Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Application of VGGish and YAMnet Model for North Indian Raga Music Recognition using Transfer Learning

Annagha A. Bidkar[1], Amogh Thakur[2], Yogesh H. Dandawate[3]

[1]Research Scholar, Vishwakarma Institute of Information Technology, Electronics and Telecommunication Department, Pune and Assistant Professor, SCTR's Pune Institute of Computer Technology, affiliated to SPPU, India, anagha109@gmail.com
[2]Student, SCTR's Pune Institute of Computer Technology, affiliated to SPPU, India
[3]Professor, Vishwakarma Institute of Information Technology, Electronics and Telecommunication Department, Pune, affiliated to SPPU, India

Raga is the foundational framework for Indian classical music (ICM), consisting of successive notes. This same principle applies to both the Carnatic and Hindustani Classical music traditions. The significant benefits of recognizing ragas from audio go beyond Music Information Retrieval (MIR), which includes content filtering, teaching/learning, and music therapy. This research paper demonstrates a method for recognizing Hindustani classical ragas using transfer learning with pre-trained models VGGish and YAMnet from raw audio spectrograms. VGGish was particularly developed to extract audio features or embeddings from raw audio streams, whereas YAMNet is a deep-learning model designed for identifying and classifying a wide range of sound events. These models are useful for audio analysis and understanding across a wide range of applications. The proposed techniques obtained an overall testing accuracy of 96.88% for VGGish and 93.3% for YAMnet on a dataset of 200+ samples of 12 ragas played on four distinct musical instruments. The standard confusion matrix is used to determine the study results for these models.

Keywords: Raga recognition, North Indian raga Music, Transfer learning, YAMnet, VGGishnet.

1. Introduction

Music is an artistic medium for expression. Indian music reflects its cultural legacy through an array of genres, ragas, and styles. Also recognized as Raga-Sangeet (Raga-Music), the concept of raga is the core component of Indian classical music. In Indian classical music, typically referred to as Hindustani classical music and Carnatic music, raga is an important component [1],[2]. In Indian Classical Music, there are 12 notes (5 deviated and 2 undeviated notes) and 22 microtones [3].

The Hindustani Classical System of music includes a variety of ragas, which are further classified into ten ‘Thaats’. A ‘Thaat’ comprises related ragas based on their notations and frequency [4]. A raga is identified by its phrases of notes and the individual notes used in the raga. A unique framework of these notes organized in ascending (Aaroha) and descending (Avroha) sequences is called a raga [5]. In some cases, the sequence of notes is the same but the most important note (Vaadi) and the second important note (Samvadi) help in distinguishing between these ragas [6]. Pitch values in Western music are set and predefined, whereas in Indian classical music, they vary according to the beginning note (Sa), which is the initial pitch. The notes are proportional to the preceding note and are referred to as the performer's Tonic Pitch [7]. This intricate framework and progression of notes make it intriguing and difficult to recognize raga in Indian Classical Music.

The key objective of this research is to develop a raga classification model for Indian classical music which is based on deep transfer learning and capable of recognizing ragas. Ragas can be identified from music in a variety of ways. Previous attempts included Hidden Markov, Basic Rule-Base Heuristics, and basic rule-based Pitch Tracking. Models are computationally inefficient and highly complex [8]. The Comp Music dataset was published by Gulati et al. and was also utilized in our investigation [9]. Deep learning models have the potential to significantly improve classification accuracy and processing efficiency. The use of deep learning-based algorithms to detect musical genres has become more common among academics due to the expansion of deep learning in various fields, such as Computer Vision (CV), Speech Recognition (SR), and Natural Language Processing (NLP). Several methods use the distinct properties or feature combinations of the deep learning model to classify sounds or music. This study suggests a method for raga recognition that makes use of YAMnet, VGGish, and a pipeline to process a WAV audio file and identify the raga.

2. Literature Survey

The significance of raga in Indian classical music has led to numerous attempts to identify it from the audio. As mentioned earlier, ragas are collections of notes in order. Additionally, it is computationally easy to extract notes from audio. Authors extracted the notes from the audio and then matched them with the already-known notes of the ragas in numerous works, including [10]. Gulati et al. [11] employed a vector space model to describe audio data in their study. For recognizing the ragas, they developed a prediction model that was analogous to vector space models used in natural language processing. They achieved 70% accuracy in classifying 40 ragas and 92% accuracy in classifying a subset of 10 ragas.

Melakarta Raga Recognition is one of the intriguing applications of KNN - which is mostly utilized in raga identification. In many circumstances, raga detection is regarded as a fundamental nearest-neighbor issue. The challenge is made more interesting by the fact that the material is audio - given audio, discover the audio closest to the query in the trained database. The logic underpinning Nearest Neighbor Classification [12] is simple: objects are categorized based on the class of their nearest neighbors. It is typically helpful to include more than one neighbor, due to which the approach is generally known as k-nearest Neighbor (k-NN) Classification, in which k nearest neighbors are utilized to determine the class.

The search phrase is received in the form of a humming song, and the melody pattern that fits the query's pitch rise and fall is obtained based on the query's pitch rise and fall. The melody retrieval is based on characteristics such as distance measurements and gestalt principles. The technique [13] is predicated on low-level signal characteristics, and the raga is identified by considering various instrument signals as input to our system. Speech processing could be used to categorize classical music using Spectrograms, Scalograms, and MFCCs, among other features. The sound and music features are investigated and extracted in order to accomplish the categorization of various music genres. The first phase, which utilized music signals, is discussed. Along with the methods used, the pitch class profiles-based features and their acoustic characteristics-based statistical measurements are taken into account.

CNNs were first designed to classify and identify images, but they were later extended to classify sounds as well. The promising method known as transfer learning aims to retrain previously trained networks using different datasets. [13] The CNNs GoogLeNet, SqueezeNet, ShuffleNet, VGGish, and YAMNet—two "Sound" and three "Image" learnt—were trained by transfer learning.

For the categorization of auditory situations, spectrogram representations show competitive performance. [14] However, the spectrogram alone does not account for a sizable quantity of time-frequency data. In this paper, we provide a method for examining the advantages of segmented deep scalogram representations taken from an audio stream. The method first converts the segmented acoustic scenes into bump and morse scalograms as well as spectrograms. Next, the spectrograms or scalograms are sent into pre-trained convolutional neural networks. Third, the features extracted from a subsequent fully connected layer are fed into (bidirectional) gated recurrent neural networks. Finally, predictions from these three systems are made.

The approach seen in “Classification of Thaats in Hindustani Classical Music using Supervised Learning” consists of recognizing the ‘Thaat’ rather than the raga. This might be tough to match with the notes being utilized [15]. Similarly, another raga recognition approach was based on the ‘Thaat’, or upper-level categorization of ragas, followed by raga recognition. This additional step might add complication to recognizing the raga. Performing this realistically might result in a maximum accuracy of 97% and a minimum accuracy of 93% [16].

Signal processing techniques were employed in 2012 by Preeti Rao et al. to extract particular musical knowledge from audio signals, such as descriptors for the melody or rhythm. Within the tradition's musicological foundation, audio signal processing techniques and data formats are studied for various retrieval purposes [17]. Ms. P. Kirthika, et al. proposed using Linear Predictive Coding (LPC) to use frequency to determine a raga in their research paper from 2014, which introduced audio feature extraction for classifying music based on the raga, which is important in Music Information Retrieval (MIR) systems. They also provide a technique employing Latent Semantic Indexing (LSI) [18] to determine the raga of the input audio sample.

3. Motivation

Indian Classical Music is a complex topic that has yet to be thoroughly studied and deep learning can be an excellent computational tool for exploring this field. Studying this field and increasing the accuracy of past studies has always been difficult. This research tries to build a more robust while including deep learning techniques to increase the capacity of raga recognition.

4. Methodology

4.1 Dataset and Feature

4.1.1 Dataset

The dataset, which consists of multiple audio files, is converted from wav data to a spectrogram, which displays the input signal's frequency spectrum as it varies over time. In this example, the wav file. The wave file data is split into training and testing data at a ratio of 8:2.
An essential first step in this kind of research is to evaluate the dataset's importance. The application was taken into consideration when designing and creating a vibrant dataset.

The dataset consists of a raga from each time cycle from a day, known as 'Raga Chakra' and 'RagaPrahar'. While designing the dataset, four separate instruments were used: sitar, sarod, santoor, and flute.

Table 4.1 gives a count of the ragas and the instruments that have been used for the training and testing of the model. Each raga includes almost 250 sample tracks of each instrument. This table presents the distribution of samples for different ragas played on four instruments —Sitar, Sarod, Santoor, and Flute—along with the collective count, which is the total number of samples for each raga across all instruments.

Table 4.1: Total count of Music tracks used

Raga Name	Sitar	Sarod	Santoor	Flute	Total count
Bageshree	72	56	87	150	365
Bhairav	59	73	56	52	240
Lalit	83	56	57	117	313
Madhuwanti	64	106	56	68	294
Ahir Bhairav	56	70	64	89	279
Bihag	71	143	63	81	358
Malkauns	56	60	110	115	341
Miya Ki Todi	77	87	51	40	255
Shuddha Sarang	60	58	54	66	238
Yaman	73	184	81	111	449
Bhimpalas	73	123	109	124	429
Puriya Kalyan	53	76	64	80	273
Total Count	797	1092	852	1093	3834

Fig 4.1 gives a graphical representation of a cumulative count of the number of audio files along with the individual count of instruments and the ragas. The total collective count across all ragas and instruments is 3,834 samples. This shows how many samples of each raga were collected from different instruments, highlighting the diversity of the dataset.

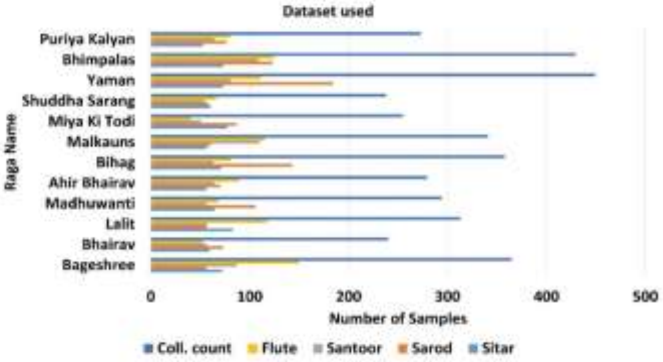


Fig 4.1 Graph of total collective count of music tracks used

It appears that Yaman and Bhimpalas have the most samples, while Bhairav has the fewest similarly, sitar and flute have the highest number of total samples, closely followed by the sarod, with the santoor having slightly fewer samples. For this project, 12 ragas were used for the analysis and prediction of the raga and tracks played on 4 instruments flute, santoor, sarod, and sitar were used. These music tracks were converted into a Log-Mel-spectrograms format for the model training.

4.1.2 Mel spectrogram

Log Mel spectrograms provide a compact and effective representation of the frequency content of an audio transmission over time. Fig 4.2. showcases the images of log mel-spectrogram for 12 ragas.

With this, the accuracy of the trained model can be calculated. For each pre-trained model, the image is converted into a log Mel-spectrogram in a frame of size 94 × 94 for processing and training. The findings are shown in the confusion matrix to calculate the values of false positive, false negative, true positive, and true negative.

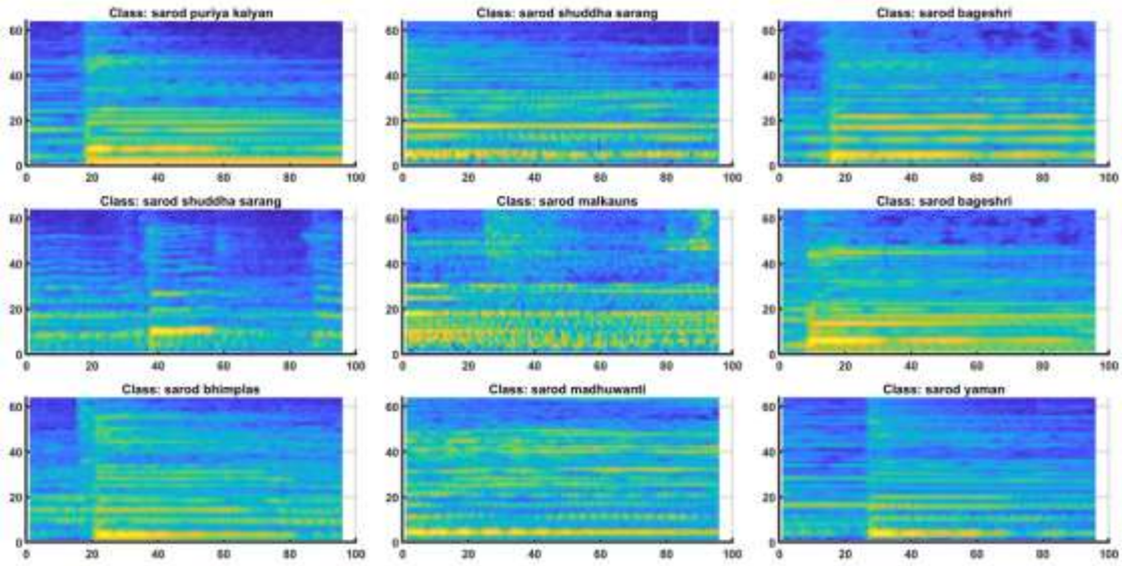


Fig 4.2 Mel-spectrogram of some sample of audio Ragas

4.2 CNN Models

4.2.1 VGGish

In VGGish approach, 24 layers are used to process the input image data. The preprocessing of the images is done with about 6 layers of convolutions, Rectified Linear Unit (ReLU), and MaxPooling. A 128 fully connected layer is obtained and then the image is processed using VGGish. The input to VGG-based convNet is a 224×224 RGB image. The preprocessing layer takes the RGB image with pixel values in the range of 0–255 and subtracts the mean image values which is calculated over the entire ImageNet training set.

The VGGish model is a Convolutional Neural Network (CNN) model, specifically designed for audio analysis tasks. It is based on the architecture of VGG16, a well-known CNN model originally used for image classification, but adapted to work with audio data. VGGish was developed by Google for tasks like audio feature extraction, sound event detection, and speech recognition. Here's a more detailed explanation of the model:

Key Features of the VGGish CNN Model:

Architecture: VGGish is a variation of the VGG16 architecture, which consists of multiple layers of convolutions followed by max-pooling layers. It extracts features, including pictures or spectrograms—which are graphic depictions of audio signals—from the incoming data. A series of convolutional layers with tiny filters (usually 3x3) are included in the model, which gradually extracts features like edges, textures, and increasingly intricate patterns. Fully linked layers come next, which categorize the input according to the features that have been learnt.

Input Type: VGGish processes audio input instead of picture data. A spectrogram, which is a two-dimensional representation of the audio frequencies across time, is commonly used to depict the audio input. As a result, the audio issue becomes one that can be studied in a manner related to that of picture categorization. The model works well for tasks like voice recognition, music analysis, and environmental sound identification because it uses mel-spectrograms, which are a transformation of audio signals that emphasize human-audible frequencies, as input.

Audio Embeddings: VGGish is often used to generate audio embeddings—high-dimensional representations of audio features. Compact representations of audio data, known as embeddings, simplify tasks like audio categorization and pattern recognition by simplifying key attributes like pitch, timbre, and rhythm. These embeddings are typically used for downstream tasks like speaker identification, music genre categorization, and raga recognition, or they are transmitted to other machine learning models.

Transfer Learning: VGGish is often used for transfer learning, where a pre-trained model on a large dataset (such as the AudioSet dataset, containing millions of labeled audio events) is fine-tuned for a specific task, such as Raga identification. By using pre-trained VGGish, researchers can leverage the model’s ability to generalize across a wide range of audio tasks without requiring large amounts of task-specific labeled data.

Application to Raga Recognition: VGGish derives salient characteristics from audio recordings of ragas performed on different instruments in the context of raga recognition. It can recognise patterns peculiar to individual ragas, like note sequences, rhythms, and melodic structures, by processing the audio's spectrograms. Even when the training data contains sounds from several instruments, such as the flute, sitar, or sarod, VGGish may still be used to recognise ragas across numerous instruments due to its good generalisation over different sound types.

Advantages of VGGish:

Pre-trained on Large Audio Datasets: VGGish is pre-trained on massive datasets like AudioSet, so it can recognize a wide variety of sound patterns, making it useful for a range of tasks.

Compact and Efficient Embeddings: The embeddings generated by VGGish are a condensed and efficient representation of the original audio, which can be used to train other models more easily.

Strong Generalization: Because it is based on the VGG architecture and pre-trained on diverse data, VGGish performs well across various sound categories, including music, speech, and environmental sounds.

Limitations:

Not Specialized for Music: While VGGish performs well across various audio tasks, it is not specifically designed for music recognition, which may make it less effective in capturing the nuances of certain musical elements like harmony and complex rhythms, compared to models designed specifically for music analysis.

Requires Spectrogram Input: VGGish requires audio to be pre-processed into a spectrogram format, which may require additional computation compared to models that work directly on raw audio. Overall, the VGGish CNN model is a powerful tool for extracting audio features and performing classification tasks, especially in scenarios where transfer learning is useful. Its strong performance in diverse audio analysis tasks makes it a popular choice for applications like Raga identification in Indian classical music.

4.2.2 YAMNet

In YAMNet 14 layers of convolution, batch normalization, ReLU, and grouped convolution are to preprocess the images. After which a 512 fully connected layer is obtained. The pre-trained neural network YAMNet uses the MobileNetV1 depth-wise-separable convolution architecture. It has the ability to independently forecast all 521 audio occurrences in the AudioSet corpus given an audio waveform as input.

Convolutional neural networks (CNNs) like the YAMNet model were developed especially for sound event detection and classification applications. It is designed to function with a broad variety of ambient noises, spoken language, and musical compositions, utilising deep learning methodologies to identify and categorise various audio occurrences. YAMNet is pre-trained, similar to VGGish, but it is tuned for a wider range of sound recognition tasks.

Key Features of the YAMNet CNN Model:

Architecture: The foundation of YAMNet is a MobileNet-like architecture that is intended to be useful for audio classification while remaining lightweight and efficient. Compared to conventional convolutional networks, MobileNet is more efficient because it makes use of depthwise separable convolutions, which lower the amount of parameters and computation required. The model gradually extracts features from the audio input through a number of layers of convolution and pooling procedures. The actual categorisation task is carried out by the fully connected final layers.

Input Type: YAMNet works directly with audio waveform data, which is transformed into a log-mel spectrogram before being fed into the model. In speech and music analysis, the log-mel spectrogram is a popular representation of audio that records both the frequency content and the loudness with time. As this spectrogram format highlights the lower frequency components that are more pertinent to human hearing, it is especially well-suited for detecting sound occurrences.

Audio Event Classification: Audio waveform data, which is converted into a log-mel spectrogram and then input into the model, is directly used by YAMNet. In speech and music analysis, the log-mel spectrogram is a popular representation of audio that records both the frequency content and the loudness with time. As this spectrogram format highlights the lower frequency components that are more pertinent to human hearing, it is especially well-suited for detecting sound occurrences.

Transfer Learning: After being pre-trained on a general dataset like AudioSet, the model can be fine-tuned on a smaller, more specific dataset (such as one containing ragas) to improve performance in a particular task. This makes YAMNet an excellent choice for tasks like Raga identification because it already understands a wide variety of audio signals, which can be adapted to recognize specific features of Indian classical music ragas.

Application to Raga Recognition: In the context of Raga recognition, YAMNet can be trained to classify ragas from different musical instruments by detecting the unique patterns of sound events that make up each raga. It searches for distinct melodic progressions, rhythmic components, and tone characteristics that set one raga apart from another. Given that YAMNet is intended for general sound classification, it may be able to distinguish even minute variations between audio samples, which is crucial for differentiating various ragas that may have similar sounds.

Compact Audio Embeddings: YAMNet also generates audio embeddings, similar to VGGish, which are compact representations of the input audio. These embeddings, which can be applied to more complex classification tasks, capture important aspects of the sound, including pitch, rhythm, and timbre. These embeddings offer a streamlined and effective method of analysing complicated audio data because they may be utilised as inputs for other machine learning models.

Versatility: Because YAMNet is trained to recognize so many types of sounds, it can be applied in a variety of contexts beyond music. For example, it can be used in environmental sound detection, speech recognition, acoustic scene classification, and more.

Strengths of YAMNet:

Pre-trained on a Wide Range of Audio: YAMNet has been trained on AudioSet, which contains a diverse set of audio events, making it highly adaptable and versatile.

Efficient Architecture: The MobileNet-based architecture is lightweight and fast, making YAMNet efficient for deployment in applications where computational resources are limited, such as mobile or embedded devices.

Generalized Sound Recognition: YAMNet’s ability to recognize over 500 different sound categories makes it suitable for many audio analysis tasks, from music recognition to noise detection.

Limitations of YAMNet:

Not Specialized for Music: Like VGGish, YAMNet is not specifically tailored to music, and while it can perform well in recognizing sound events in general, it might miss some of the more subtle musical nuances found in genres like classical music or complex ragas.

May Struggle with Complex Audio: Because YAMNet is trained on broad sound categories, it might have difficulty distinguishing between very similar sound events or audio samples, such as closely related ragas, without further fine-tuning.

Comparison with VGGish:

Architecture: While VGGish is based on the VGG16 architecture, which is larger and more complex, YAMNet is based on the MobileNet architecture, which is more lightweight and efficient.

Application: YAMNet is more general in its approach, trained to detect a broader range of sound events, whereas VGGish is optimized for extracting audio embeddings for tasks like music analysis.

Efficiency: YAMNet is typically more efficient and faster in terms of processing, due to its lightweight design, but VGGish may perform better in extracting richer audio features for music-related tasks.

The YAMNet CNN model is a powerful and flexible tool for recognizing a wide variety of sound events, including music, environmental noises, and human speech. Though more specialised audio tasks would require fine-tuning, its lightweight architecture and adaptability make it well-suited for applications like Raga recognition. The versatility of YAMNet allows it to function well in a wide range of acoustic settings. This is one of its main strengths.

YAMNet is a powerful CNN model tailored for sound event classification, capable of recognizing a wide variety of sounds in noisy environments. It works effectively with mel-spectrograms and can be adapted for tasks like Raga recognition through transfer learning.

Though not specifically designed for music, YAMNet's strong pre-training and efficient architecture make it useful for general audio classification tasks, including music analysis, when fine-tuned for such purposes.

4.3 Experimental Set-up

In this experiment, the focus is on Raga identification using two deep learning models, VGGish and YAMNet, applied to audio recordings from different musical instruments. The experiment is conducted in two parts:

1. Individual Instrument Experimentation:

The first part of the experiment analyzes how well the models can identify ragas played on specific musical instruments (e.g., sitar, sarod, santoor, and flute). Each instrument's audio recordings are processed separately, and the models attempt to classify the raga based on the instrument's sound profile. This process allows the researchers to evaluate how effectively the models perform on each instrument, identifying any variations in accuracy due to the specific characteristics of the instruments.

VGGish Model: With the use of this technique, one can capture more complex characteristics of sound by extracting audio features, or embeddings, from unprocessed audio spectrograms. In order to determine how well it can identify the raga from those specific sound patterns, it examines each instrument independently.

YAMNet Model: Music recognition is one of the areas of expertise for this deep learning model. To assess its accuracy in determining the raga from each instrument's sounds, it is similarly applied to the recordings of the individual instruments.

2. Combined Raga Set from All Instruments:

In the next phase, the models are tested with a variety of audio samples from the sitar, sarod, santoor, and flute since the research merges the raga data from all instruments into a single set. This resembles a more intricate situation in which the models must determine the raga regardless of the instrument being performed.

VGGish and YAMNet now work on a larger dataset that includes samples from all instruments. The goal is to see how well these models generalize across different instruments and whether they can consistently recognize ragas when instrument variation is introduced.

Key Insights:

Model Performance Across Instruments: The study aims to determine whether the models' performance is enhanced by the unique sound signatures of individual instruments or impeded by them.

Cross-Instrument Generalization: By combining samples from all instruments, the experiment tests the models' ability to recognize ragas in a more generalized and diverse audio environment.

Accuracy Comparison: The models' performance is evaluated to determine which one works better for individual instruments and in the combined setup: VGGish or YAMNet. It demonstrates how each model adjusts to the various instruments' levels of complexity as well as the overall dataset.

This experimentation is valuable because it simulates real-world scenarios where raga recognition needs to work across various musical contexts, offering insights into the robustness of each model in handling diverse musical data as shown in Fig. 4.3.

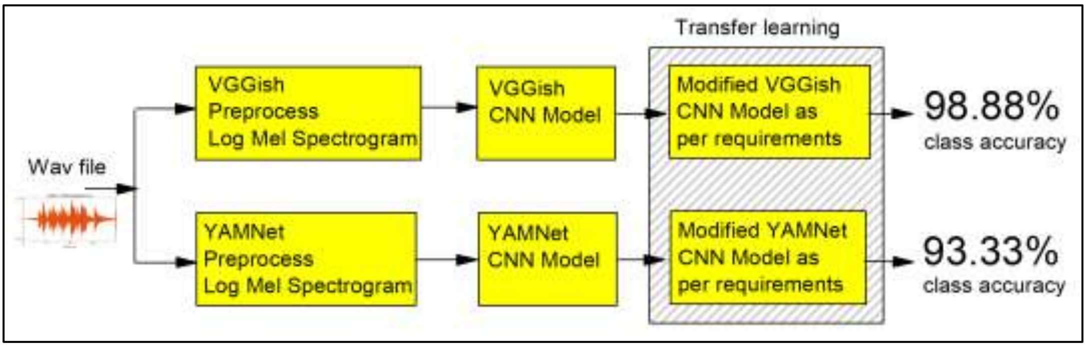
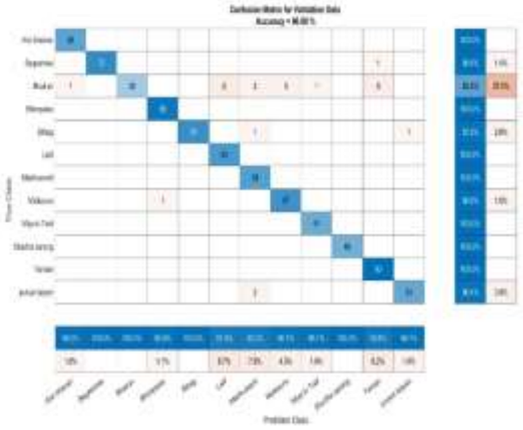

Transfer learning
Wav file
VGGish Preprocess Log Mel Spectrogram
VGGish CNN Model
Modified VGGish CNN Model as per requirements
98.88% class accuracy
YAMNet Preprocess Log Mel Spectrogram
YAMNet CNN Model
Modified YAMNet CNN Model as per requirements
93.33% class accuracy

Fig 4.3 Methodology for experiments

5. Results and Discussion

The training and testing datasets are used for experimentation with both classifiers. Accuracy and precision are used to evaluate the classifiers' performance. Performance analysis of database using VGGish and YAMNet deep transfer learning model is tested and shown in Fig 5.1 and Fig 5.2 in a confusion matrix.



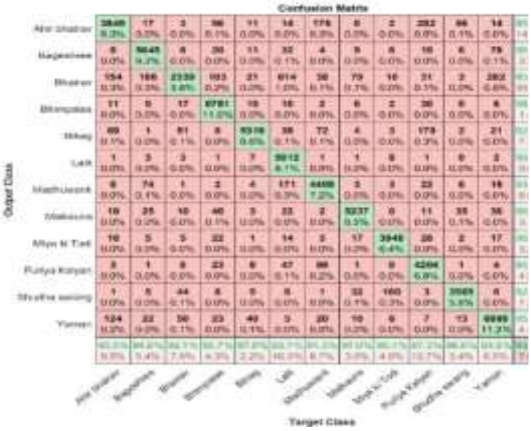


Fig 5.1 Confusion Matrix of VGGish

Fig 5.2 Confusion Matrix of YAMnet

Confusion matrix parameters shows the output of the anticipated class in comparison to the output of the actual class. A table with two rows and two columns that shows the number of true positives, false negatives, false positives, and true negatives is known as a table of confusion (also known as a confusion matrix). This makes it possible to analyse data in greater detail than just looking at the percentage of accurate classifications (accuracy). If the data set is imbalanced, meaning that there are large differences in the amount of observations between classes, accuracy will produce deceptive results. An error matrix is a particular table arrangement that makes it possible to visualize the performance of an algorithm, usually one that uses supervised learning.

Details are provided in the following Table 5.1 and Table 5.2.

Table 5.1. Accuracy score of all individual ragas for each instrument and a collective accuracy count using the VGGish classifier.

Table 5.1 Accuracy score using VGGish Classifier

Raga Name	Sitar	Sarod	Santoor	Flute	Collective count
Bageshree	42.9	100	100	100	98.6
Bhairav	100	100	100	100	62.5
Lalit	100	100	100	100	100
Madhuwanti	100	100	100	100	100
Ahir Bhairav	100	100	100	100	100
Bihag	100	100	100	100	97.2
Malkauns	81.8	100	100	100	98.5
Miya Ki Todi	100	64.7	80	100	100
Shuddha Sarang	100	100	100	100	100
Yaman	100	100	93.8	100	100
Bhimpalas	100	100	100	100	100
Puriya Kalyan	100	100	100	100	96.4
Accuracy	93.75	97.26	98.24	100	96.88

Table 5.2. Accuracy score of all individual ragas for each instrument and a collective accuracy count using the YAMNet classifier.

Table 5.2 Accuracy score using YAMnet classifier

Raga Name	Sitar	Sarod	Santoor	Flute	Collective count
Bageshree	58.8	98.8	100	99.8	96.7
Bhairav	95.8	95.3	100	94.6	60.9
Lalit	92.9	99.5	100	100	99.4
Madhuwanti	73.8	100	84.8	83.8	93.4
Ahir Bhairav	96.7	98.6	99.8	99.6	85.9
Bihag	93.9	99.2	99.3	99.9	92.3
Malkauns	80.6	95.1	99	99.7	96.3
Miya Ki Todi	96.2	65	78.6	88.3	96.8
Shuddha Sarang	100	96	96.9	93.2	92.8
Yaman	99.3	99.1	87	94.9	95.6
Bhimpalas	99.6	96.8	99.5	99	98.6
Puriya Kalyan	80.3	99.7	93.9	99.6	95.5
Accuracy	89.3	95.7	95.6	97.1	93

The table 5.1 , 5.2 and Fig. 5.1 , 5.2 provides a comparison between the VGGish and YAMNet models for the task of Raga identification using different musical instruments such as Sitar, Sarod, Santoor, and Flute, along with the collective count of all instruments.

The VGGish model shows consistently high performance for Raga identification across most instruments, with perfect or near-perfect accuracy for several ragas. For instruments like Sitar, Sarod, Santoor, and Flute, most ragas have been identified with 100% accuracy. However, there are some variations, particularly in the Raga Bageshree, where the Sitar accuracy is much lower (42.9%). Malkauns also shows lower accuracy with Sitar (81.8%).

The collective count (combined accuracy for all instruments) gives an overall idea of the Raga identification accuracy when all instruments are considered together. The average accuracy for the collective count is 96.88%.

Highest performing Ragas such as Bhairav, Lalit, Ahir Bhairav, and Bihag have consistently high accuracy across all instruments.

There is a slight dips; notably, in Raga Miya Ki Todi, the Sarod accuracy is only 64.7%. Despite this, the collective count remains 100%.

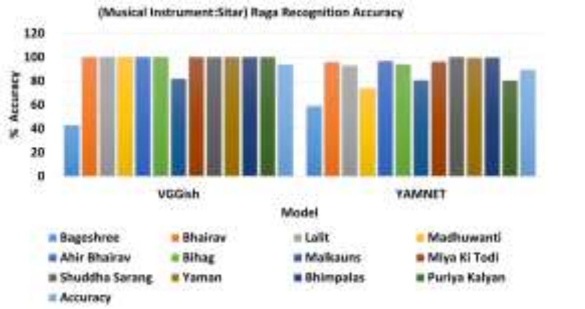


Fig 5.3a Raga recognition using Sitar

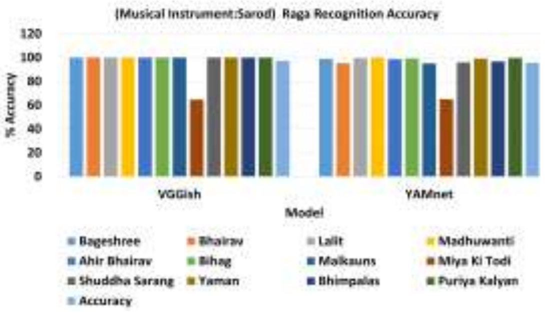


Fig 5.3b Raga recognition using Sarod

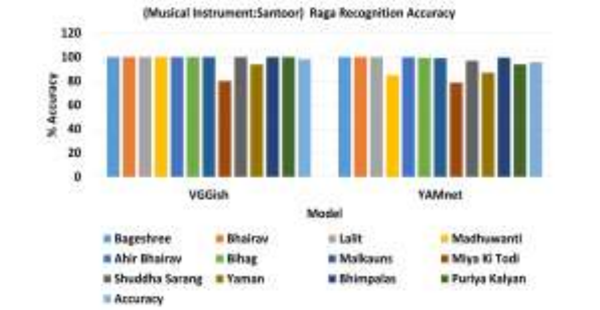


Fig 5.3c Raga recognition using Santoor

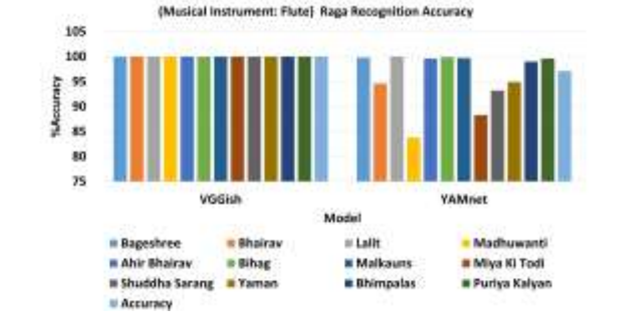


Fig 5.3d Raga recognition using Flute

YAMNet Model:

The YAMNet model also performs impressively, but with slightly more variation compared to the VGGish model. Sitar accuracy tends to be lower in certain ragas, particularly in Bageshree (58.8%) and Madhuwanti (73.8%). Sarod, Santoor, and Flute show strong performance with high accuracies across most ragas. Some exceptions include Miya Ki Todi and Puriya Kalyan, where the Sarod accuracy dips to 65% and 80.3% respectively. The collective count accuracy for the YAMNet model is slightly lower compared to VGGish, at 93% overall.

Top-performing Ragas as Raga Ahir Bhairav and Bhimpalas show very high accuracy across all instruments in YAMNet, with accuracies close to or at 100%.

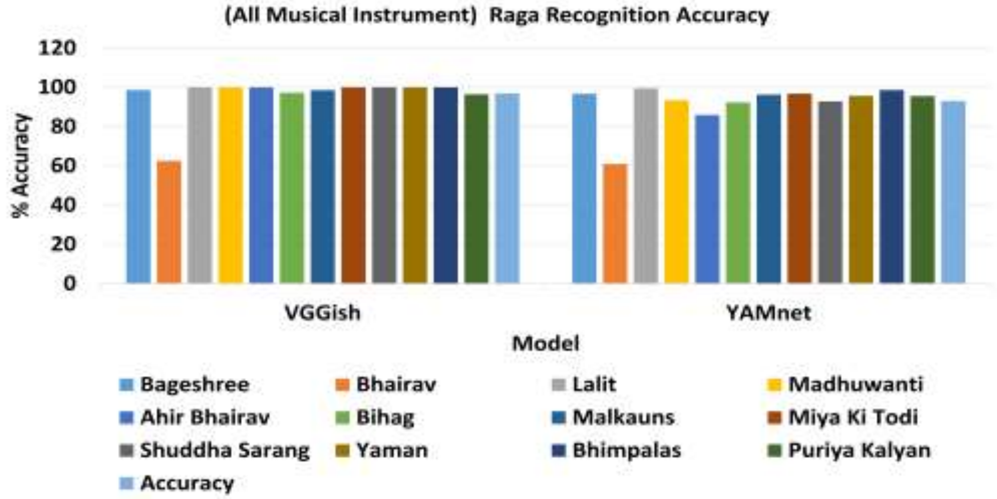


Fig 5.3 Raga recognition using Combined dataset

Model Comparisons:

Overall Accuracy: The VGGish model outperforms YAMNet slightly in terms of collective accuracy (96.88% vs. 93%), making it a better performer for Raga identification across instruments.

Instrument-wise Performance:

For Sitar, both models show some variation, but VGGish performs better, especially with Bageshree and Malkauns.

For Sarod, YAMNet performs better overall, but specific ragas such as Miya Ki Todi present challenges for both models.

Santoor performance is strong for both models, though YAMNet shows a slightly higher variance in specific ragas like Shuddha Sarang.

Flute performance is nearly flawless for both models, with only a few dips in YAMNet's recognition.

Insights:

Both models perform very well on Raga identification tasks, with VGGish having a slight edge in overall accuracy. Some ragas like Bageshree and Miya Ki Todi seem to be more difficult for both models to classify correctly using certain instruments. The Sarod and Flute generally yield higher accuracy for YAMNet, while Sitar and Santoor show higher accuracy with VGGish. This comparison helps to highlight that, depending on the specific instrument and the raga, different models may perform better. Both models can be used in Raga recognition, but VGGish provides a more stable and high-performing solution across different ragas and instruments.

6. Conclusion

In conclusion, this research paper delves into the significance of ragas as a foundational framework in Indian classical music, applicable to both Hindustani Classical and Carnatic traditions. This paper presents a technique for identifying Hindustani classical ragas using raw audio spectrograms and transfer learning using pre-trained VGGish and YAMnet models. The outcomes demonstrate the efficacy of these methods, attaining a remarkable 96.88% total testing accuracy with VGGish and 93.3% with YAMnet on a varied dataset of 12 ragas performed on four distinct musical instruments. These results highlight how versatile and adaptable the suggested methodologies are in the field of audio analysis, offering insightful information on teaching-learning processes, content-based filtering, music therapy, and retrieving music-related information. Overall, this study uses cutting-edge computational techniques to advance our knowledge of and ability to recognize ragas in Indian classical music.

By increasing database, recommendation system can be developed as per requirement of music therapist in future.

Disclosure Statement

Conflict of Interest

The authors confirm that there is no conflict of interest to declare for this publication.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgments

The researcher acknowledges their gratitude to Mr. Deepak Desai, Sitarist for the preparation of the database.

References

1. Dani´Elou, The Rāgas of Northern Indian Music. Munshiram Manoharlal Publishers, New Delhi, 2010.
2. T. Viswanathan and M. H. Allen, Music in South India: The Karāak Concert Tradition and Beyond. Oxford University Press, 2004.
3. S. Manjabhat, S. Koolagudi, K. Rao, and P. Ramteke, “Raga and tonic identification in Carnatic music,” Journal of New Music Research, vol. 46, pp. 1–17, 05 2017.
4. S. Banerjee, M. K. Rath, U. C. De and R. Sanyal, "Classification of Thaats in Hindustani Classical Music using Supervised Learning," 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2022, pp. 1-6, doi: 10.1109/GCAT55367.2022.9971937. keywords: {Supervised learning;Real-time systems;Hindustani Classical Music;Thaat classification;Spectrogram;Supervised Learning;Frequency domain analysis},
5. T. Krishna and V. Ishwar, “Carnatic music: Svara, gamaka, motif and raga identity,” in Serra X, Rao P, Murthy H, Bozkurt B, editors. Proceedings of the 2nd CompMusic Workshop; 2012 Jul 12-13; Istanbul, Turkey. Barcelona: Universitat Pompeu Fabra; 2012. Universitat Pompeu

Fabra, 2012.

6. R. Sarkar, S. Naskar, and S. Saha, \Raga identi_cation from hindustani classical
7. music signal using compositional properties," 09 2017.
8. Shah, Devansh & Jagtap, Nikhil & Talekar, Prathmesh & Gawande, Kiran. (2021). Raga Recognition in Indian Classical Music Using Deep Learning. 10.1007/978-3-030-72914-1_17.
9. A. Singha, N. R. Rajalakshmi, J. Arun Pandian and S. Saravanan, "Deep Learning-Based Classification of Indian Classical Music Based on Raga," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-7, doi: 10.1109/ISCON57294.2023.10111985.
10. Sharma, Akhilesh Kumar, Gaurav Aggarwal, Sachit Bhardwaj, Prasun Chakrabarti, Tulika Chakrabarti, Jemal H. Abawajy, Siddhartha Bhattacharyya, Richa Mishra, Anirban Das, and Hairulnizam Mahdin. “Classification of Indian Classical Music With Time-Series Matching Deep Learning Approach.” IEEE Access 9 (2021): 102041–52. doi:10.1109/ACCESS.2021.3093911.
11. Chakraborty, Sayanti & De, Debashis. (2012). Object-oriented classification and pattern recognition of Indian Classical Ragas. 2012 1st International Conference on Recent Advances in Information Technology, RAIT-2012. 505-510. 10.1109/RAIT.2012.6194630.
12. Gulati, Sankalp & Serra, Joan & Ishwar, Vignesh & Şentürk, Sertan & Serra, Xavier. (2016). Phrase-based rĀga recognition using vector space modeling. 66-70. 10.1109/ICASSP.2016.7471638.
13. Prof. Prasad Reddy P.V.G.D, Tarakenotea B Rao, Dr. K R Sudha, and Hari CH.V.M.K. Article: K-Nearest Neighbour and Earth Mover Distance for Raaga Recognition. International Journal of Computer Applications 33(5):30-38, November 2011. 10.5120/4017-5705
14. Schroeder, M. (1968). Period Histogram and Product Spectrum: New Methods for Fundamental-Frequency Measurement. The Journal of the Acoustical Society of America. 43. 829-34. 10.1121/1.1910902.
15. Z. Ren, K. Qian, Z. Zhang, V. Pandit, A. Baird, and B. Schuller, "Deep Scalogram Representations for Acoustic Scene Classification," in IEEE/CAA Journal of Automatica Sinica, vol. 5, no. 3, pp. 662-669, May 2018, doi: 10.1109/JAS.2018.7511066.
16. S. Banerjee, M. K. Rath, U. C. De and R. Sanyal, "Classification of Thaats in Hindustani Classical Music using Supervised Learning," 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2022, pp. 1-6, doi: 10.1109/GCAT55367.2022.9971937. keywords: {Supervised learning;Real-time systems;Hindustani Classical Music;Thaat classification;Spectrogram;Supervised Learning;Frequency domain analysis},
17. S. Banerjee, G. Hota, R. Sanyal and M. K. Rath, "Two Step Recognition of Raags in Hindustani Classical Music Using Supervised Deep Learning," 2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS), Bangalore, India, 2022, pp. 1-5, doi: 10.1109/ICDDS56399.2022.10037397. keywords: {Industries;Deep learning;Plagiarism;Detectors;Bandwidth;Raag Classification;Hindustani Classical Music;MEL Spectrogram;Supervised Learning;Deep Learning},
18. P. Rao et al., “Audio Metadata Extraction: The Case For Hindustani Classical Music” Department Of Electrical Engineering, Indian Institute Of Technology Bombay, Proc. of SPCOM 2012.
19. P. Kirthika, et al., “Frequency Based Audio Feature Extraction For Raga Based Musical Information Retrieval Using LPC and LSI”, Journal of Theoretical and Applied Information Technology, Vol. 69(3), Pg. No.