12-1-2022

Towards building a Deep Learning based Automated Indian Classical Music Tutor for the Masses

Vishnu S. Pendyala
San Jose State University, vishnu.pendyala@sjsu.edu

Nupur Yadav
San Jose State University

Chetan Kulkarni
San Jose State University

Lokesh Vadlamudi
San Jose State University

Recommended Citation

Vishnu S. Pendyala, Nupur Yadav, Chetan Kulkarni, and Lokesh Vadlamudi. "Towards building a Deep Learning based Automated Indian Classical Music Tutor for the Masses" *Systems and Soft Computing* (2022). https://doi.org/10.1016/j.sasc.2022.200042

4 (2022) 200042

Contents lists available at ScienceDirect

Systems and Soft Computing

journal homepage: www.journals.elsevier.com/soft-computing-letters

Towards building a Deep Learning based Automated Indian Classical Music Tutor for the Masses

Vishnu S. Pendyala [1,*], Nupur Yadav, Chetan Kulkarni, Lokesh Vadlamudi

San Jose State University, 1 Washington Sq, San Jose, CA 95192, USA

ARTICLE INFO

Keywords:
Machine Learning
Indian Classical Music
Deep Learning
Long Short-term Memory
Cloud Computing
DevOps

ABSTRACT

Music can play an important role in the well-being of the world. Indian classical music is unique in its requirement for rigorous, disciplined, expert-led training that typically goes on for years before the learner can reach a reasonable level of performance. This keeps many, including the first author of this paper, away from mastering the skill. The problem is particularly compounded in rural areas, where the available expertise may be limited and prohibitively expensive, but the interest in learning classical music still prevails, nevertheless. Machine Learning has been complementing, enhancing, and replacing many white-collar jobs and we believe it can help with this problem as well. This paper describes efforts at using Machine Learning techniques, particularly, Long Short-Term Memory for building a system that is a step toward provisioning an Indian Classical Music Tutor for the masses. The system is deployed in the cloud using orchestrated containerization for potential worldwide access, load balancing, and other robust features.

Introduction

Music is the universal language of mankind, said the famous 19th century American poet and educator, Henry Wadsworth Longfellow. Making music tutoring accessible to the masses is a humanitarian cause. Many studies, including recent ones such as [1,2], and [3] have confirmed the positive effect of music on mental health. Literature also provides evidence that Machine Learning tools have been used to leverage this fact [4]. Indian classical music, which dates back to the Vedic times thousands of years ago, is particularly said to be effective for quite a few medical conditions [5,6,7]. However, powerful music is a result of several years of rigorous practice under the tutelage of an established master musician. The expertise of a master musician is not easily available or is prohibitively expensive in a substantial number of scenarios, particularly in the rural areas of India. This work attempts to alleviate the problem and make training in Indian classical music more accessible.

Audio processing using Machine Learning such as for speech recognition and acoustic scene classification has reached near-human performance [8]. It is a matter of time that Machine Learning will take over several audio tasks, one of which, we believe will be music tutoring. The first author, along with two of his students recently published their work to convert Indian classical music from one melodic framework to another [9]. A particular style of music, known as Hindustani classical music, is a multi-faceted and ancient form of musical art that has its roots in the Indian subcontinent. The key element in Hindustani classical music is called raga, sometimes also called "raag." A raga is a musical theme or melodic framework. It can be thought of as a scale created by choosing a specific set of notes from within an octave where different combinations of notes evoke different moods and inspire different feelings.

Hindustani classical music is gaining popularity all around the world, especially in the west [10]. Many people are desirous of learning Hindustani classical music for personal passion or interest. However, Hindustani classical music is substantially complicated. and requires constant guidance. Many music themes or ragas can be based on the same scale with several improvisations which are difficult to distinguish, and there are not many musicians who have mastered it. These improvisational patterns are not even concretely defined in any format and are just passed through a fading oral mentor-student tradition. The project aims to build a music tutoring website to enable real-time recording and uploading of songs that can help the user in getting immediate feedback on the musical notes they are singing. In this project, deep learning is used to identify melodic frameworks based on audio features such as

* Corresponding author.
E-mail address: vishnu.pendyala@sjsu.edu (V.S. Pendyala).
[1] URL: https://www.sjsu.edu/people/vishnu.pendyala

https://doi.org/10.1016/j.sasc.2022.200042
Received 20 January 2022; Received in revised form 9 May 2022; Accepted 26 May 2022
Available online 9 June 2022

scale, pitch, tone and provide feedback and suggestions to the users. The website can be accessed from anywhere, anytime, and free of cost to encourage people to follow their musical passion.

Hindustani Classical music is an ancient musical form predominant in northern parts of India, Pakistan, and Bangladesh. It focuses mainly on melodic development. A music composition in the Hindustani classical style should typically confirm to a "raga." The "raga" is a musical theme, or a melodic framework created by choosing a specific set of notes called "swaras", on a scale, ordered in melodies with musical motifs. It has characteristic intervals, rhythms, and embellishments that are used as a basis for improvisation. Experts [11,12] define raga technically as a collection of melodic atoms and a technique to develop them. Hindustani classical music focuses more on the space between the notes than the notes themselves. A musician playing a melodic framework may use the same notes but may improvise or emphasize certain degrees of the scale evoking a mood that is unique to the melodic framework.

Fig. 1 shows a sample of the scale for a specific melodic framework or raga, named "Bhairav." Each melodic framework is constructed from five or more musical notes and can be written on a scale. Many such melodic frameworks are based on the same scale with several improvisations such as timing between the notes, timing of the attack of each note, and its sustain. These improvisational patterns are difficult to determine and are not properly documented in any form and known to only a few musicians who have mastered it. This melodic variation and inconsistent temporal spacing make the identification of the specific framework (raga) a substantially challenging problem. This paper proposes a system for the identification of melodic frameworks that a music composition adheres to. Such identification helps the users to recognize the melodic framework to which their rendering confirms to and practice it. The users get immediate feedback by way of a confidence score indicating how close they are to the melodic framework. This work aims to create an ecosystem where Hindustani Classical music can thrive and generate interest globally. The vision of the web as the ubiquitous computer [13] makes this further possible.

In this paper, a deep learning solution for identification of the melodic framework (raga) using a bidirectional Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) architecture is proposed. Initially, the system generates Mel-frequency cepstrum coefficients (MFCC) representations of the input audio data which are then fed to the BiLSTM RNN architecture for the identification of the melodic framework. Further, the website enables users to either upload songs or do live recordings and identify the type of the melodic framework present in the song along with the generated confidence score. The system also gives recommendations to the users for the type of the melodic framework they are interested in.

Literature survey

The literature survey provides an overview of Music Information Retrieval (MIR), various pre-processing techniques for extracting audio features, deep learning techniques like convolutional and recurrent neural networks for MIR tasks, and state-of-the-art analysis of melodic frameworks (ragas).

Music Information Retrieval

Music Information Retrieval (MIR) is an interdisciplinary research field that focuses on extracting information from music and its applications [14]. It has many real-world applications such as recommender systems, instrument recognition, automatic categorization, automatic music transcription, and music generation. Besides these potential applications, extracting the raga from Hindustani Classical music is quite challenging because of the inconsistent temporal spacing between notes and the intense variation in its rhythmic patterns [15].

Audio Feature Extraction

Feature extraction is the most important part of the machine learning process. The performance of any machine learning model depends on the features it is trained and tested on. There are several techniques to extract features for audio data such as temporal domain, frequency domain, cepstral domain, wavelet domain, and time-frequency domain features [16]. The important music-related features that are often used for classification, prediction, and recommendation algorithms include zero-crossing rate, spectral centroid, spectral roll-off, Mel-Frequency Cepstral Coefficients (MFCCs), and chroma frequencies.

The zero-crossing rate defines the rate at which the signal changes from positive to negative or back. It has many applications such as music discrimination, music genre classification [17], etc. The Spectral centroid indicates the location of the center of mass of the spectrum. It is also called the brightness feature of a sound as it describes the brightness of the sound. This feature is highly used as a measure of the timbre of the music, music classification, and music mood classification [18]. Spectral roll-off is the measure of the shape of the signal. It represents the frequency such that 95% of the signal energy is contained below this frequency. Its applications include musical instrument classification, music classification [19], music genre classification [20–23], and audio-based surveillance systems [24].

The Mel frequency cepstral coefficients (MFCCs) of a signal are derived from the cepstral representation of an audio clip and is a key feature in any audio signal processing. They represent a small set of about 10-20 features that concisely describe the overall shape of a spectral envelope. MFCCs can mimic the human voice very closely as the frequency bands are equally spaced on a mel-scale. MFCCs have wide variety of applications such as music information retrieval [25], speech enhancement [26], speech recognition [27], music genre classification [28], vowel detection [29] etc.

Another important feature for representing music audio is by using chroma features or chromagrams. In a chromagram, the entire spectrum is projected into 12 bins which represents the 12 distinct chroma (or semitones) of the musical octave. This project uses MFCCs and chroma features for representing the music data.

Deep Learning for the Analysis of Melodic Frameworks

Many approaches have been used in the past for recognition of the melodic frameworks. Most recent approaches involve extracting features such as spectrograms, MFCCs, chromagrams from music data and then applying a clustering algorithm or a classification algorithm for extracting raga information [30–35]. However, the existing systems cannot detect raga in real-time as they require the entire audio data to be

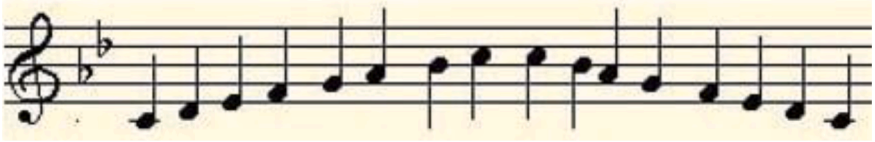

Fig. 1. Raga Bhairav scale.

processed first and then generate a raga prediction. Having a raga evaluation system that can predict raga in real-time can help users practice well because of live feedback. It can be more like a tutoring service.

The other approaches mentioned in [34,36,37] require manual extraction of features like pitch histograms that are unable to capture music features such as note transitions essential for raga analysis. Also, hand-crafting features can be time-consuming and tedious especially when there are a huge number of classes as in this case. Another point to note is that the existing approaches for classifying melodic frameworks of the Hindustani Classical music cannot be applied to other Audio Information retrieval (AIR) tasks as they make assumptions about the note emphasis [31] and pitch distribution [34,36–38] for music classification and make them limited in scope.

Some recent works like [39,40] have used deep learning techniques like convolutional neural networks and recurrent neural networks for feature extraction and processing raga information from the music data. However, these works also require human intervention for extracting features along with some other additional inputs like composition annotation. Also, these cannot inspect music features like overall pitch distributions and note ornamentations separately which are essential in music data for raga classification.

Music Tutors

Massive Open Online Courses (MOOC) with automated assessment and leveraging Human-Computer Interaction (HCI) have been reviewed in [41]. The authors propose using Hidden Markov Model (HMM) for this purpose. However, studies such as [42] proved that RNNs outperform HMMs and will replace HMMs. A type of RNN called LSTM is used for the work described in this paper. There are also attempts such as [43], which talk about a "Circuit diagram of machine learning" for teaching music. A few ideas have been floated in [44] for designing a distance teaching system using machine learning for music dance courses. There seem to have been attempts such as [45] to design systems to teach piano using machine learning, but the publication has been withdrawn subsequently.

Contribution of this paper

Making music tutoring, particularly that of ancient, powerful, and labyrinthine music that can help with certain medical and mental health conditions is an important milestone in social innovation. Based on the literature survey, to the best of our knowledge, this work is unique in proposing an architecture based on deep learning and orchestrated containerization in a cloud environment that is a step toward making Indian classical music tutoring globally accessible inexpensively.

System Architecture

A bidirectional LSTM, an RNN architecture is used for the problem of identification of melodic frameworks. As pointed out in the literature survey, this is a popular architecture used for sequence classification and sequence to sequence learning tasks and is fast replacing Hidden Markov Models (HMM). This neural network contains three LSTM layers followed by a simple RNN layer, which are then followed by dense layers. This model trains itself on features from MFCCs to form output in the form of a five-dimensional array representing the probability scores towards the five different classes that were chosen. The class with the highest score is the raga label for that song. The Softmax activation function was used in the last layer to generate the probability scores and sparse_categorical_cross-entropy as the loss function. The architectural block diagram of the model is shown in Fig. 2.

Tools and Environments

Python was used as the programming language to develop this project because of its simplicity, consistency, platform independence, and access to great libraries and frameworks for machine learning (ML) and artificial intelligence (AI). The version used is Python 3.7.10. TensorFlow 2.4.1 was used as the model backend. The TensorFlow Extended (TFX) is an end-to-end platform for deploying production machine learning pipelines. In this project, the pusher component of the TensorFlow Extended platform known as the "TensorFlow serving" was used to serve the machine learning model. By default, TensorFlow serving provides the REST endpoint for users to get their predictions. The important machine learning libraries used are Keras, scikit-learn, and Librosa. Google Colab Pro with GPUs was used for building the AI model and Django a python-based open-source web framework for developing the website.

Cloud and containerization infrastructure

Software build, integration, and release engineering have evolved substantially over time into robust DevOps [46] using cloud and containerization infrastructure., which takes care of load balancing to meet elastic demand from worldwide clientele.To handle hundreds or even millions of requests at the same time, we, therefore, leverage DevOps in a good cloud infrastructure. Google Cloud Platform (GCP) services were used for hosting the model and the web application.

Docker was used to create two separate images one for the Django app and another for the model which is being served using TensorFlow serving in TFX. Both the images are then deployed on a Kubernetes engine of GCP. One of the deployments contains three replica pods each serving the machine learning model. The other deployment hosts the Django web application with replicas for uninterrupted service. Fig. 3 shows the model and web deployment architecture. It shows the user accessing web application service which in turn uses model service to fetch predictions.

Docker was used for containerization. Docker is a Platform-as-a-service (PaaS) product that uses OS-level virtualization to deliver software in packages called containers. Docker enables developers to easily pack, ship, and run any application as a lightweight, portable, self-sufficient container, which can run virtually anywhere. Docker containers are easy to deploy in a cloud environment. The two main docker containers in this project are 1) Django web application Container and 2) Raga Prediction Model Container.

The Django application image was created with the help of a base python image. The docker-compose file was configured including steps for copying Django content into the root directory of the base python image and starting the server at a specific port. The model container was created using the base TensorFlow/serving image. Fig. 4 shows the steps to create a custom TensorFlow serving image for the model. The model protocol buffer files from the generated model are copied into the server

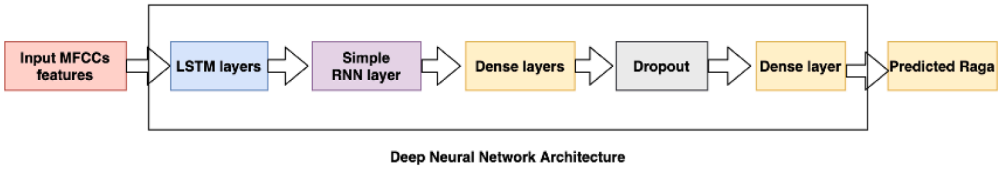

Fig. 2. Architectural block diagram for deep neural network model.

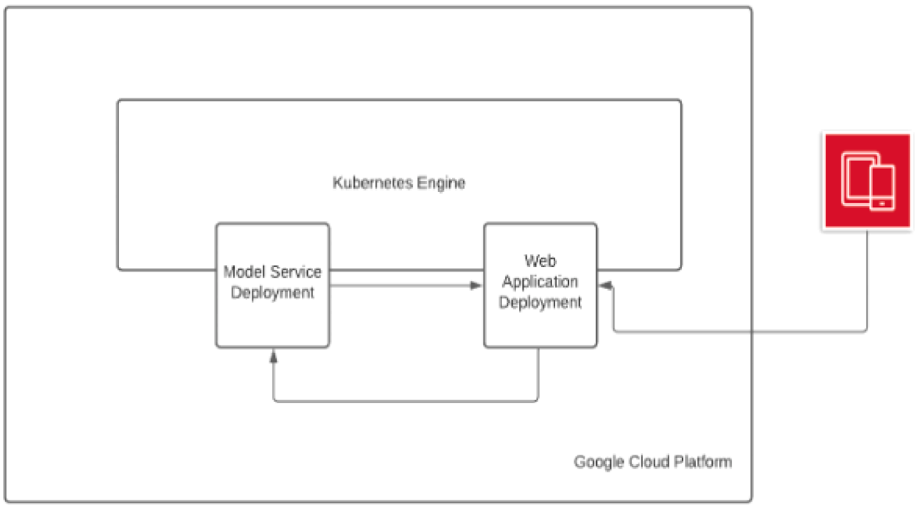

Fig. 3. Model and web application deployment architecture.

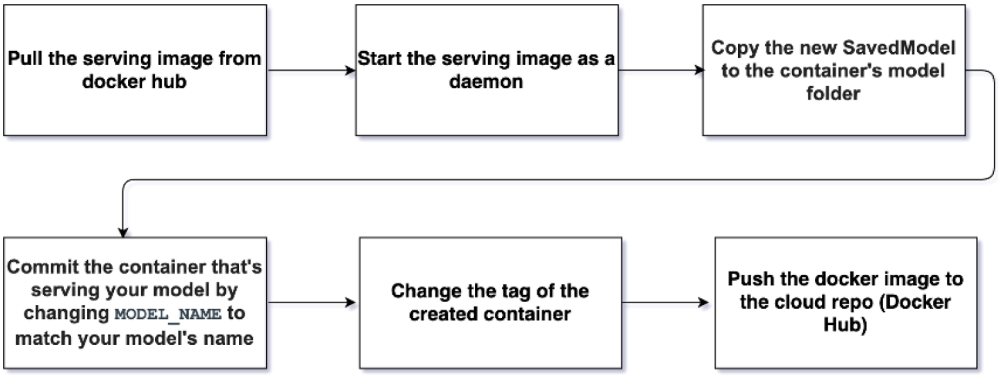

Fig. 4. Steps to create a custom TensorFlow serving image.

image. Later the custom model docker is pushed to cloud storage as discussed in the TFX section above. The cloud storage for the docker containers is Docker Hub. GCP pulls the images from this storage for creating deployments on the cluster.

Kubernetes was used for container orchestration. Kubernetes is an open-source container orchestration platform that enables the operation of an elastic web server framework for cloud applications. It offers portability, and faster, simpler deployment times. Both the Django web application Container and the Raga Prediction Model Container were deployed on the Kubernetes cluster of GCP. Fig. 5 shows a schematic of the virtual node machines in the Kubernetes cluster. Each node can contain many pods and each pod can contain one or more containers. Generally, it is advisable to run a single container inside a pod for better stability. There are three identical pods in total, and load is distributed between the three with the help of a load balancer service.

Preparation for Experiments

As in any machine learning application, we collect a dataset, extract features where applicable, split the dataset into training and validation sets, and generate a model based on the dataset.

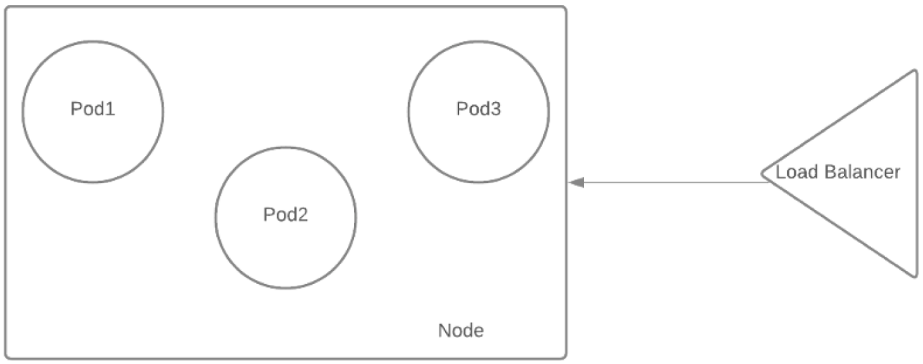

Fig. 5. Internals of virtual node machines in Kubernetes cluster.

Data Collection

Dataset collection is the most crucial part of any machine learning task. Initially, despite trying hard, we did not find any readily available, relevant dataset for Hindustani Classical music online. A few Indian Classical music (ICM) datasets existed but not many datasets were publicly available for Hindustani Classical music (HCM) with predefined raga labels. It was therefore decided to create the dataset exclusively for this project. The dataset curation was done in two parts: (a) Personal collections from friends and family. (b) Downloading songs from Dunya (https://dunya.compmusic.upf.edu/) using the PyCompmusic library. Fig. 6 demonstrates the steps followed to download and prepare the dataset to be further used by the AI model.

A function was written to automate the process of downloading songs from Dunya using the PyCompmusic library which provides a wrapper to Dunya APIs. Initially, *compmusic.Dunya.Hindustani.get_recordings(recording_detail=True)* API was used to generate a JSON file containing a list of song Ids, the title of songs, and raga type. This JSON file was then converted to CSV for easy parsing. All the song ids in the CSV file are iterated through to download the corresponding mp3 songs into separate raga folders as per the raga type associated with it using the *compmusic.Dunya.Hindustani.download_mp3(recordingid, location)* API. The final dataset contains 25 full-length recordings per raga and features 5 different ragas, hence 125 recordings.

Data Preprocessing and Feature Extraction

The HCM dataset obtained was loaded into a Google Colab Pro notebook for preprocessing and feature extraction. The librosa library was used to extract audio features which are understandable to a machine learning algorithm. Each mp3 audio data was converted into 13 Mel-frequency cepstral coefficients (MFCCs) features and stored into a JSON file along with their corresponding raga label. The MFCCs are a powerful representation of an audio signal as it scales the frequency to match more closely what the human ear can hear.

Fig. 7 shows a sample of the extracted MFCC features with their corresponding label identifying the melodic framework. The features are essentially a numerical representation of the frequencies in the audio signal of the music snippet on what is called as the Mel scale. The scale is primarily a logarithmic conversion of the audio frequencies to make them better aligned with the human perception of audio. Fig. 8 shows the MFCC representation of an audio clip in the form of a heatmap. The x-axis represents the 13 MFCC features extracted over time and the y-axis is the frequency spectrum. The color indicates the strength of the audio signal.

Further, feature representations were generated using spectrogram and chromagram images to see if these features can help in building a better model for the raga identification problem. Fig. 9 shows the spectrogram and chromagram images for the raga bhairav sample audio file. The spectrogram, chromagram, and heatmaps are different visual representations of audio signals and are commonly used in audio analysis projects. A spectrogram is a 3-dimensional plot of the audio signal. The horizontal dimension or x-axis is time, y-axis is frequency and the third dimension represented by the color indicates the amplitude. The fine changes in the color across time and frequency domains captured in the spectrogram give substantial insights into the music clip. Chromagram identifies pitches that differ by an octave and is also a powerful approach to analyzing music snippets.

Methodology

Teaching Indian classical music is an excellent example of similarity-based learning. The teacher renders a melody, and the students imitate the teacher to be as like the teacher's rendering as possible. The similarity is a key element of machine learning. Recent studies [47] proved that every machine learning algorithm that uses gradient descent and that includes deep learning, is essentially a kernel machine, where the kernel function measures similarity between data points. It is quite appropriate that deep learning is used to induce similarity-based music learning. At the crux of the methodology is therefore deep learning and specifically LSTM, as detailed below.

Identification of the melodic framework (raga): Model Training

The machine learning model purposed to identify the raga of an uploaded music segment is trained on the audio features that were generated, as described in section 4. The MFCC features from the music files are used to train the bidirectional LSTM RNN model. As mentioned earlier, LSTM is a popular choice for sequential classification and sequence to sequence tasks. LSTM is a special kind of Recurrent Neural Network (RNN) that can consider the long-term dependencies among the features during training. This model contains a memory state which helps to remember or forget features like how music comprehension works in human beings.

Important features of the music which are needed for later reckoning are preserved by the memory state. This capability enables LSTM to work well with sequential data such as audio signals. The key aspect of LSTM is to split the hidden state into two. The first part is for long-term retention in a "memory cell" and the second part is for short-term. Some information about the audio features is selectively read, some remembered, some of it is forgotten or ignored, quite analogous to how music comprehension works in the human mind. The gate eqs. (1),(2),(3) and state eqs. (4),(5),(6) update equations are given below.

$$o_t = \sigma(W_o h_{t-1} + V_o x_t + b_o) \tag{1}$$

$$i_t = \sigma(W_i h_{t-1} + V_i x_t + b_i) \tag{2}$$

$$f_t = \sigma\left(W_f h + V_f x_t + b_f\right) \tag{3}$$

$$s_t^{\sim} = \sigma(W h_{t-1} + V x_t + b) \tag{4}$$

$$s_t = f_t \odot s_{t-1} + i_t \odot s^{\sim} \tag{5}$$

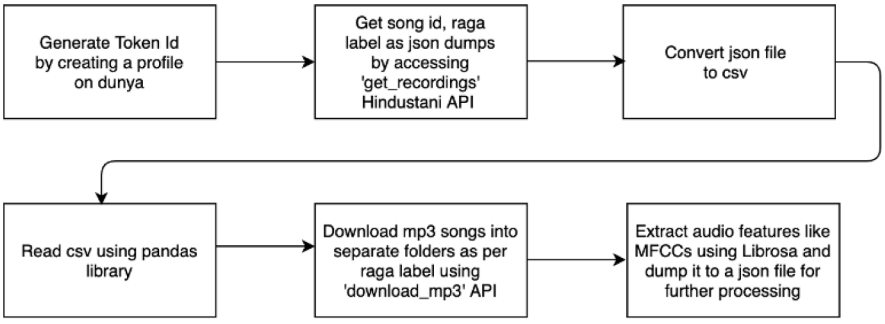

Fig. 6. Steps to download and prepare HCM dataset.

mfcc1	mfcc2	mfcc3	mfcc4	mfcc5	mfcc6	mfcc7	mfcc8	mfcc9	mfcc10	mfcc11	mfcc12	mfcc13	label
-338.281053	113.817955	-14.052405	8.551622	-5.788107	-5.635894	-10.244627	-8.675543	-10.974425	1.525304	-2.884718	7.884231	-0.295875	madhuvanti
-272.303322	119.241958	-11.244108	11.319374	-2.548124	-6.570148	-12.734617	-7.217122	-8.655407	1.841116	-0.884794	7.683995	1.280020	madhuvanti
-509.803622	73.050575	-8.811494	10.525128	-0.858952	-0.146839	-8.890942	-1.944142	-4.980757	2.031216	-2.907506	8.473813	-0.128429	madhuvanti
-260.629208	121.702862	-10.383890	10.917586	-1.730320	-4.178613	-13.270481	-6.671383	-0.754154	0.845042	-2.934038	6.622784	3.343194	madhuvanti
-696.569207	4.576745	-0.495340	0.662960	-0.007068	0.012834	-0.409690	-0.082544	-0.259814	0.133246	-0.169076	0.420270	-0.027064	madhuvanti
...	...	...	...	...	...	...	...	...	...	...	...	...	...
-198.717968	129.919135	-15.070532	15.539915	-15.808483	0.439417	0.440207	-9.561070	-7.697309	3.332678	-15.943490	-2.292459	-12.214308	multani
-227.857703	120.894675	-16.167212	16.042029	-16.922651	-7.386678	-4.247232	-12.899577	-10.193567	1.571645	-19.803650	-0.746795	-14.709724	multani
-187.867286	109.892690	-35.066026	8.101116	-9.711220	-13.607133	-10.112420	-7.702000	-2.879457	1.073179	-7.411411	2.659039	-1.413216	multani
-192.349212	123.895564	-16.105062	19.597401	-12.457826	-4.577696	-3.930122	-10.114332	-9.244178	-0.755781	-14.367605	-2.424859	-12.324200	multani
-162.931756	114.775892	-40.825779	-0.614991	-9.189457	-10.207901	-8.421380	-5.949722	-2.424465	1.347141	-9.525109	-1.475691	1.210340	multani

Fig. 7. Sample dataset after MFCC feature extraction from audio data.

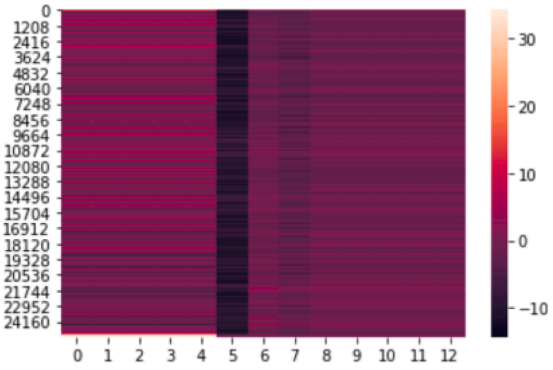

Fig. 8. MFCC representation of Raga Ahir Bhairav audio clip.

where

$$h_t = o_t \odot \sigma(s_t) \tag{6}$$

- s_t is the state of the RNN at timestep t x_t is the new information at timestep t
- o_t is the output gating vector at timestep t that decides how much of the state needs to be handed over to the next state
- i_t is the input gating vector at timestep t that implements selective read f_t is the forget gate vector that implements selective forget
- W_i,$W_{o,}$ W_f are the parameters that need to be learned for the input, output, and forget gates respectively to weigh the hidden state from the previous timestamp, t-1
- V_i,$V_{o,}$ V_f are the parameters that need to be learned for the input, output, and forget gates respectively to weigh the input at timestamp t
- b_i,$b_{o,}$ b_f are the respective biases
- σ is the sigmoid function
- h_t is the portion of the hidden state at timestep t that moves forward
- $\widetilde{s_t}$is the state before selective read

The usual train_test_split method from the Scikit-learn library is used to divide the dataset into train, test, and validation set, s. As is typical, 25% of the dataset was allocated to the test split and 20% of the train data to the validation split. The BiLSTM RNN model was then trained on 75% of the dataset for 500 epochs. As described in the next subsection, Adam optimizer with 0.001 learning rate and sparse_categorical_cross-entropy was used as the loss function for training the model.

For the alternative experiments, a deep Convolutional Neural Network model was also trained on the spectrogram and chromagram images extracted from the mp3 audio data along with a pre-trained ResNet [48] model. The idea here is to compute the accuracy from the chromagram, and spectrogram images generated from the music. The pre-trained ResNet model was also used because, in the image classification tasks, pre-trained models yielded better results in some cases. In the process of training using the ResNet model, one only needs to train the top layers. Pre-trained weights work to our advantage because of transfer learning.

Loss Function and Optimizer

The choice of loss function and optimizer depends on the problem domain and model performance. The *sparse_categorical_cross-entropy* loss function, which is a categorical cross-entropy with integer targets was used for the project. Categorical Cross-Entropy (CCE) is particularly suited for the multi-class classification problem, such as the one we have at hand. It must be noted that the targets are also in integer form. The target is sparse because the representation requires much less space than one-hot encoding. For example, a batch with b targets and k classes

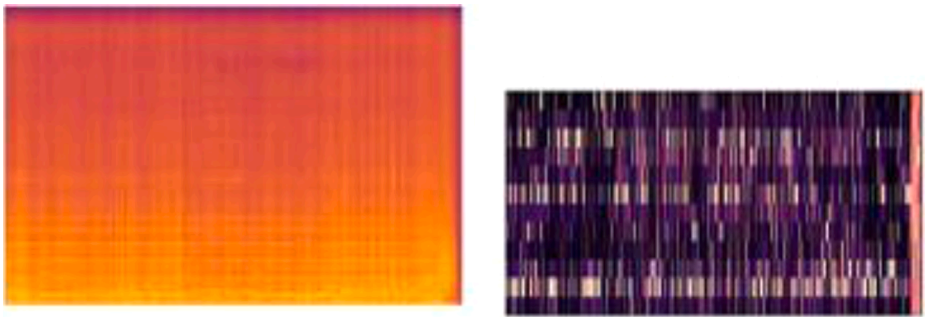

Fig. 9. (a) Spectrogram and (b) Chromagram representation of Raga Bhairav audio clip.

needs b * k space to be represented in one-hot, whereas a batch with b targets and k classes needs b space to be represented in integer form. Equation 7 gives the expression for computing CCE.

$$CCE(t,s) = -\Sigma t_{i,c} log(s_{i,c}) \quad (7)$$

where $t_{i,c,}$ and $s_{i,c}$ are the ground truth and the predicted integer category for each class c in C.

The optimizer used is Adam with a learning rate of 0.001. Adam optimization is a stochastic gradient descent method that is based on the adaptive estimation of the first-order and second-order moments. It combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.

Deployment

Model deployment is one of the crucial stages in a machine learning application. To serve thousands of users, the model needs to be highly scalable and optimized. The system therefore was deployed using the cloud and containerization infrastructure as described in section 3.2. A substantial amount of computing happens in the cloud [49] these days, anyway. The deployment allows a few handfuls or many millions of users to utilize the system seamlessly and scales elastically. The system, in addition to identifying the melodic framework about the music that is uploaded, also provides feedback as to the proximity to the raga and plays similar tunes confirming to the same raga, so that the user can learn from them. A simple schematic of the use-model is given in Fig. 10.

Experiments and results

First, experiments were performed to come up with the best LSTM RNN model configurations for the raga identification task using the MFCCs features as input. Later, experiments were also run with other audio features such as spectrograms and chromagrams. Since these were image features, CNN was used in addition to the pre-trained models such as ResNet for training. The number of raga classes remained fixed at five across all experiments. The model performance was evaluated in terms of test accuracy achieved. Table 1 shows the experiments performed and the accuracies achieved.

Since the dataset was already balanced, the need for kappa statistic or other metrics was not necessitated. Train and validation accuracies were also considered during the experiments. TensorBoard integration was used for observing the training process and tracking metrics like accuracy and loss.

As the above experiments show Deep BiLSTM RNN architecture achieved the best accuracy of 78%. This architecture was therefore used as the final model for raga prediction and further deployment.

Table 1
Results from the experiments.

	Experiment	Accuracy (%)
1.	Deep BiLSTM RNN with MFCC input	78%
2.	ResNet with chromagram input	65%
3.	Deep CNN with chromagram input	62%
4.	BiLSTM with MFCC input	52%
5.	Deep CNN with spectrogram input	55%
6.	ResNet with spectrogram input	60%
7.	Deep LSTM with MFCC input	59%

Discussion

Overall, for the limited HCM dataset that could be collected, the model performed reasonably well. Just 25 recordings per category of the melodic framework are considerably small to train machine learning models. We could still achieve an acceptable peak accuracy of 78%. Another limiting aspect is the use of ResNet for transfer learning the chromagrams and spectrograms.

Conclusion and future directions

In this work, a substantial effort towards building a music tutor for learning Indian classical music was proposed. A deep learning solution was used for the problem of identifying melodic frameworks also called "ragas". The model was deployed in a containerized and orchestrated environment on the cloud for seamless load balancing and worldwide access. Currently, the system provides limited guidance to the learner by playing similar music that confirms to the specific raga. Through various experiments and validations, it was found that the best deep learning solution uses three LSTM layers (two BiLSTM and one LSTM) followed by a simple RNN layer which is further followed by three dense layers. At this time, the system can distinguish between five Hindustani ragas with an accuracy of 78%. The system is adaptable and can be used for other Audio Information Retrieval (AIR) tasks as well.

The work is only tested on the Hindustani Classical Music (HCM) dataset. As a future direction, it can be enhanced to analyze different styles of music and instruments by training on a more diverse dataset. The model performance can also be enhanced by using a larger dataset of music clips belonging to diverse ragas. Another future direction is to identify the raga in real-time, as the user is singing, and provide immediate feedback. The system is only a proof-of-concept and has not been tested on real users yet. In its current implementation, the system should be seen as a substantial step toward building a tutor with considerable scope for improvements before it can be opened to world audiences and users.

Declaration of Competing Interest

The authors declare that they have no known competing financial

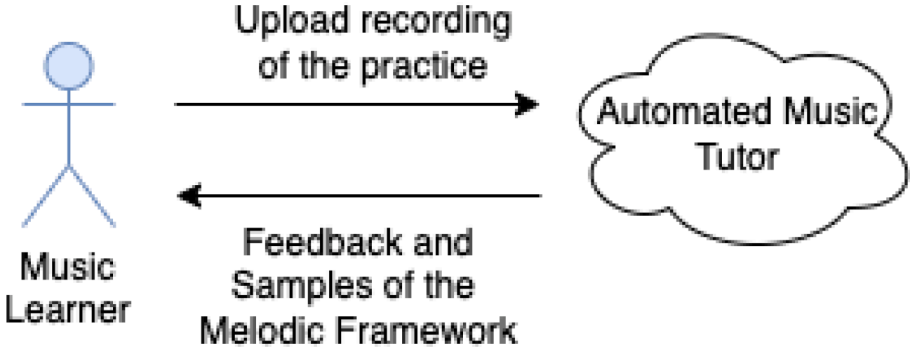

Fig. 10. Current Use Model.

interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge the help of Dunya, and other providers of the dataset used for the experiments. The authors are also grateful to San Jose State University under the aegis of which this project was executed. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Kim Hyun-Sil, Jin-Suk Kang, Effect of a group music intervention on cognitive function and mental health outcomes among nursing home residents: A randomized controlled pilot study, Geriatr. Nurs. 42 (3) (2021) 650–656, no.
[2] Sema İçel, Ceyda Başoğul, Effects of progressive muscle relaxation training with music therapy on sleep and anger of patients at Community Mental Health Center, Complement. Ther. Clin. Pract. 43 (2021), 101338.
[3] R.N. Gurbuz-Dogan, A. Ali, B. Candy, M. King, The effectiveness of Sufi music for mental health outcomes. A systematic review and meta-analysis of 21 randomized trials, Complement Ther. Med. (2021), 102664.
[4] Jessica Sharmin Rahman, Tom Gedeon, Sabrina Caldwell, Richard Jones, Zi Jin, Towards effective music therapy for mental health care using machine learning tools: Human affective reasoning and music genres, J. Artif. Intell. Soft Comput. Res. 11 (1) (2021) 5–20.
[5] Sravanti L. Sanivarapu, India's rich musical heritage has a lot to offer to modern psychiatry, Indian J. Psychiatry 57 (2) (2015) 210, no.
[6] Shantala. Hegde, Music therapy for mental disorder and mental health: the untapped potential of Indian classical music, BJPsych Int. 14 (2) (2017) 31–33, no.
[7] Rajiv Balan, Sandeep B. Bavdekar, Sandhya Jadhav, Can Indian classical instrumental music reduce pain felt during venepuncture? Indian J. Pediatr. 76 (5) (2009) 469–473, no.
[8] Diogo Moreira, Ana Paula Furtado, Sidney Nogueira, Testing acoustic scene classifiers using Metamorphic Relations, in: 2020 IEEE International Conference On Artificial Intelligence Testing (AITest), 2020, pp. 47–54. IEEE.
[9] Rohan Surana, Aakash Varshney, Vishnu Pendyala, Deep Learning for Conversions Between Melodic Frameworks of Indian Classical Music, in: Proceedings of Second International Conference on Advances in Computer Engineering and Communication Systems, 2022, pp. 1–12. Springer, Singapore.
[10] Peter. Lavezzoli, The dawn of Indian music in the West, A&C Black, 2006.
[11] Arvindh. Krishnaswamy, Melodic Atoms for Transcribing Carnatic Music. ISMIR, 2004, p. 2004, 5th.
[12] Arvindh. Krishnaswamy, Multi-dimensional musical atoms in South-Indian classical music, in: Proc. of International Conference on Music Perception and Cognition, 2004.
[13] Vishnu S. Pendyala, Simon SY Shim, The Web as the ubiquitous computer, Computer 42 (09) (2009) 90–92, no.
[14] J.Stephen Downie, Music information retrieval, Annual Rev. Info. Sci. Technolo. 37 (1) (2003) 295–340, no.
[15] RS. Gottlieb, The major traditions of North Indian tabla drumming: a survey presentation based on performances by India's leading artists: illustrated with recordings and transcriptions of the performances, Musikverlag E. Katzbichler 1 (1977). Vol.
[16] Garima Sharma, Kartikeyan Umapathy, Sridhar Krishnan, Trends in audio signal feature extraction methods, Appl. Acoust. 158 (2020), 107020.
[17] Peter Ahrendt, Anders Meng, Jan Larsen, Decision time horizon for music genre classification using short time features, in: 2004 12th European Signal Processing Conference, 2004, pp. 1293–1296. IEEE.
[18] Giulio Agostini, Maurizio Longari, Emanuele Pollastri, Musical instrument timbres classification with spectral features, EURASIP J. Adv. Signal Process. (1) (2003) 1–10, 2003no.
[19] Abdullah I. Al-Shoshan, Speech and music classification and separation: a review, Journal of King Saud University-Engineering Sciences 19 (1) (2006) 95–132, no.
[20] George Tzanetakis, Perry Cook, Musical genre classification of audio signals, in: IEEE Transactions on speech and audio processing 10, 2002, pp. 293–302, no.
[21] Lie Lu, Dan Liu, Hong-Jiang Zhang, Automatic mood detection and tracking of music audio signals, IEEE Transactions on audio, speech, and language processing 14 (1) (2005) 5–18, no.
[22] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, Balázs Kégl, Aggregate features and a da b oost for music classification, Mach. Learn. 65 (2–3) (2006) 473–484, no.
[23] Tao Li, Mitsunori Ogihara, Qi Li, A comparative study on content-based music genre classification, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003, pp. 282–289.
[24] Asma Rabaoui, Manuel Davy, Stéphane Rossignol, Noureddine Ellouze, Using one-class SVMs and wavelets for audio surveillance, IEEE Trans. Inf. Forensics Secur. 3 (4) (2008) 763–775, no.
[25] Ning Hu, Roger B. Dannenberg, George Tzanetakis, Polyphonic audio matching and alignment for music retrieval, in: 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No. 03TH8684), 2003, pp. 185–188. IEEE.
[26] Alexander Krueger, Reinhold Haeb-Umbach, Model-based feature enhancement for reverberant speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 18 (7) (2010) 1692–1707, no.
[27] Steven Davis, Paul Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE transactions on acoustics, speech, and signal processing 28 (4) (1980) 357–366, no.
[28] Meinard. Müller, Information retrieval for music and motion, 2, Springer, Heidelberg, 2007. Vol.
[29] Yunus Korkmaz, Aytuğ Boyacı, Türker Tuncer, Turkish vowel classification based on acoustical and decompositional features optimized by Genetic Algorithm, Appl. Acoust. 154 (2019) 28–35.
[30] Pranay Dighe, Parul Agrawal, Harish Karnick, Siddartha Thota, Bhiksha Raj, Scale independent raga identification using chromagram patterns and swara based features, in: 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2013, pp. 1–4. IEEE.
[31] P. Kirthika, Rajan Chattamvelli, A review of raga based music classification and music information retrieval (MIR), in: 2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA), 2012, pp. 1–5. IEEE.
[32] Gaurav Pandey, Chaitanya Mishra, Paul Ipe, TANSEN: A System for Automatic Raga Identification, in: IICAI, 2003, pp. 1350–1363.
[33] Vijay Kumar, Harit Pandya, C.V. Jawahar, Identifying ragas in indian music, in: 2014 22nd International Conference on Pattern Recognition, 2014, pp. 767–772. IEEE.
[34] Salamon, Justin, Sankalp Gulati, and Xavier Serra. "A multipitch approach to tonic identification in indian classical music." In Gouyon F, Herrera P, Martins LG, Müller M. ISMIR 2012: Proceedings of the 13th International Society for Music Information Retrieval Conference; 2012 Oct 8-12; Porto, Portugal. Porto: FEUP Ediçoes; 2012. International Society for Music Information Retrieval (ISMIR), 2012.
[35] Surendra Shetty, K.K. Achary, Raga mining of Indian music by extracting arohana-avarohana pattern, Int. J. Recent Trends Eng. Sci. 1 (1) (2009) 362, no.
[36] Rohith Joseph, Smitha Vinod, Carnatic raga recognition, Indian J. Sci. Technol. 10 (13) (2017) no.
[37] S. Samsekai Manjabhat, Shashidhar G. Koolagudi, K.S. Rao, Pravin Bhaskar Ramteke, Raga and tonic identification in carnatic music, J. New Music Res. 46 (3) (2017) 229–245, no.
[38] Surendra Shetty, K.K. Achary, Raga mining of Indian music by extracting arohana-avarohana pattern, Int. J. Recent Trends Eng. Sci. 1 (1) (2009) 362, no.
[39] Varsha N. Degaonkar, Anju V. Kulkarni, Automatic raga identification in Indian classical music using the Convolutional Neural Network, J. Eng. Technol. 6 (2) (2018) 564–576, no.
[40] Joe Cheri Ross, Abhijit Mishra, Kaustuv Kanti Ganguli, Pushpak Bhattacharyya, Preeti Rao, Identifying Raga Similarity Through Embeddings Learned from Compositions' Notation, in: ISMIR, 2017, pp. 515–522.
[41] Fatemeh Jamshidi, Daniela Marghitu, Richard Chapman, Developing an Online Music Teaching and Practicing Platform via Machine Learning: A Review Paper, in: International Conference on Human-Computer Interaction, 2021, pp. 95–108. Springer, Cham.
[42] Heiga. Zen, Acoustic modeling in statistical parametric speech synthesis-from HMM to LSTM-RNN, Google (2015).
[43] Shi. Zhaoran, Wireless processor application in home music teaching based on machine learning, Microprocess Microsyst 80 (2021), 103359.
[44] Ensi Zhang, Yue Yang, Music dance distance teaching system based on Ologit model and machine learning, J. Ambient Intell. Humaniz. Comput. (2021) 1–17.
[45] Chang Liu, Bei Tu, Network piano teaching platform based on FPGA and machine learning, Microprocess Microsyst. (2020), 103414.
[46] Vishnu Pendyala, Evolution of integration, build, test, and release engineering into devops and to DevSecOps, in: Vishnu Pendyala (Ed.), Tools and Techniques for Software Development in Large Organizations: Emerging Research and Opportunities, IGI Global, 2020, pp. 1–20.
[47] Pedro. Domingos, arXiv preprint, 2020.
[48] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[49] Vishnu S. Pendyala, JoAnne Holliday, Cloud as a Computer, in: Advanced Design Approaches to Emerging Software Systems: Principles, Methodologies and Tools, IGI Global, 2012, pp. 241–249.