xie, "SVM-Based Automatic Classification of Musical Instruments," IEEE Conf. on Intelligent Computation Technology and Automation, 2010.

[3] S. Gaikwad et al., "Classification of Indian Classical Instruments Using Spectral and PCA-Based Cepstrum Features," IEEE, 2014.

[4] E. J. Humphrey et al., "OpenMIC-2018: An Open Dataset for Multiple Instrument Recognition," Spotify/NYU, 2018.

[5] R. Bittner et al., "MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research," ISMIR, 2014.

[6] S. Rouard, F. Massa, and A. Défossez, "Hybrid Transformers for Music Source Separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023, pp. 1–5.

[7] Q. Kong et al., "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," IEEE/ACM TASLP, 2020.

[8] P. Patel et al., "Audio Separation and Classification of Indian Classical Instruments," Eng. Applications of AI, 2024.

[9] V. V. Krishnan et al., "Sanidha: A Studio Quality Multi-Modal Dataset for Carnatic Music," arXiv, 2025.

[10] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Demucs: Deep extractor for music sources with extra unlabeled data remixed," in *Proc. Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, 2021.

[11] S. M. Elghamrawy and S. E. Ibrahim, "Audio signal processing and musical instrument detection using deep learning techniques," Academic Report, 2021.

[12] W.-T. Lu, J.-C. Wang, Q. Kong, and Y.-N. Hung, "Music Source Separation with Band-Split RoPE Transformer," *arXiv preprint arXiv:2309.02612*, 2023.

[13] W. Tong, J. Zhu, J. Chen, S. Kang, T. Jiang, Y. Li, Z. Wu, and H. Meng, "SCNet: Sparse Compression Network for Music Source Separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2024.

[14] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Zenodo, Dec. 2017.