Applying Natural Language Processing and Deep Learning Techniques for Raga Recognition in Indian Classical Music

Deepthi Peri

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Eli Tilevich, Chair
Sang Won Lee
Eric Lyon

August 6th, 2020
Blacksburg, Virginia

Keywords: Raga Recognition, ICM, MIR, Deep Learning

Applying Natural Language Processing and Deep Learning Techniques for Raga Recognition in Indian Classical Music

Deepthi Peri

(ABSTRACT)

In Indian Classical Music (ICM), the *Raga* is a musical piece's melodic framework. It encompasses the characteristics of a scale, a mode, and a tune, with none of them fully describing it, rendering the Raga a unique concept in ICM. The Raga provides musicians with a melodic fabric, within which all compositions and improvisations must take place. Identifying and categorizing the Raga is challenging due to its dynamism and complex structure as well as the polyphonic nature of ICM. Hence, *Raga recognition*—identify the constituent Raga in an audio file—has become an important problem in music informatics with several known prior approaches. Advancing the state of the art in Raga recognition paves the way to improving other Music Information Retrieval tasks in ICM, including transcribing notes automatically, recommending music, and organizing large databases. This thesis presents a novel melodic pattern-based approach to recognizing Ragas by representing this task as a document classification problem, solved by applying a deep learning technique. A digital audio excerpt is hierarchically processed and split into subsequences and gamaka sequences to mimic a textual document structure, so our model can learn the resulting tonal and temporal sequence patterns using a Recurrent Neural Network. Although training and testing on these smaller sequences, we predict the Raga for the entire audio excerpt, with the accuracy of *90.3%* for the *Carnatic Music Dataset* and *95.6%* for the *Hindustani Music Dataset*, thus outperforming prior approaches in Raga recognition.

Applying Natural Language Processing and Deep Learning Techniques for Raga Recognition in Indian Classical Music

Deepthi Peri

(GENERAL AUDIENCE ABSTRACT)

In Indian Classical Music (ICM), the *Raga* is a musical piece's melodic framework. The Raga is a unique concept in ICM, not fully described by any of the fundamental concepts of Western classical music. The Raga provides musicians with a melodic fabric, within which all compositions and improvisations must take place. *Raga recognition* refers to identifying the constituent Raga in an audio file, a challenging and important problem with several known prior approaches and applications in Music Information Retrieval. This thesis presents a novel approach to recognizing Ragas by representing this task as a document classification problem, solved by applying a deep learning technique. A digital audio excerpt is processed into a textual document structure, from which the constituent Raga is learned. Based on the evaluation with third-party datasets, our recognition approach achieves high accuracy, thus outperforming prior approaches.

Dedication

For Amma, Daddy, Rinki and Bhargav.

Acknowledgments

First and foremost, I would like to express my heartfelt gratitude to my advisor, Dr. Eli Tilevich for being a constant source of immense support, guidance, and enthusiasm. Without his persistent help, the goal of this thesis would not have been realized. I am indebted to my committee, Dr. Sang Won Lee and Dr. Eric Lyon, for their valuable suggestions, comments and timely feedback. I am extremely thankful to my parents and my sister, for their love, understanding and support through all times, thick and thin. To Bhargav, Thank you for being my source of inspiration and motivating me throughout. To Deepika and Nivedita, thank you for your love and patience. I would not be where I am without all of you. Finally, I would like to thank all the people whose assistance was a milestone in the completion of this thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

CMD Carnatic Music Dataset

GRU Gated Recurrent Unit

HAN Hierarchical Attention Model

HMD Hindustani Music Dataset

HMM Hidden Markov Model

ICM Indian Classical Music

KNN K-Nearest Neighbor

LSTM Long Short Term Memory

MIR Music Information Retrieval

MLP Multi Layer Perceptron

NLP Natural Language Processing

PCD Pitch Class Distribution

TDMS Time Delayed Memory Surfaces

Chapter 1

Introduction

As one of the human mind's most inspired and purest forms of creativity, music is intrinsic in nature and present in everything around us. It serves as a form of expression, providing entertainment and enjoyment, and even therapy and healing. Irrespective of the many divides in the world, music is a universal language. Cultures across the globe have their own special forms of music that have evolved and blossomed, making classical music a niche art form. Amongst these, Indian Classical Music (ICM) is one the most elaborate, rule intensive and rhythmically structured variants of traditional music [1]. It is one of the oldest musical traditions in the world and has ancient roots, dating all the way back to the thirteenth century. ICM has been passed down through generations, from teacher to student and to this day, has a wide audience base. Several music festivals in India and across the world feature this musical tradition, with abundant active research exploring Music Information Retrieval (MIR) techniques in ICM as well. Research like [49] and [5] have experimentally proven that certain Ragas in ICM can have a positive effect on psychiatric treatment and overall mental health. Many forms of contemporary Indian and South Asian film and folk music draw inspiration from ICM [4].

In ICM, the core concept of the Raga is of utmost importance, as it is used to construct melodies in both Hindustani and Carnatic art music traditions. It is loosely defined as the melodic framework that acts as the grammar defining the boundaries of a musical piece [57]. We elaborate more on the construct as well as the importance of the Raga in Section 2.2.1.

The significance of the Raga concept makes Raga recognition an imperative task within MIR in ICM. Classification systems based on Raga recognition in ICM can assist musicians and connoisseurs alike in MIR tasks, such as note transcription and music recommendation. Raga recognition is highly researched, with numerous prior approaches, powered by techniques that range from Markov Models (HMMs and GMMs) [40] [59] and Pitch Class Distribution(PCD) [37] to more recent neural networks. These approaches use various defining features and/or nuances of the Raga of a musical piece to identify it. Nevertheless, the problem of Raga recognition is far from being completely solved, leaving room for improving precision and efficiency.

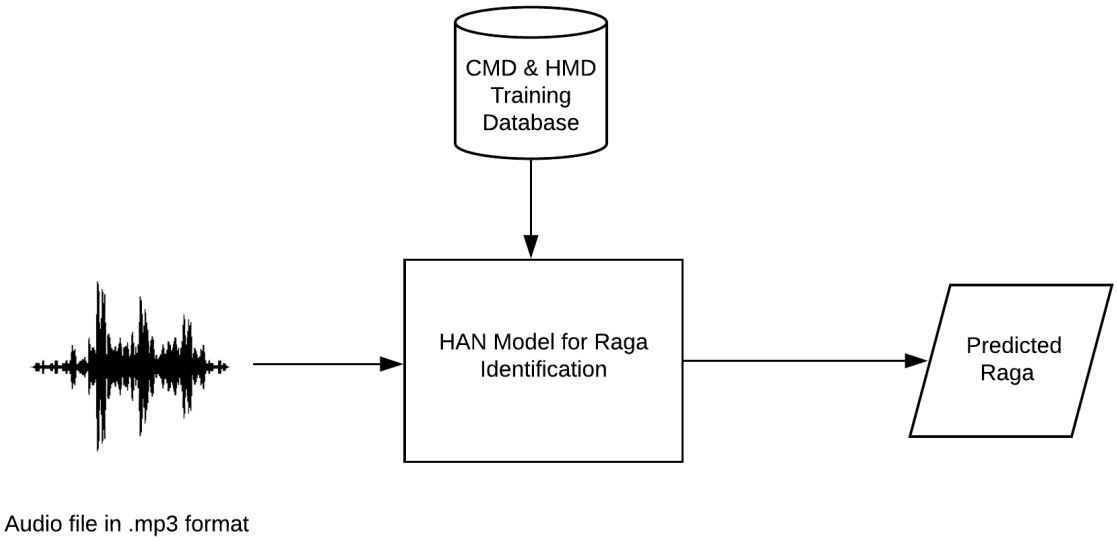


Figure 1.1: A Blackbox architecture of the Raga Recognition system using a HAN model

In this work, we investigate the Raga recognition problem from a text document classification problem's perspective. Music, like language, has many discernible patterns that can play a major role in identifying its genre or style. Natural Language Processing (NLP) relies on recognizing patterns and classifying sequences, so applying NLP to identify the Raga of

a melody can have great potential. The recent success of Hierarchical attention networks (HAN) using Gated Recurrent Units (GRUs) to classify documents has inspired this work to adapt the model for Raga recognition.

The novelty of our approach lies not only in discerning the similarities between classifying documents and recognizing Ragas, but also in reinterpreting the Raga recognition problem, so it can be solved through a model primarily used for classifying documents. We cast the digital audio excerpt into an effective sequential representation, whose structure is similar to that of a document. We break it down into subsequences and gamaka sequences, to enable classification using the HAN model. This technique can be further extended to solve other digital media classification problems, in which the input data has a rough hierarchical structure akin to that of a document (i.e., split into classifiable entities that hold information like that of words, sentences, paragraphs and the complete document itself).

In this thesis, we explain the background of Indian Classical Music and the model we implement in Chapter 2. We elaborate on the previous techniques and approaches to the Raga recognition problem in Chapter 3 and our motivation to work on the problem in Chapter 4. Chapter 5 gives a brief description of the datasets and their pre-processing and Chapter 6 expounds on the intuitions, architecture and modules of the approach. We display our observations, inferences and results in Chapter 7. Finally, we present the discussion, conclusions, and future work in Chapters 8 and 9.

Chapter 2

Background

In section 2.2 of this chapter, we elaborate on the fundamentals of Indian Classical Music and explain each element of ICM in detail. Further, in the consequent section 2.3, we describe the Hierarchical Attention Network model and its constituent entities thoroughly.

2.1 Music Information Retrieval

Music Information retrieval or MIR for short is the collective of all computational musicology tasks and as quoted by [47] is the ”extraction and inference of meaningful features from music (from the audio signal, symbolic representation or external sources such as web pages), indexing of music using these features, and the development of different search and retrieval schemes”. There have been several advancements in the field of MIR for western music, like novel systems for music classification and auto tagging[36], context based music description and indexing, Computational Assessment of the Complexity of Music Scores [23] etc. MIR based work is even being applied to tasks like developing tools that employ sonification to assist software program understanding by rendering sonic cues [24]. Although Indian Classical Music is a largely popular art music group there isn't extensive work performed in MIR in ICM.

2.2 Indian Classical music

Indian Classical Music (ICM) or Indian Art Music (IAM) can be broadly classified into Carnatic music and Hindustani music. Carnatic music is prominent in the southern regions of the Indian subcontinent while Hindustani music is more widespread in the northern and central regions of India as well as in parts of Pakistan, Afghanistan and Bangladesh, and is heavily influenced by ancient vedic and Persian traditions. Both forms are usually performed by an ensemble of musicians consisting of a lead vocalist, accompanied by melodic and rhythmic instruments like the violin, sitar, mridangam and tabla. There is also a monophonic drone instrument in the background to establish tonality,upon which the rest of the piece is built. A performance usually lasts for half an hour or more. Each melody is sung in a specific order of notes,verses and ornamentation.

2.2.1 Raga

Indian classical music is inherently different from Western music because of its embellished structure and schema of rules. There are several components of ICM that do not have a western counterpart, the foremost being the Raga or the Raag. The Raga is the melodic framework for composition and improvisation. It is based on a set of swaras on a scale and they appear in a typical order. As elegantly defined by Chordia & Senturk [11], "A Raga is most easily explained as a collection of melodic gestures along with the techniques for developing them. The gestures are sequences of notes that are often inflected with various micro-pitch alterations and articulated with expressive sense of timing. Longer phrases are built by joining these melodic atoms together." In loose translation, Raga is similar western melodic mode. It is richer and more fixed than mode but less rigid than a melody. Different Ragas sound different, like different modes, but a Raga dictates how each note is used, more

Figure 2.1: A Carnatic music concert by MS Subbulakshmi

than the modal system does.

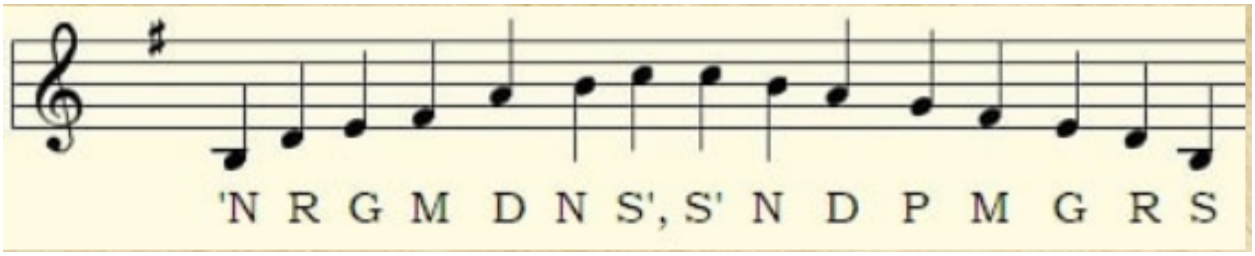


Figure 2.2: Notation of the Raag Yaman Kalyan (Or Kalyani Raga in carnatic music) [3]

The performance of a Raga is bounded within its ascending and descending note sequences, known as the arohana and avarohana. But otherwise, the artists are free to improvise other aspects like the timing between each note, the sustain and the tonic pitch, all done to expound the beauty of the Raga.

Each Raga in Indian classical music is associated with a different set of emotions(*rasa*) and different times of the day or seasons [35]. For example,one of the most popular carnatic Raga, *Kambhoji* symbolises romance, humour and pathos. There are hundreds of Ragas in use today, in art settings, in Bollywood music etc.

Considering how intriguing and unique the notion of the Raga is, it is a much researched and scrutinized course of study in Indian Classical Music. There are many works in that study the effects of a Raga on emotional and mental health, in psychology. Furthermore, there are several works that investigate the Raga recognition problem in depth. This is elaborated on in chapter 3.

2.2.2 Swara

Swara in Indian classical Music simply refers to the musical note. There are seven primary swaras corresponding to the seven basic western notes (Table 2.1). The difference between the western note and the Indian swara is that the swara is dependent on the Tonic [43]. The tonic pitch is the base pitch of the lead artist, which is usually reinforced in the background using drone instrument like a *shrutibox* or a *tanpura*. Except for the two constant or base swaras Sa (usually the tonic) and Pa(the fifth), every other swara has either two or three variations, totally yielding 12 different musical pitches altogether, known as the swarasthanams or the 12 semitones. [29]

Every Raga is typically comprised of five to seven swaras. Ragas that have all seven swaras are called complete or heptatonic Ragas.

Note in ICM	**Nomenclature in ICM**	**Note in the Western Notation**	**Frequency**
Sa	*Shadjam*	C	220 Hz
Ri1	*Suddha Rishabam*	C#	234.7 Hz
Ri2	*Chatusruthi Rishabam*	D	247.5 Hz
Ga1	*Sadharana Gandharam*	D#	264 Hz
Ga2	*Anthara Ghandaram*	E	275 Hz
Ma1	*Suddha Madhyamam*	F	293.3 Hz
Ma2	*Prathi Madhyamam*	F#	312.9 Hz
Pa	*Panchamam*	G	330 Hz
Da1	*Suddha Dhaivatham*	G#	352 Hz
Da2	*Chatusruthi Dhaivatham*	A	371.3 Hz
Ni1	*Kaisika Nishadham*	A#	396 Hz
Ni2	*Kakali Nishadham*	B	412.5 Hz

Table 2.1: Notes and Nomenclature in ICM with their western music counterparts and the relative frequency

2.2.3 Vadi and Samvadi Swaras

The vadi swara is the most important and most prominent swara in a Raga. It usually is also the most repeated swara in a Raga. It is said to be the note that reveals the melodic entity or the soul of the Raga and can also indicate the time of the day the Raga is most suited for. There can be more than one vadi swara, usually not more than two [39]. The samvadi is the second most prominent swara in the Raga. The vadi swara exhibits high relative frequency of occurrence of its pitch and it should stabilize in the shortest period of time. Hence, if two Ragas share the same set of swaras, we can differentiate between them based on the vadi and the samvadi swara.

2.2.4 Gamaka

A *gamaka* is an embellishment that is unique to Indian Classical Music. A gamaka could either be a special phrase movement or ornamentation between two swaras or an articulation on a single swara. It involves variation of the pitch of a swara with heavy oscillations, glides, skips and swings [2]. There are at least fifteen known types of gamakas in use today [29]. Each Raga derives its unique character from the gamaka and there are specific rules on the types of gamakas that can be applied to specific swaras [56].

2.3 Hierarchical Attention Network Model

Yang et al. proposed a hierarchical structured model for Document classification that mirrors the structure of a document [60]. This work is based off this model since melodies in Indian Classical music also are structured in a linear/hierarchical fashion and we can achieve a high level of success in Raga identification using the same ideology. The model dual attention

mechanisms first applied at the word level and then at sentence-level, allowing it to identify and cater to more and/or less salient content when formulating the document representation.

2.3.1 Attention

The model uses the concept of attention to identify the more important words in the text document while assigning a label to the document. Different words and sentences in a document are differently informative or relevant to the document. Yang et al extract these words that contribute more to the meaning of the sentence as well as sentences that are salient to the document and aggregate them as vectors. Hence, there are two parts to the system: Word level attention mechanism and encoder and Sentence level attention mechanism and encoder. Attention based models result in drastically better performance in classification tasks[55]. They also provide useful insight about which parts of the document is of more importance to the classification task and can contribute to performance and analysis . The Hierarchical Attention network model yields an accuracy of 75.8% on the Yahoo Answers document and 71.0% on the Yelp'15 document, trumping the other document classification models that use just an LSTM.

2.3.2 GRU encoders

The model uses a bidirectional GRU, introduced by Bahdanau et. al in 2014[6], to get annotations of words by analyzing knowledge from both directions of words and hence accruing the contextual information in the annotation. A GRU stands for a gated recurrent unit, a variation of the LSTM network. They are superior to an LSTM network [22] [58] since they retain sequential information from long ago in the network without washing it away with time or lose information that is irrelevant to the current prediction.

Since it is a bidirectional GRU, it reads the sentence s_{i} in the forward direction from word w_{i1} to w_{iT} and calculates the hidden states in the forward direction $(\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots. \overrightarrow{h}_t)$ The GRU in the reverse direction calculates the sequence of hidden states from word w_{iT} to word w_{i1}, $(\overleftarrow{h}_t, \overleftarrow{h}_{t-1}, \ldots. \overleftarrow{h}_1)$. Bahdanau et al. states that "the GRU has two gates as opposed to three: The reset gate r_{t} and the update gate z_{t}". These gates are responsible for computing the current hidden state $\mathrm{h_t}$:

$$h_t = (1 - z_t) \odot h_t - 1 + z_t \odot \tilde{h_t} \tag{2.1}$$

where $h_{\mathrm{t\text{-}1}}$ is the previous state and $\tilde{h_t}$ is the current new state, called the candidate state.

$$\tilde{h_t} = tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \tag{2.2}$$

The update gate z_{t}, as the name suggests, is responsible for deciding if the past information is kept and how much of it is included, as well as how much of the new information is kept.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{2.3}$$

The reset gate r_{t}, controls how much the previous state contributes to the candidate state. Since its a sigmoid, the value ranges between 0 and 1. If it is 0, the previous state is forgotten and if the value is 1, its is completely propagated to the next state.

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2.4}$$

2.3.3 HAN Model

The word annotation h_{it} is fed through a Multi Layered Perceptron(MLP) to get u_{it}. It is then aggregated with its corresponding value in the word level context vector u_w, which is essentially a high level vector representation of how important each word in the sentence is. It is initialized randomly and is a trainable parameter [51]. They obtain the normalized importance weight $_{it}$ through a softmax function and compute the sentence vectors_i as a weighted sum of the word annotations. Further, a similar process is followed for sentence encoding and attention as well.The sentence level context vector u_s is used to measure the emphasis of the sentence in the document. We elaborate further on our iteration of the model for the Raga identification model in the Model and Architecture chapter (chapter 6

Chapter 3

Related Work

Ragas are a pivotal notion in Indian Classical music (ICM) and Raga identification is a significant research subject in the field of Music Information Retrieval(MIR) in ICM. There are many preceding works that have attempted to perform automatic recognition of the Raga of a given melody, out of which a greater part of them are statistical modelling and Machine learning based methods. These methods either conform to a sequence classification method like this work or can be Pitch Class Distribution based.

The first substantial automatic Raga recognition system built was Tansen, by Pandey et al[38]. It was a breakthrough in the field of computation musicology in ICM. Tansen used a Hidden Markov Model (HMM) augmented by a string matching algorithm. Although the data points consist of only two Ragas Yaman and Bhupali, Tansen achieves an accuracy of 77%. Following the success of Tansen, several methods were proposed using GMMs and HMMs. [42] uses a Hidden continuous Markov model with pakad matching to detect complete Ragas, using a larger database. Similarly, [12] uses a Continuous Density HMM. Although HMMs achieved considerable amount of success in the Raga Identification task, it is also highly reliant on the size of the dataset.

Pitch class distribution (PCD) gives the relative frequency of each scale degree. It is a histogram of the pitch tracks, where the bins correspond to the notes in n number of octaves, after the scale has been centered around the tonic (section 5.2.2) for the specific audio signal. It is natural that this technique would seem intuitive in the Raga identification task since each

bin or vector in the PCD can correspond to each of the 12 semitones(swarasthanams) (section 2.2.2). Chordia and Rae [10] generate Pitch class distribution (PCDs) and Pitch Class Dyad Distributions (PCDDs) using Harmonic Pitch Class Profiles (HPCPs) and perform supervised classification on them using a Support vector machine(SVM). They use a dataset with 17 Ragas and achieve an accuracy of 78.1% for PCDs and 97.1% for PCDDs, which is state of the art for the PCD based methods. Since PCD represents swara salience as pitch histogram, it does not give rise to pitch - octave errors, where a pitch is binned into the incorrect octave. Therefore, PCD based Raga identification techniques are known to perform well for large datasets as well. Inspired from Chordia et al.[10], Dighe et al [13] extract feature vectors from audio chromagram patterns instead of HPCP features and extract swara based features to use for Raga modelling and identification. The scale of the performance is not known prior to classification in this case. G.K.Koduri et al.(2014) [28] propose two parametrized methods to categorize pitch contours based on melodic context: obtaining performance pitch histograms and generating context-based swara distributions. Other notable works like Chordia & Senturk [11], G.K.Koduri et al.(2012) [27], and Belle et al. [7], and [30] also use variants of PCD, like coupling with n-gram and Fine-grained Pitch Distributions (FPD) for Raga identification. Another improved version of PCD methods, are the ones involving generating the distribution with the help of Kernel Density Estimation (KDE). This method [33] shows a slight improvement as compared to the former two methods involving PCD.

The drawback with most of these methods is that they only consider monophonic audio settings and disregard the polyphonic nature of ICM. They also do not consider the temporal aspect which can be descriptive in a Raga recognition task.

Apart from using the Swara set for Raga identification, many works have used other audio context based features to perform Raga and Tonic identification. Shridhar et al.[50] and

Shetty at al.[48], use elements of ICM like the Arohana-Avarohana pattern i.e., the ascent and descent of the swaras in the Raga, thereby capturing the temporal facet of a melody. Dighe et al.[14] emphasize on the the vadi swara (section 2.2.3) in a Raga and uses it as the tonal feature, as opposed to all the previous works using the pitch. Although the Arohana - Avarohana and the vadi swara are integral factors and can facilitate Raga recognition, they are not as informative as compared to the entire audio source itself.

Our work is inspired from the current state of the art approach, the Time-Delayed Melodic Surfaces(TDMS) for Raga recognition, introduced by Gulati et al. [19]. The algorithm does not require transcription of the melody or conversion to a different representation and details both the tonal and temporal characteristics of the Raga. TDMS obtains a 2D surface that highlights the swaras in the audio as well as the transition between them (gamakas) and then applies a KNN classifier for the Raga Recognition task.

Figure 3.1 shows the 2D surfaces generated by TDMS to extract temporal features for Raga recognition of a music piece, before and after their pre-processing algorithm. They achieve an accuracy of 98% on the Hindustani music dataset with 30 Ragas and 87% on the carnatic music dataset with 40 Ragas (Section 5.1).

As far as Deep Learning methods in Raga recognition goes, there is not much precedent to refer to. Suma et al. [52] use an Artificial Neural Network(ANN) on a non sequence based 36 -feature vector derived from the PCD of the melody and limit themselves to only carnatic music. Madhusudhan et al. [32] use a 5 layer deep CNN for Tonic independent classification of Ragas. They state that achieves an accuracy of around 72-77% on two test datasets containing 3 and 7 Ragas each. Again, the results aren't conclusive enough since the dataset sizes are limited.

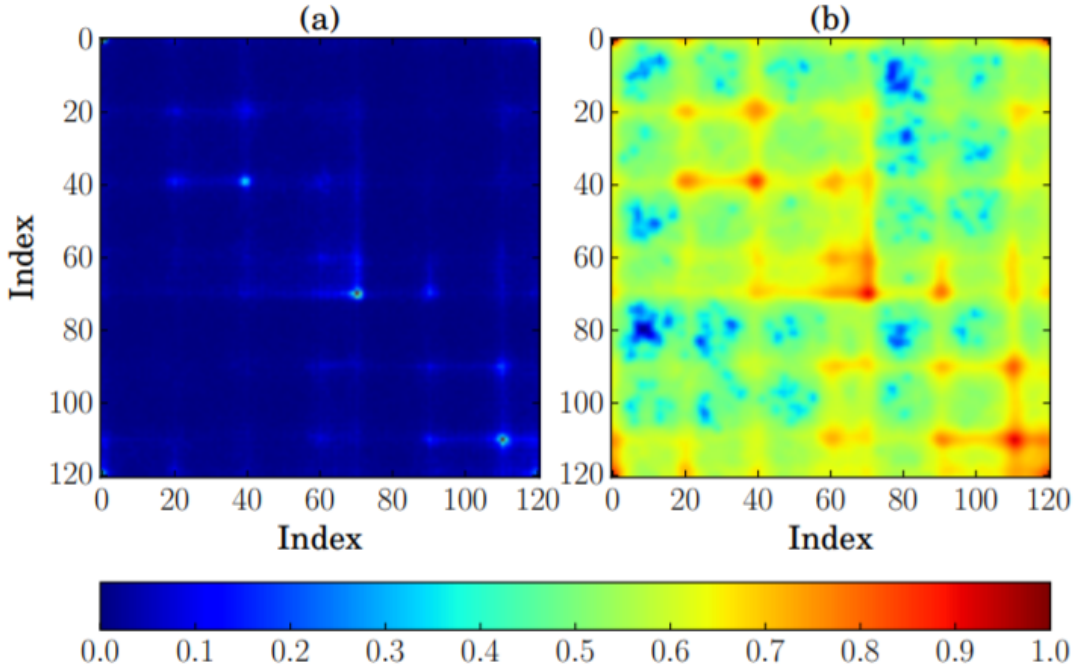


Figure 3.1: 2D surfaces generated by TDMS to extract temporal features for Raga recognition of a music piece

Chapter 4

Motivation and Intuitions

Rather than a mode or a scale, the Raga is a unique, precisely structured, harmonious and symphonic system in ICM. This makes Raga recognition an imperative task, pivotal to research in music information retrieval in ICM. Raga recognition can pave the way for other research in techniques like:

- Music recommendation systems in ICM
- Organizing music data collections for Carnatic and Hindustani music
- Automatic composition of music
- Automated arrangement of instruments/accompaniments in a concert setting
- Automated note transcription

We treat the Raga recognition problem as a sequence classification problem. The intuition to carry out Raga recognition by treating it as a sequential input to a pattern recognition system is from the fact that most Raga recognition models do not consider:

a) The temporal aspect of the Raga,

b) The gamakas in the audio excerpt, and

c) The polyphonic nature of the song.

We can take all three of the above into account with our proposed model.Moreover, most systems are not fully automatic, since they require manual input of the tonic note or the fundamental frequency. In a musical rendition, if a pattern occurs repetitively, it cannot be coincidence. The lead artist renders the same fragment multiple times to impose some level of significance on it. A pattern identification and hence a sequence classification system is hence intuitive for a task like Raga recognition.

Each song is divided into subsequences and gamaka sequences. In the HAN model, each pitch represents a word, each gamaka sequence represents a sentence, and each subsequence represents a paragraph. We apply attention after each encoding layer.

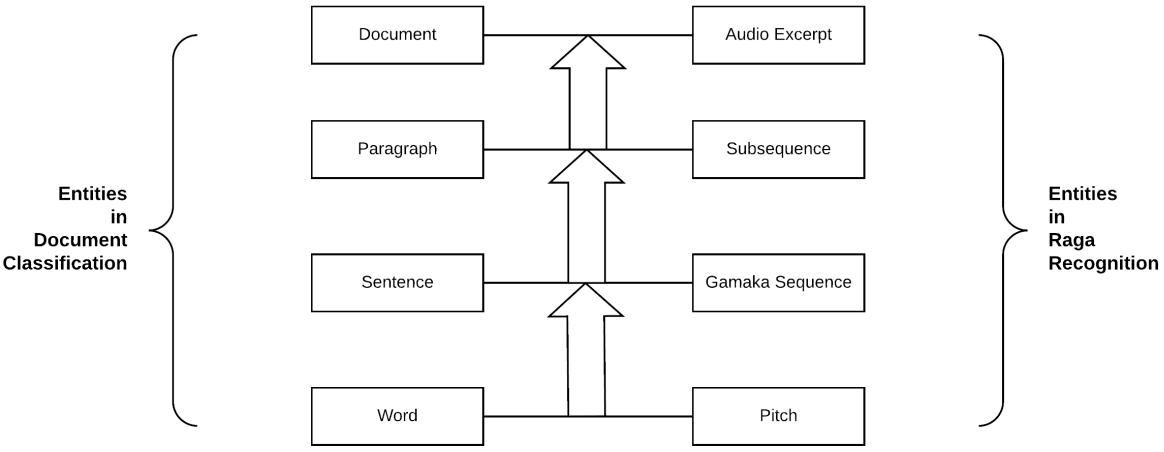


Figure 4.1: Elements of the Document Classification task analogous to the elements of the Raga Recognition task

The assumption is that the most important swaras in the Raga, the vadi and the samvadi swaras, are identified as the salient components of the sequence being classified by the attention mechanism. Also, we aim to capture and learn information about the gamaka in each audio excerpt to better distinguish between similar Ragas.

Chapter 5

Evaluation Datasets

5.1 Description

We use the Indian art Music Raga recognition datasets from the CompMusic Corpora, assembled by the Music Technology Group at University of Pompeu Fabra, Barcelona, Spain. [19] [20] They comprise of two significantly large datasets: Carnatic music dataset (CMD) and Hindustani music dataset (HMD). They contain full length mp3 audio recordings and their corresponding Raga labels. They are the most extensive and exhaustive datasets available for research in the Raga recognition field. The Carnatic music dataset comprises of 124 hours of audio recordings and associated metadata that includes verified Raga labels. The metadata is corroborated and revised by domain experts. There are 480 recordings from 40 Ragas, i.e. 12 recordings per Raga. Each recording has a Raga label and an MBID. MBID stands for MusicBrainz Identifier. It is a 36 character universally unique identifier that is permanently assigned to each entity in the database. Figure 5.1 gives the summary of the Carnatic music dataset.

Similarly, the Hindustani music dataset (HMD) contains 116 hours of audio recordings with their corresponding Raga labels and MBID. It contains 300 recordings in 30 Ragas, with 10 recordings for each Raga. Figure 5.2 gives the summary of the Hindustani music dataset.

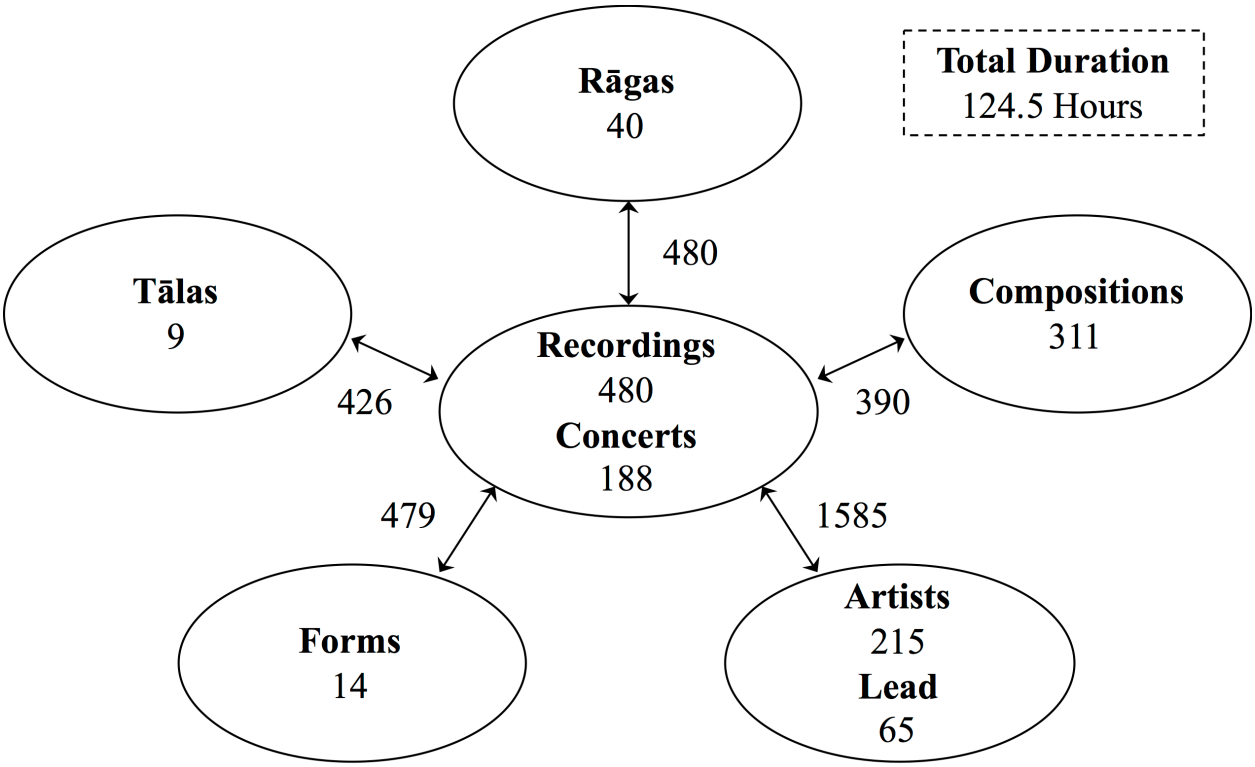


Figure 5.1: Carnatic Music Dataset (CMD) description, assembled by the MTG at UPF

5.2 Pre-processing

In this section, we describe the pre-processing techniques applied on the data before performing Raga recognition. We discuss the salience of predominant melody estimation and tonic normalization and go on to explain the data transformation techniques used to make the data suitable for our model.

5.2.1 Predominant Pitch Estimation

Predominant pitch extraction or predominant melody extraction refers to the fundamental frequency f_0 estimation of a polyphonic melody. Pitch estimation is an active research topic in MIR. The challenge in predominant pitch extraction lies in whether the audio is

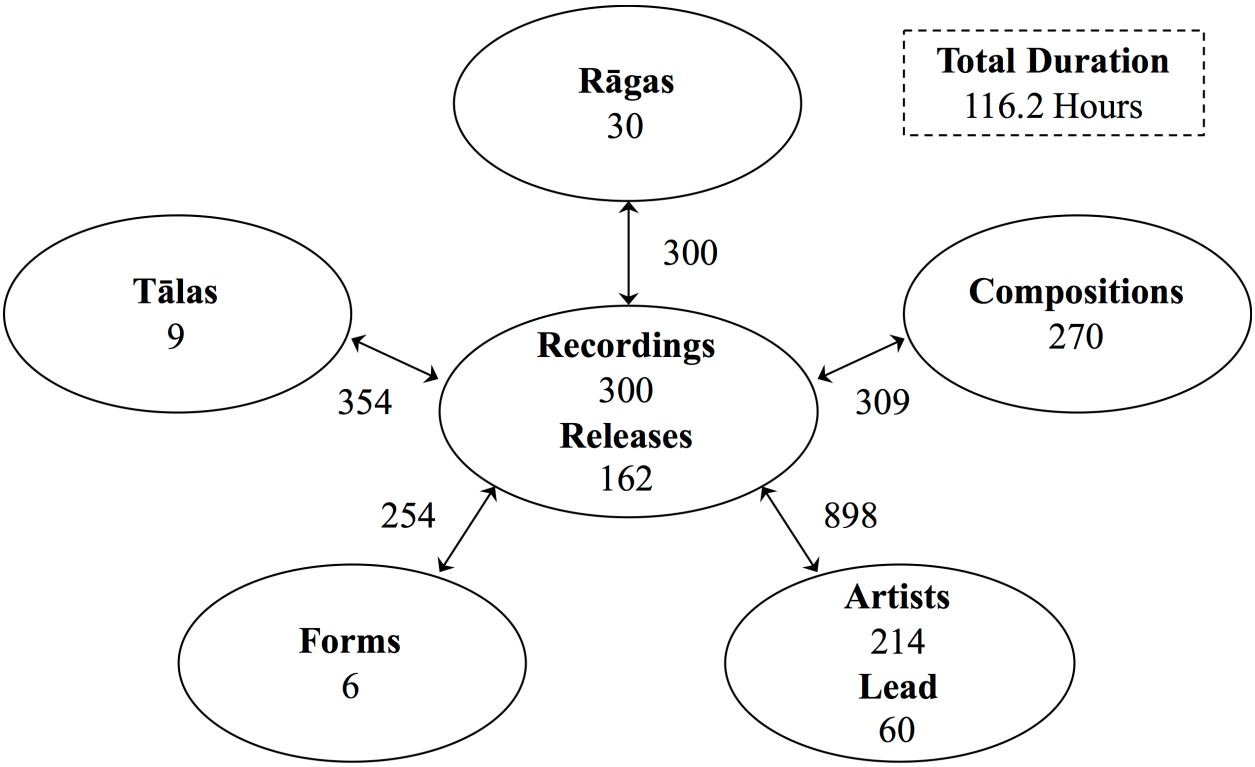


Figure 5.2: Hindustani Music Dataset (HMD) description, assembled by the MTG at UPF

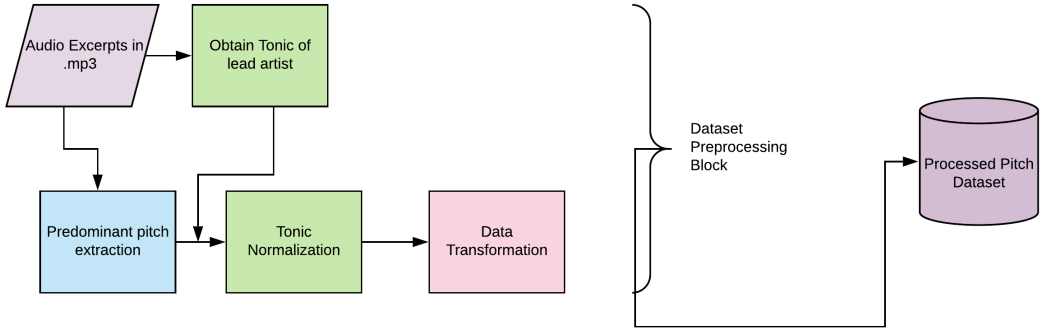


Figure 5.3: Data pre-processing of audio excerpts to get the pitch sequences of the lead artist.

monophonic or polyphonic. It is especially problematic if the melody is highly polyphonic [46]. Melodies in Indian classical music are polyphonic since they have many string and

rhythm instruments in the background in accompaniment to the lead singer. There might also be more than one singer. But in comparison to other western music genres, music in ICM is sparsely polyphonic. This makes the task of predominant pitch estimation slightly easier. We need to perform this before the other pre-processing tasks since we need to identify the pitch of the lead singer to proceed.

Several previous works have proposed methods for fundamental frequency estimation in melodies. The majority can be classified as *salience-based methods*. Works like [15], [17] use the spectral representation of the audio signal to estimate the *time-frequency* representation of the pitch salience. The spikes in this representation are considered to be the fundamental frequency f_0. These approaches do not use any audio source splitting techniques to procure a signal that contains only the lead singers vocals and therefore, many of them are only applicable to monophonic audio settings. Other approaches provide methods to detect the predominant melody in polyphonic audio settings using techniques like tracking the additional pitches apart from the lead voice in polyphonic audio [44], and utilizing frequency relationships of simultaneous components [26].

Salamon et al.[45] propose an approach based on the creation and characterization of pitch contours.The algorithm estimates the fundamental frequency f_0 of the polyphonic audio signal using four blocks, namely: *Sinusoid Extraction,Salience function computed using harmonic summation, contour creation and Melody selection.*The algorithm does not give rise to any pitch octave errors at the frame level and is hence state of art for the task.

We use the predefined pitch estimation function variant of the algorithm available in Essentia [9]. Essentia is an open source C++ library with python binding for audio analysis and audio based music information retrieval (MIR). The function takes in the audio signal as the input and returns a vector of estimated melody pitch values and a vector of confidence values. We use the default values of the parameters for the function with hopsize as 196 and the sample

rate as 44100. In audio analysis, hop size refers to the number of samples that represent a single recorded pitch and the sample rate is the number of samples taken from 1 second of the sound wave. Before the predominant pitch estimation function, we apply an equal loudness filter as recommended by [9] to enhance or attenuate the frequencies and make the processing smoother. Finally, we obtain the vector with the predominant melody to process further.

The temporal resolution TR, i.e. the duration of the analysis window is given by the hop-size/sample rate. We can estimate the relationship between the pitches and time through this. If β is the number of pitches the song is represented by and t_s is the duration of the song in seconds:

$$TR = HopSize/SampleRate \tag{5.1}$$

$$\beta = TR/t_s \tag{5.2}$$

We further divide the processed audio pitch sequences of length β into S_r length subsequences and G_s length gamaka sequences.So in essence, a 5000 length subsequence of pitches would represent approximately 23 seconds of the song and a 50 length gamaka sequence would represent about 2.3 seconds.

5.2.2 Tonic Normalization

As mentioned earlier, in an ICM performance the lead singer's tonic serves as the reference pitch and is reinforced in the background by a drone instrument 2.2.2. The pitch of this drone instrument is considered the *Tonic* of the whole audio excerpt.

As different performances of the same Raga can have different tonic pitches, all the notes of the Raga are translated to the corresponding tonic pitch's octave. This induces translational invariance, which needs to be rectified before the pitches are input to the model.

We redress this problem by normalizing the predominant melody pitches of each audio excerpt with their corresponding tonic pitch. A cent is a unit of measure for the ratio between two frequencies, a logarithmic unit used for music intervals. There are 100 cents in each half-step. The cent was introduced as a unit that allows us to calculate and consider what we heard using a logarithm scale. It is more intuitive than frequency in hertz. We identify the Tonic of each audio excerpt using Essentia and use this to normalize its pitches. The mathematical expression below illustrates this:

$$f_n = 1200 * \log_2(\frac{f}{T_n}) \tag{5.3}$$

Where f_{n} is the tonic normalized interval in cents[18], f is the original frequency in Hz, and T_n is the Tonic of the audio excerpt.

The expression in 5.3 will result in 100 cents between each half-step, i.e. there can be 100 possible cent values between two notes in the octave. This will result in a large number of unique cent values hence increasing the size of the vocabulary. Experimentally, we have determined that we can reduce the size of the vocabulary by shrinking the number of cents separating two notes without losing much of the information. Therefore, we modify 5.3 as follows,

$$f_n = (1200 * \log_2(\frac{f}{T_n})) * (n_c/100) \tag{5.4}$$

Where n_c is the constrained number of cents between each half-step. We round the normalized pitch to make sure the vocabulary consists only of integer values. We round the obtained f_n to the nearest integer to discretize the pitches in the sequence.

5.2.3 Data Transformation

Data Transformation is the pre-processing pipeline's final step. As mentioned earlier, we model the Raga detection problem as a sequence classification task. The purpose of this step is to transform the pitches of each excerpt to suit the input requirements for a sequence.

The first step in most deep-learning based sequence classification models is the word embedding layer. The function of this layer is to convert each word into a vector so that similar words can have similar vectors. The embedding layer takes the indices of each unique word in the whole dictionary as input. In document classification tasks outlined in works like [34], [21], the word embedding technique is elaborated on in meticulous detail, since it is a dynamic way of representing text in NLP tasks. The embedding technique is ideal in cases where we want to convert to a real valued vector representation. Each word is mapped to a unique integer.

In our case, each word corresponds to a pitch. We observe that the pitches obtained after Tonic Normalization were not contiguous, i.e. the range of the pitches had some gaps and some values in the range did not correspond to any pitch. There were also several pitch values which were in the negative spectrum. The embedding layer takes only non-negative

and contiguous values as input. Hence, we transform the Tonic Normalized pitches to a representation suitable for embedding.

We obtain the total number of unique pitches in the whole dataset after Tonic Normalization. We then assign a unique index ranging from 1 to the total number of unique pitches for each pitch in the processed audio excerpts. This way, we make sure that the transformed pitches are non-negative and lie in the defined contiguous range.

Chapter 6

Methodology and Model Architecture

After obtaining the transformed dataset, we proceed to implement the model on it. In this section, we describe our model in depth and explain the design choices we make. As elaborated on earlier in Section 2.3, we use a Hierarchical Attention Model to perform classification on the Raga datasets. We outline our architecture in figure 6.4 and explain the modules in the sections below.

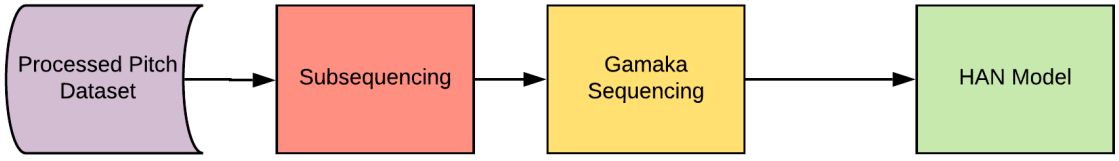


Figure 6.1: Data preparation before input to the model. This includes subsequencing and gamaka sequencing of the audio excerpts.

6.1 Methodology

6.1.1 Subsequencing of audio excerpts

As explained in the 5.2.3 we have transformed each audio file in the ICM datasets to a sequence of non-negative values representing each pitch in the excerpt. The sequences obtained

are of lengths around 100,000 to 300,000 pitches. Training a neural network with sequences of such extensive lengths is impractical and will require copious amounts of resources.

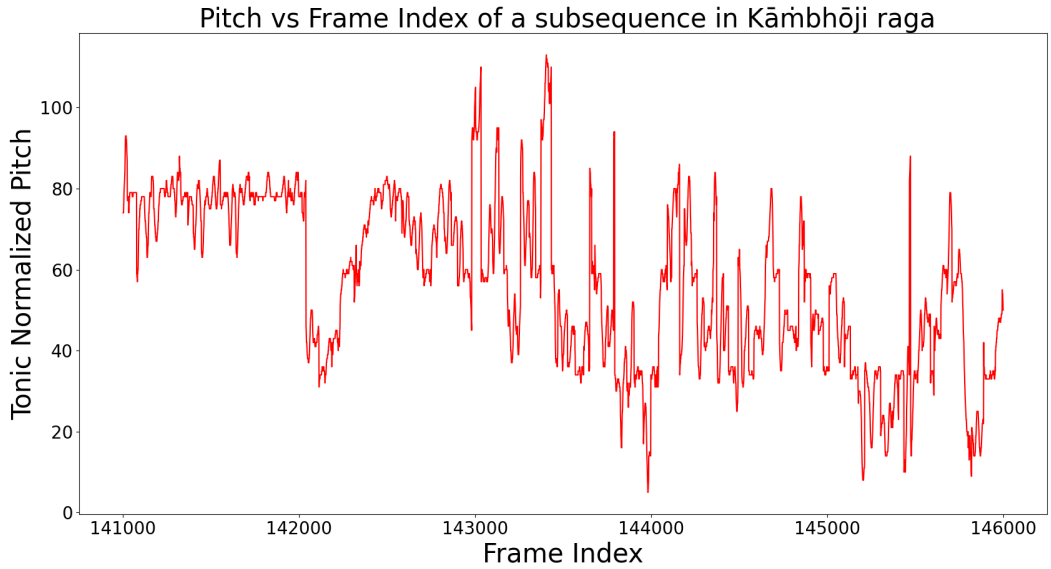


Figure 6.2: Pitch-Frame Index plot for a subsequence of length 5000 in Raga Kambhoji

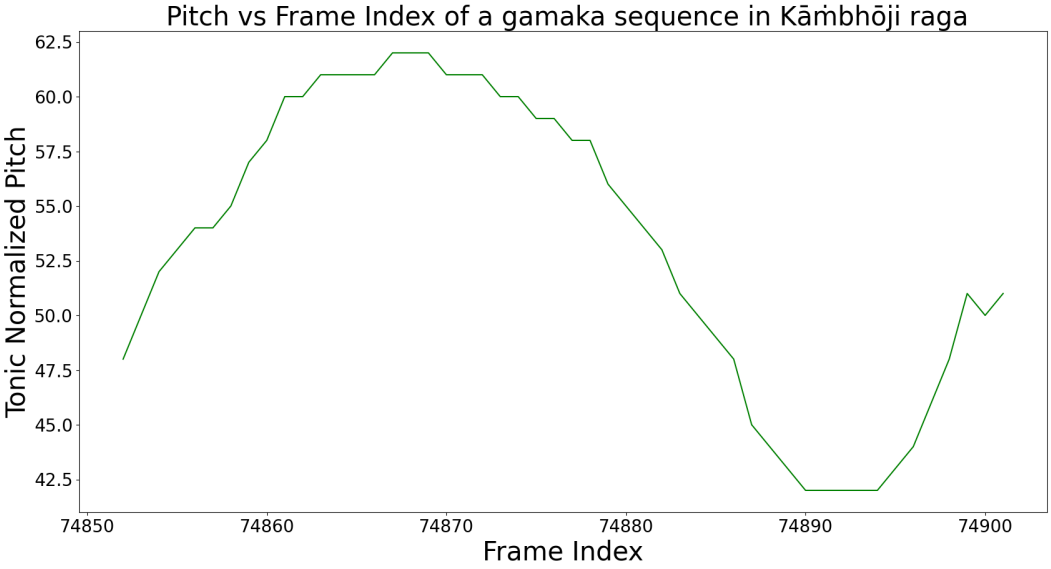


Figure 6.3: Pitch-Frame Index plot for a gamaka sequence of length 50 in Raga Kambhoji

To circumvent this complication, we decide to break down the pitch sequences into much

shorter subsequences each. This way, we try mimic the manner in which a *rasika* or a connoisseur of ICM judges the Raga of a song by listening in parts.We also want the subsequences to be able to capture information about the vadi and samvadi swaras of the Raga, to help identify it. We assume that most of these subsequences will carry enough information to help the model learn crucial characteristics about a Raga of the audio excerpt as well. We plot the pitches across time for the gamaka sequence in Figure **??** to assert this and we see that there definitely are subtle nuances that can be captured through a gamaka sequence of shorter lengths. We achieve best results with subsequences of lengths 4000 to 6000. We randomly sample subsequences from the initial pitch sequences to avoid bias.

6.1.2 Gamaka sequencing of subsequences

We further divide the subsequences into shorter length gamaka sequences to capture the particular gamaka information for each Raga. We see in Figure 6.3 the pitch versus time plot of the gamaka sequence in a melody in Raga kambhoji displays the aroha and the avaroha gamakas. As discussed in 2.2.4, a gamaka is a characteristic embellishment of the shift from one note to another. It is crucial for differentiating very similar Ragas. Two Ragas can have the same set of notes by vary with respect to their gamakas.

The ideal length of a gamaka sequence is empirically determined to be of size 50. In contrast to subsequencing, we do not randomly sample the gamaka sequences. Instead, we split the whole subsequence into gamaka sequences of equal length contiguously.

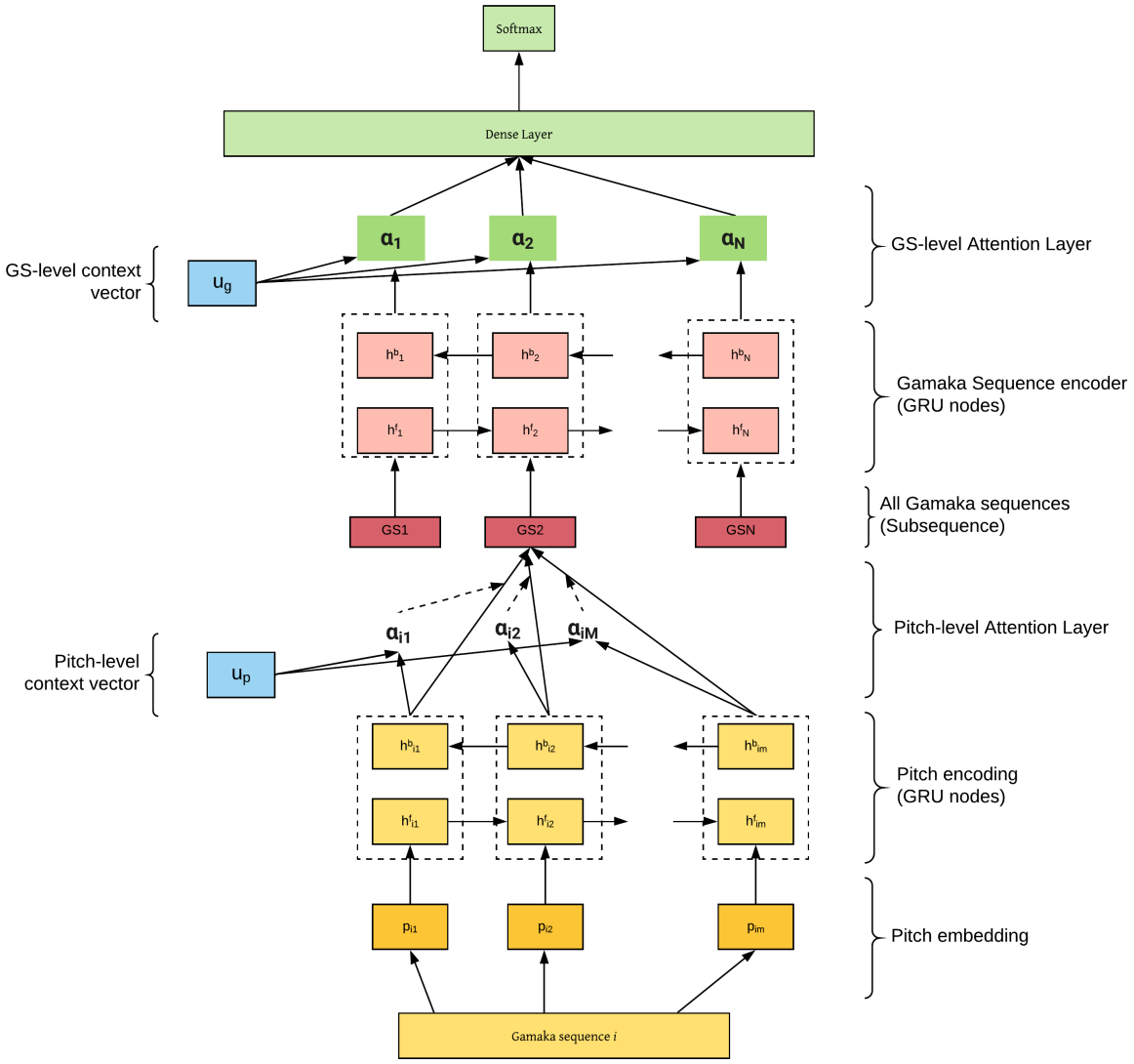


Figure 6.4: Hierarchical Attention model Architecture for Raga recognition

6.2 Model Description and methods

6.2.1 Pitch Embedding

We pass each gamaka sequence in the subsequence into a Pitch Embedding layer. The purpose of the embedding layer is to convert the pitches to real valued low-dimensional vectors. In order to capture sequential information, it is important to capture information of the pitches relative to each other in the sequence. The pitch embedding layer is responsible for learning this contextual relationship between the various pitches.

This technique is widely used in Natural Language Processing tasks [31] [16] [53], especially in document classification. In this task, there is a word embedding layer as the input layer which converts every word into an n dimensional vector. Identical words, that have the same context, will have similar vectors. For example, the words *King* and *Emperor* will have very similar vectors and the vector distance between the words *Man* and *King* and the words *Woman* and *Queen* will be comparable. We treat each word as a pitch and apply the embedding layer for a similar effect.

Essentially, this similarity information of the pitches is imperative to learn the sequential pattern for classification. For our model, we convert the pitches to a 128 dimensional vector. Since the weights for the embedding layer are not already predefined in our case, we randomly initialize these weights and let them train implicitly along with the model. This gives us the pitch embedding matrix P_e.

6.2.2 Pitch Encoding

After we have obtained the 128-dimensional vector representation of the pitches, we pass it to the Pitch Encoder. It contains a bi-directional GRU cell for each pitch vector. This layer

learns the sequential information of the pitches in both directions of the gamaka sequence and gives us pitch annotations.

The GRU cells has 2 control gates, an update gate to learn new information about a pitch in the gamaka sequence, and a reset gate to limit the flow of irrelevant information further down. Each GRU cell also has an output gate, that passes the information from the current unit to both the next cell and the attention layer. In our model, there are 100 nodes in each hidden layer of the GRU cell.

Given a sequence with pitches p_{im}, where $m \in [0, M]$ we first embed the pitches into vectors using our previously generated pitch embedding matrix P_e. h^f_{im} contains the pitches encoded by the GRU in the forward direction and h^b_{im} contains the pitches encoded by the GRU in the backward direction. Further, both h^f_{im} and h^b_{im} are concatenated together as h_{im} to obtain the pitch annotation vector.

$$p_{im} = P_e g_{im}, m \in [1, M], \tag{6.1}$$

$$h^f_{im} = \overrightarrow{GRU}(p_{im}), m \in [1, M], \tag{6.2}$$

$$h^b_{im} = \overleftarrow{GRU}(p_{im}), m \in [M, 1], \tag{6.3}$$

$$h_{im} = [h^f_{im}, h^b_{im}]. \tag{6.4}$$

6.2.3 Attention Layer

The GRU encoded pitch annotated vectors are then passed on to the Gamaka level attention layer to capture salient information. In section 2.3.1, we explain the concept of attention [55] [41] and how it can amplify the efficacy of the model, amplifying the more important

information.

When we pass the encoded pitch vectors corresponding to the gamaka sequences to the attention layer, we expect the pitches which are most likely to be the vadi and the samvadi swaras (Section2.2.3) to be amplified as the salient pitches. This is done by multiplying the pitch vector values with their corresponding salience value in the pitch level context vector U_{p}. The pitch level context vector U_{p} is a randomly initialized vector that is trainable over time and learns the salience values for each pitch. This way, the model can learn to identify various gamakas with different vadi and samvadi swaras to differentiate between very similar Ragas.

Specifically, we pass the encoded pitch annotation h_{im} to a single layered MLP to obtain a hidden representation u_{im}. Then we apply the softmax function on u_{im} with the pitch level context vector u_p to obtain the normalized weight α_{im} .

$$u_{im} = tanh(W_p h_{im} + b_p), \tag{6.5}$$

$$\alpha_{im} = \frac{exp(u_{im}^T u_p)}{\sum_m exp(u_{im}^T u_p)}, \tag{6.6}$$

$$GS_i = \sum_m \alpha_{im} h_{im} \tag{6.7}$$

The output of the attention layer will be a single vector that holds information gathered about the whole gamaka sequence. We get the weighted average of the normalized weight α_{im} and the encoded pitch annotation to get the Gamaka Sequence vector gs_i to be passed the next level of encoding and attention. The same process is repeated for all the gamaka sequences in the subsequence to be aggregated and passed onto the Gamaka sequence encoder.

6.2.4 Gamaka Sequence Encoding and GS level Attention Layer

The Gamaka Sequence encoder has the same function as the pitch encoder, but for the aggregated output obtained from the previous level in network. Again, it comprises of bi-directional GRU node for each aggregate vector. It learns information in both the forward and the backward directions of the aggregate vector. We have 100 nodes in each hidden layer of the GRU cell this level as well.

The Gamaka Sequence vector GS_i obtained from the previous level is passed through both the forward GRU and the backward GRU to obtain h_i^f and h_i^b. The concatenated annotation of h_i^f and h_i^b is represented as h_i.

$$h_i^f = \overrightarrow{GRU}(GS_i), i \in [1, N], \tag{6.8}$$

$$h_i^b = \overleftarrow{GRU}(GS_i), i \in [N, 1], \tag{6.9}$$

$$h_i = [h_i^f, h_i^b]. \tag{6.10}$$

After the encoding layers, the output is passed to the Gamaka Sequence level attention layer to capture the important gamaka sequences in the subsequence. The GS level context vector U_{gs} holds the salience values of each gamaka sequence. Each gamaka sequence is multiplied by it's salience value and it's importance is amplified or decreased accordingly.

we pass the encoded gamaka sequence annotation h_i to a single layered MLP to obtain a hidden representation u_i. Then we apply the softmax function on u_i with the gamaka sequence level context vector u_g to obtain the normalized weight α_n .

$$u_i = tanh(W_{gs}h_i + b_{gs}), \tag{6.11}$$

$$\alpha_n = \frac{exp(u_i^T u_g)}{\sum_m exp(u_i^T u_g)}, \tag{6.12}$$

$$v = \sum_i \alpha_n h_i \tag{6.13}$$

Finally, we get the weighted average of the normalized weight α_n and the encoded gamaka sequence annotation h_i to get the final subsequence vector v.

6.2.5 Dense and Softmax Layers

We then pass this output vector v from the second level attention layer to a Dense layer of 100 nodes. This layer can learn additional information to associate the specific attention patterns to their corresponding Ragas. Finally, the Softmax layer is used to generate the probability scores for each of the output Ragas, since it is a multi-class classification problem. The final classification outcome is the Raga with the highest probability.We use the high level representation of the audio, the vector v we generated from the previous steps for feature representation:

$$c = softmax(W_a v + b_a) \tag{6.14}$$

6.3 Hyper-parameter tuning and optimization

- **Optimizer**: While training, we compared the performance of the models using both RMSProp [8] and Adam optimizers [25]. We observed a much faster learning speed

with the Adam optimizer and proceeded to use it for all of the model iterations.

- **Subsequence Length** S_r: We experiment with three different subsequence length values S_r: 4000, 5000 and 6000 pitch sequences. We choose these values based on the intuition that an expert *rasika* can identify the Raga in 1 minute for an audio of an average length of 30 minutes. 1 minute approximately translates to 5000 pitches for our pitch sequences. We empirically conclude that 4000 length subsequences give us the best results.

- **Gamaka Sequence Length** G_r: We start with 40 and experiment with gamaka sequences lengths of up to 60. We notice that 50 length sequences yield superior accuracy in comparison to other lengths of gamaka sequences.

- **Number of Hidden layer nodes**: We train the model with 100 and 200 nodes in each hidden layer. We notice that the models with 200 hidden layers perform slightly better than the models with 100 nodes.

Chapter 7

Observations and Results

7.1 Training

Our assumption is that every subsequence contains enough information to identify the Raga and hence, we train our model to classify these subsequences. After we compile the model, we train the model on 80% of the training set and validate on the remaining 20% of the training set. The training dataset is split into the subsequences x and their labels in y. The labels are in one hot encoded form. We use the Adam optimizer Categorical Cross Entropy loss function to train the model. This is the most appropriate loss function for a multi-class classification problem since the labels are in categorical format and only a single label applies to each vector to be classified. This means that the correct Raga gets the highest probability distribution of 1 and the rest get 0.

We train for 10 epochs with a batch size of 50 and use Accuracy as our metric to judge the classification's success rate.

7.2 Testing

After we have successfully trained our model, we want to test the efficacy of the model by predicting the Raga for a melody. Although we train at the subsequence level of the excerpt,

we want to predict the Raga for the entire audio excerpt. Hence for testing purposes, we generate subsequences for the test audio excerpt and classify each subsequence using the trained model. We consider that a Raga class gets one vote if one of the subsequences is classified as that Raga. We then aggregate the votes for each Raga and choose the Raga with maximum votes as the classified Raga for the entire audio excerpt.

We perform holdout and k-fold cross validation for testing our model. In this method, each song in the dataset appears in the test set exactly once. For the CMD, we use $k = 12$, since the dataset has 12 songs of each Raga. Similarly for HMD we use $k = 10$. We split the dataset into k folds, train on $k\text{-}1$ parts of the dataset and predict the Ragas for the remaining 1 fold. Furthermore, we perform prediction with maximum voting and obtain the accuracy of the predictions for each test set. We then average out the accuracy to obtain the final testing accuracy of the model.

7.3 Results

CMD has 12 songs per Raga and HMD has 10 songs per Raga. We divide each audio excerpt into N_s number of subsequences. Hence we have $12*N_s$ number of subsequences for each Raga in CMD and $10*N_s$ number of subsequences for each Raga in HMD. We further divide these subsequences into N_g gamaka sequences per subsequence. This makes the classification task a balanced one. Therefore, we use accuracy and loss as our measurement metrics since they are the most suitable metrics for a balanced classification problem like ours. We compare our results to Gulati et al.'s TDMS model (current state-of-the-art). To facilitate a just comparison, we use the same metrics as them and also perform holdout with n-cross fold validation.

To arrive at the optimal values for the hyper-parameters subsequence length S_r, gamaka

Subsequence length S_r	Gamaka sequence length G_r	Number of Hidden Layers HL	Training accuracy	Validation Accuracy	Testing Accuracy
4000	50	100	99.12	97.47	88.75
4000	**50**	**200**	**98.98**	**97.82**	**91.25**
5000	50	100	97.07	96.22	90.00
5000	50	200	99.26	98.20	88.75
6000	50	100	99.28	97.32	90.00
6000	60	100	99.39	98.02	88.25
6000	50	200	99.5	98.81	90.00

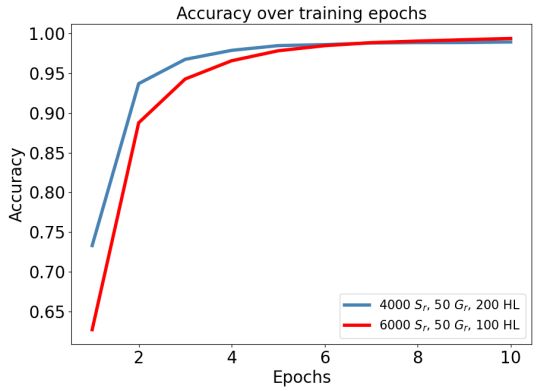
Table 7.1: Empirical results from series of experiments identify the ideal Subsequence length S_r, Gamaka sequence length G_r and Number of hidden layers HL

sequence G_r and number of hidden layers HL we perform testing with a train test split of 80-20 on the datasets. We display the empirical results below in table 7.1:

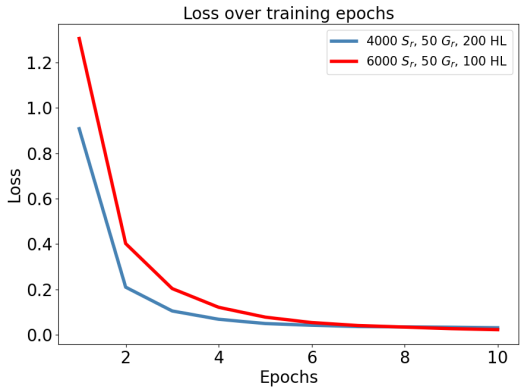
We plot the accuracy and loss curves for both the 4000 length subsequences with 50 gamaka sequences and 200 hidden layers as well as the 6000 length subsequences with 50 gamaka sequences and 100 hidden layers since they gave us the best training and validation accuracy.

Upon investigation, we observe that both the 4000 length subsequences with 50 gamaka sequences and 200 hidden layers and the 6000 length subsequences with 50 gamaka sequences and 100 hidden layers perform similarly in terms of loss convergence and accuracy. So we choose the model with 4000 length subsequences with 50 gamaka sequences and 200 hidden layers as it gives us the better testing accuracy at 91.25%.

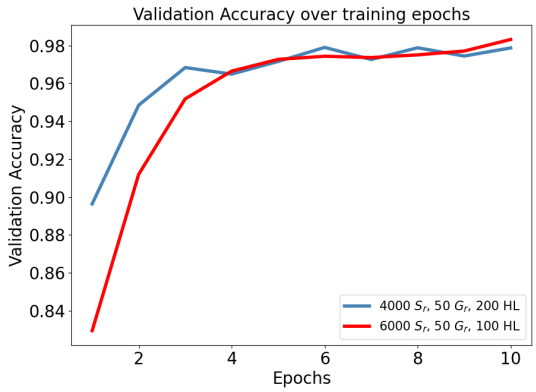
For CMD, we obtain an average accuracy of 90.3% for the 12-cross fold validation with holdout method. This beats the current state of the art model TDMS[19] by 3.6%. For the same hyper-parameter configuration, we obtain an accuracy of 95.6% for HMD. We tabulate the performance comparison of our method with previous Raga recognition techniques performed on the CMD-40 as well as the HMD-30 datasets in 7.2. We compare our results with [19], [11], and [20]. M_{kl} is the version of TDMS that uses the KL divergence, M_f is



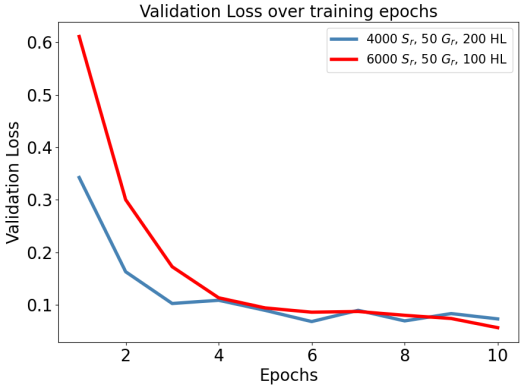
(a) Training accuracy of 4000 length subsequences with 50 gamaka sequences and 200 hidden layers and the 6000 length subsequences with 50 gamaka sequences and 100 hidden layers, across 10 epochs



(b) Training loss of 4000 length subsequences with 50 gamaka sequences and 200 hidden layers and the 6000 length subsequences with 50 gamaka sequences and 100 hidden layers,across 10 epochs



(c) Validation accuracy of 4000 length subsequences with 50 gamaka sequences and 200 hidden layers and the 6000 length subsequences with 50 gamaka sequences and 100 hidden layers,across 10 epochs



(d) Validation loss of 4000 length subsequences with 50 gamaka sequences and 200 hidden layers and the 6000 length subsequences with 50 gamaka sequences and 100 hidden layers,across 10 epochs

Model	Accuracy for CMD	Accuracy for HMD
R_{HAN}	**90.3%**	95.6%
M_{kl}	86.7%	97.7%
M_b	86.7%	97.7%
M_f	81.5%	91.3%
ϵ_{VSM}	68.1%	83.0%
ϵ_{PCD}	73.1%	91.7%

Table 7.2: Accuracy of various models for both datasets.

the version of TDMS that uses the euclidean distance and M_b is the version of TDMS that uses the bhattacharya distance. ϵ_{VSM} represents the vector space model in [20] and ϵ_{PCD} represents the pitch class distribution method in [11]. We represent out results as R_{HAN}. While we outperform other models for the Carnatic music dataset, we do not perform better than the state of the art for the Hindustani music dataset. This could be because of the smaller size of the dataset. We can infer that our model performs better for the sequences in CMD with 480 songs than the sequences in HMD with only 300 songs although the duration of the entire datasets were comparable.

Chapter 8

Discussion

Although we have obtained an admirable accuracy of 90.3% and 95.6% in the CMD and HMD Raga recognition task respectively, we would like to analyse the results and understand the shortcomings of the methodology.

We notice that in a majority, the actual Raga of the music piece and the wrongly predicted Raga are both closely related. Ragas that are very similar to each other in terms of the set of notes and their embellishments are called allied Ragas [54]. Even trained musicians and connoisseurs find it challenging to differentiate between pairs or triplets of allied Ragas.Allied Ragas could be derived from the same base Raga and some of them even share the same vadi and samvadi swara. We find that our model has difficulty in classifying the following pairs of allied Ragas:

- CMD :
  - {Atana, Sri, Madhyamavati }
  - {Kambhoji, Harikambhoji }
  - {Bhairavi, Mukhari}
  - {Sriranjani, Karaharapriya}
- HMD :
  - {Bagesri, Ragesri }

– {Bhairav, Ahir Bhairav }

It is also important to notice that we have achieved validation accuracy of 98% in the classification at the subsequence level for both the datasets. The reason for the disparities in the validation and testing accuracies can be due to the smaller size of the dataset. We just have 12 instances of each Raga in the HMD dataset, and 10 instances for the HMD dataset. With a bigger dataset, this disparity would be lesser as it is a generally accepted fact that the performance of a deep Learning model increases proportionally to the size of the dataset.

As our model performs classification at the subsequence level, we can also perform instantaneous Raga detection. As a subsequence is typically around *20 seconds* of the whole melody, our model does not need the whole audio excerpt to perform Raga detection. Therefore, by just listening to a fraction of the whole melody our method can determine what the whole song's Raga is most likely to be. This can be very useful for students and admirers interested to know more information about the melody being performed at a concert or a piece they are listening to online.

Limitations

Even though our method has achieved considerable results, beating the state-of-the-art in the CMD dataset, it has some limitations. One of the major drawbacks of a Deep Learning based approach is the lack of insight into understanding how the model is able to perform the classification. Other methods like PCD, can provide an intuitive understanding about the distribution of the pitches for melodies of a particular Raga. Further, we do not consider other audio feature aspects like loudness, timbre, the rhythm of the song etc. These features could also help in pre-processing as well as classification of the audio piece. Finally,

Deep Neural Networks take considerably longer to train as compared to traditional Machine Learning based approaches. For this case, 1 epoch of the model took approximately 20 minutes to finish training. This would be even longer for larger datasets.

Chapter 9

Conclusions

In this work, we make an effort to look at the Raga recognition problem in a different perspective. We successfully devise a method to represent digital audio in a format parallel to a document, while extracting melodic patterns crucial to Raga identification. Our technique efficiently classifies digital audio excerpts into their respective Raga class with considerable accuracy of 90.3% for CMD and 95.6% for the HMD. This provides evidence that melodic pattern-based classification approaches can perform effectively in identifying complex patterns with continuous pitch changes and diverse scales.

Future work

Currently, the CMD and HMD are the best available corpus, with 40 and 30 Ragas each. With sizeable and better annotated datasets, we can increase the efficiency of the model manifold and can give rise to the possibility of a complete and expansive Raga detection system, capable of identifying the majority of the existing Ragas. The approach can be extended to other digital media classification problems that can be translated into hierarchically structured data. It can be used to identify the *Turkish makam* or the *Arabic maqam* since they share many similarities with that of the Indian Raga. The model can also be extended or modified to perform Music Genre classification, and Sentiment analysis in music.

Bibliography

[1] Music in india. URL https://www.oxfordbibliographies.com/view/document/obo-9780199757824/obo-9780199757824-0200.xml.

[2] Itc sangeet research academy. URL "http://www.itcsra.org/gamak".

[3] Raag hindustani.com. URL "https://raag-hindustani.com/".

[4] *The Cambridge History of World Music*. The Cambridge History of Music. Cambridge University Press, 2013. doi: 10.1017/CHO9781139029476.

[5] Sandeep Bagchee. Music and emotion-a case for north indian classical music. *Front Psychol*, 8 2115, 12 2017. doi: 10.3389/fpsyg.2017.02115.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

[7] Shreyas Belle and Rushikesh Joshi. Raga identification by using swara intonation. 2010.

[8] Yoshua Bengio and MONTREAL CA. Rmsprop and equilibrated adaptive learning rates for nonconvex optimization. *corr abs/1502.04390*, 2015.

[9] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, O. Mayor, Gerard Roma, Justin Salamon, J. R. Zapata, and Xavier Serra. Essentia: an audio analysis library for music information retrieval. In *International Society for Music Information Retrieval Conference (ISMIR'13)*, pages 493–498, Curitiba, Brazil, 04/11/2013 2013. URL http://hdl.handle.net/10230/32252.

[10] Parag Chordia and Alex Rae. Raag recognition using pitch-class and pitch-class dyad distributions. pages 431–436, 01 2007.

[11] Parag Chordia and Sertan Şentürk. Joint recognition of raag and tonic in north indian music. *Computer Music Journal*, 37:82–98, 09 2013. doi: 10.1162/COMJ_a_00194.

[12] D. Dharini, A. Revathi, and M. Kalaivani. Cd-hmm modeling for raga identification. In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, pages 486–489, 2018.

[13] Pranay Dighe, Parul Agarwal, Harish Karnick, Siddartha Thota, and Bhiksha Raj. Scale independent raga identification using chromagram patterns and swara based features. *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–4, 2013.

[14] Pranay Dighe, Harish Karnick, and Bhiksha Raj. Swara Histogram Based Structural Analysis And Identification Of Indian Classical Ragas. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 35–40, Curitiba, Brazil, November 2013. ISMIR. doi: 10.5281/zenodo.1417841. URL https://doi.org/10.5281/zenodo.1417841.

[15] Karin Dressler. An auditory streaming approach for melody extraction from polyphonic music. pages 19–24, 01 2011.

[16] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[17] Masataka Goto. A predominant-f0 estimation method for polyphonic musical audio signals. 2004.

[18] Clive Greated. Cent, 2001. URL https://www.oxfordmusiconline.com/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-0000005277.

[19] S. Gulati, J. Serrà, K. K. Ganguli, S. Şentürk, and X. Serra. Time-delayed melody surfaces for rāga recognition. In *Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, pages 751–757, New York, USA, 2016.

[20] S. Gulati, J. Serrà, V. Ishwar, S. Şentürk, and X. Serra. Phrase-based rāga recognition using vector space modeling. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 66–70, Shanghai, China, 2016.

[21] Sukrat Gupta, Teja Kanchinadam, Devin Conathan, and Glenn Fung. Task-optimized word embeddings for text classification representations. *Frontiers in Applied Mathematics and Statistics*, 5:67, 2020. ISSN 2297-4687. doi: 10.3389/fams.2019.00067. URL https://www.frontiersin.org/article/10.3389/fams.2019.00067.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[23] Ethan Holder, Eli Tilevich, and Amy Gillick. Musiplectics: Computational assessment of the complexity of music scores. Onward! 2015, page 107–120, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336888. doi: 10.1145/2814228.2814243. URL https://doi.org/10.1145/2814228.2814243.

[24] K. Hussein, E. Tilevich, I. I. Bukvic, and SooBeen Kim. Sonification design guidelines to enhance program comprehension. In *2009 IEEE 17th International Conference on Program Comprehension*, pages 120–129, 2009.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[26] A. P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, 11(6):804–816, 2003.

[27] Gopala Krishna Koduri, Sankalp Gulati, Preeti Rao, and Xavier Serra. Rāga recognition based on pitch distribution methods. *Journal of New Music Research*, 41(4): 337–350, 2012. doi: 10.1080/09298215.2012.735246. URL https://doi.org/10.1080/09298215.2012.735246.

[28] Gopala Krishna Koduri, Vignesh Ishwar, J. Serrà, and Xavier Serra. Intonation analysis of rāgas in carnatic music. *Journal of New Music Research*, 43:72–93, 2014. doi: http://dx.doi.org/10.1080/09298215.2013.866145. URL http://hdl.handle.net/10230/25676.

[29] T. M. Krishna and Vignesh Ishwar. Carnatic music: Svara, gamaka, motif and raga identity. 2012.

[30] Vijay Kumar, Harit Pandya, and C. V. Jawahar. Identifying ragas in indian music. *2014 22nd International Conference on Pattern Recognition*, pages 767–772, 2014.

[31] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[32] Sathwik Tejaswi Madhusdhan and Girish Chowdhary. Tonic independent raag classification in indian classical music. 2018.

[33] S. Samsekai Manjabhat, Shashidhar G. Koolagudi, K. S. Rao, and Pravin Bhaskar Ramteke. Raga and tonic identification in carnatic music. *Journal of New Music Research*, 46(3):229–245, 2017. doi: 10.1080/09298215.2017.1330351. URL https://doi.org/10.1080/09298215.2017.1330351.

[34] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.

[35] M Nawasalkar, Ram Nawasalkar, and Pt Mangrulkar. The concept of rasatatva and raga presentation, 07 2017.

[36] R. K. Nayyar, S. Nair, O. Patil, R. Pawar, and A. Lolage. Content-based auto-tagging of audios using deep learning. In *2017 International Conference on Big Data, IoT and Data Science (BID)*, pages 30–36, 2017.

[37] B. Ong, Emilia Gómez, and S. Streich. Automatic extraction of musical structure using pitch class distribution features. 2006. URL files/publications/88ec9b-LSAS06-egomez.pdf.

[38] Gaurav Pandey, Chaitanya Mishra, and Paul Ipe. Tansen: A system for automatic raga identification. pages 1350–1363, 01 2003.

[39] Ananth Pattabiraman. Carnatic music theory year ii. 2009.

[40] Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[41] FA Rezaur rahman Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan. Attention-based models for text-dependent speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5359–5363. IEEE, 2018.

[42] B. Tarakeswara Rao. Automatic raaga identification system for carnatic music using hidden markov model. 2011.

[43] Suvarnalata Rao and Preeti Rao. An overview of hindustani music in the context of computational musicology. *Journal of New Music Research*, 43(1):24–33, 2014. doi: 10.1080/09298215.2013.831109. URL https://doi.org/10.1080/09298215.2013.831109.

[44] V. Rao and P. Rao. Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2145–2154, 2010.

[45] J. Salamon and E. Gomez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.

[46] J. Salamon, E. Gomez, D. P. W. Ellis, and G. Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014.

[47] Markus Schedl, Emilia Gómez, and Julián Urbano. Music information retrieval: Recent developments and applications. *Found. Trends Inf. Retr.*, 8(2–3):127–261, September 2014. ISSN 1554-0669. doi: 10.1561/1500000042. URL https://doi.org/10.1561/1500000042.

[48] Surendra Shetty and Kk Achary. Raga mining of indian music by extracting arohana-avarohana pattern. *International Journal of Recent Trends in Engineering*, 1, 04 2009.

[49] Sanivarapu SL. India's rich musical heritage has a lot to offer to modern psychiatry. *Indian J Psychiatry*, 57(2), 04 2015. doi: 10.4103/0019-5545.158201.

[50] Rajeswari Sridhar and T.V Geetha. Raga identification of carnatic music for music information retrieval. *SHORT PAPER International Journal of Recent Trends in Engineering*, 1, 04 2009.

[51] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks, 2015.

[52] S.M. Suma and Shashidhar Koolagudi. Raga classification for carnatic music. *Advances in Intelligent Systems and Computing*, 339:865–875, 01 2015. doi: 10.1007/978-81-322-2250-7_86.

[53] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, 2014.

[54] Prithvi Upadhyaya, Suma M., and Shashidhar Koolagudi. Identification of allied raagas in carnatic music. pages 127–131, 08 2015. doi: 10.1109/IC3.2015.7346666.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[56] Venkata Subramanian Viraraghavan, Rangarajan Aravind, and Hema A. Murthy. A statistical analysis of gamakas in carnatic music. In *ISMIR*, 2017.

[57] T. M.H Allen Viswanathan. *Music in South India: The Karṇāṭak Concert Tradition and Beyond; Experiencing Music, Expressing Cultures*. Oxford University Press, 2004.

[58] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[59] Ming-Hsuan Yang and Narendra Ahuja. Gaussian mixture model for human skin color and its applications in image and video databases. In *Storage and retrieval for image*

and video databases VII, volume 3656, pages 458–466. International Society for Optics and Photonics, 1998.

[60] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1174. URL https://www.aclweb.org/anthology/N16-1174.

Appendices

Appendix A

First Appendix

A.1 List of Symbols

- k - Constrained number of cents
- S_r - Subsequence Length
- G_r - Gamaka Sequence Length
- N_s - Number of subsequences
- N_g - Number of Gamaka Sequences per subsequence
- HL - Number of Hidden Layers
- M_{kl} - TDMS using KL divergence
- M_f - TDMS using euclidean distance
- M_b - TDMS using bhattacharya distance
- ϵ_{VSM} - Vector space model
- ϵ_{PCD} - Pitch class distribution method
- R_{HAN} - Our Raga recognition model using HAN

- p_{im} - Pitch sequence
- P_e - Pitch embedding matrix
- f - Frequency
- f_n - Tonic normalized frequency
- T_n - Tonic
- TR - Time Resolution
- t_s - Time duration of a song
- β - Number of pitches representing an audio excerpt
- h_{im}^f - Pitches encoded by the GRU in the forward direction
- h_{im}^b - Pitches encoded by the GRU in the backward direction
- h_{im} - Pitch annotation vector
- α_{im} - Pitch level normalized weight
- u_p - pitch level context vector
- gs_i - Gamaka Sequence vector
- u_g - Gamaka Sequence level context vector
- h_i - Gamaka Sequence annotation
- α_n - Gamaka Sequence level normalized weight
- v - Final subsequence vector